

SwissText 2024

Preprint¹
Proceedings of the 9th SwissText Conference

PREPRINT

10–11 June, 2024
University of Applied Sciences of the Grisons
Chur, Switzerland

¹Please report mistakes to swisstext2024@easychair.org

Organisers:

General Chairs: Corsin Capol, University of Applied Sciences of the Grisons

Mark Cieliebak, Zurich University of Applied Sciences

Program Chair: Albert Weichselbraun, University of Applied Sciences of the Grisons

Scientific Track Chair: Claudiu Musat, Google

Workshops Chair: Elisabeth Maier, Karakun AG

Conference Happiness Manager: Lucas Zimmermann, University of Applied Sciences of the Grisons

Organizational Memory: Manuela Hürlimann, Zurich University of Applied Sciences

Program Committee:*Applied Track*

Albert Weichselbraun, FHGR (Track Chair)

Andrei Popescu-Belis, HEIG-VD

Claudiu Musat, Google

Corsin Capol, FHGR

Hatem Ghorbel, HE-ARC

Mark Cieliebak, ZHAW

Philipp Kuntschik, adesso Schweiz AG

Simone Griesser, HSLU

Scientific Track

Adrian M.P. Brasoveanu, Modul Technology GmbH

Adrien Barbaresi, BBAW

Alexander Mehler, Goethe-University Frankfurt am Main

Andreas Witt, Institut für Deutsche Sprache

Andrei Popescu-Belis, HEIG-VD / HES-SO

Asya Fadeeva, Google

Blagoj Mitrevski, Google

Don Tuggener, ZHAW

Egon Werlen, FFHS

Ekaterina Lapshinova-Koltunski, University of Hildesheim

Felix Sasaki, Cornelsen Verlag GmbH & TH Brandenburg

Georg Rehm, DFKI

Gerold Schneider, University of Zurich

Heike Zinsmeister, University of Hamburg

Helmut Schmid, Ludwig Maximilian University of Munich

Juergen Vogel, BFH

Ludovic Tanguy, CLLE-ERSS

Manfred Vogel, FHNW

Margot Mieskes, University of Applied Sciences, Darmstadt

Marie-Francine Moens, Katholieke Universiteit Leuven

Miriam Butt, Konstanz

Musat Claudiu, Google (Track Chair)

Paolo Rosso, Universitat Politècnica de València
Philippe Schlattner, Google
Sebastian Pado, University of Stuttgart
Serge Heiden, ENS de Lyon
Thomas Schmidt, IDS Mannheim
Torsten Zesch, FernUni Hagen
Valia Kordoni, Humboldt University Berlin
Yves Scherrer, University of Helsinki

Partners:

Bern University of Applied Sciences
CLARIN-CH
Lucerne University of Applied Sciences and Arts
NLP Zurich
School of Business and Engineering Vaud
Swiss Distance University of Applied Sciences
Swiss Group for Artificial Intelligence and Cognitive Science
swiss made software
Università della Svizzera italiana
University of Applied Sciences and Arts Northwestern Switzerland
University of Applied Sciences of the Grisons
University of Applied Sciences Western Switzerland - Valais University of Neuchâtel
University of Zurich
Zurich University of Applied Sciences (datalab, InIT, CAI, ILC)

Sponsors:

Adesso
Contexity
D ONE
Google
Graubündner Kantonalbank
Inventx AG
Karakun
Swissdox.ch

Invited Speakers:

Lewis Tunstall, HuggingFace
Torsten Zesch, FernUniversität Hagen
Michèle Balthasar, Balthasar Legal
Leon Derczynski, University of Copenhagen
Jesse Berent, Google

Invited Talks

Lewis Tunstall: Aligning Foundation models in 2024

Most foundation models typically undergo two stages of training: pre-training and post-training. Pre-training involves training on large-scale datasets to develop a broad understanding of language and other modalities, while post-training tailors the model for specific tasks and improves alignment with human values and objectives. In this talk, I will provide an overview of the primary methods for alignment, emphasizing practical insights and ongoing challenges. These methods include Reinforcement Learning from Human Feedback (RLHF) as well as more recent, popular approaches such as Direct Policy Optimization (DPO) and Kahneman-Tversky Optimization (KTO). Additionally, the talk will explore the role of synthetic data in alignment, where a strong model like GPT-4 is used to distill its capabilities into small, yet performant models. I will also discuss the challenges associated with evaluating aligned models and highlight the use and limitations of LLMs as proxies for human evaluation.

Michèle Balthasar: Understanding AI and Copyright

The integration of generative Artificial Intelligence (AI) in creative processes raises complex copyright questions. The keynote highlights two key aspects: the copyright protection of AI outputs and the use of copyrighted works in AI training. Can works generated by AI be protected by copyright? Are copyrighted works allowed to be used for AI training? These questions are analyzed against the backdrop of current legal frameworks in the EU and Switzerland. The keynote examines the limits of copyright protection, particularly the role of human contribution and the requirements for creating copyright-protected works. Additionally, the keynote discusses the legal implications and risks of unauthorized use, exploring how both creators and users can navigate these emerging challenges in the digital age.

Jesse Berent: Online Handwriting Recognition in the Vision-Language Models age

Digital note-taking and hand drawn input is gaining popularity, offering a durable, editable, and easily indexable way of storing notes in the vectorized form known as digital ink. At the same time, the adoption of tablets with touchscreens and styluses is increasing, and a key feature is interpreting handwritten or drawn input. This talk explores the intersection of handwriting recognition and modern AI by focusing on two new approaches. The first part delves into the application of large vision-language models (VLMs) to online handwriting recognition using new representations and tokenizers. This approach, which is compatible with off-the-shelf models and methods, offers a promising avenue for seamless integration of online handwriting recognition into existing multi-modal models. The second part will focus on converting images of handwriting (pen-and-paper notes) into digital ink with VLMs. This capability bridges the gap between traditional and digital note-taking, facilitating seamless integration of handwritten content into digital AI-assisted workflows. The presentation will conclude with a discussion of the broader implications of these advancements for the future of handwriting recognition and human-computer interaction.

Torsten Zesch: Why aren't available NLP methods more frequently used in the classroom? - The curious case of free-text answer scoring

Giving feedback on free-text answers (in the form of grades or helpful hints) is a core educational task. Despite a large body of research on the topic, assisting teachers with this task remains challenging. In this talk, we outline the linguistic and external factors influencing the performance level that NLP methods may reach for a given question. However, even in settings where automatic performance rivals humans, there are various practical requirements often overlooked in research that hinder adoption in the classroom.

Leon Derczynski: Attacks on large language models

Generative language models got a lot better recently; they escaped the research "yard" and now act in public. This has the consequence of risking reputational and other harms as models produce undesirable output. Simultaneously, generative language models have become alluring targets for hackers, who attempt to subvert them. Unlike traditional information security, the entry-level language for interacting with language models is natural language (at least for English), and so no coding skills are required. This has led to a huge effervescent and creative frontier where thousands of people attack language models in an attempt to make them misbehave. We investigate how people attack language models - what motivates them, what their techniques and strategies are, how to describe the activity, and how attacks are developed. And, of course, what the attacks are. New resources have emerged for automatically attacking language models and for collecting various attacks from the community, forming an active baseline for discovering language model vulnerabilities. We will discuss the issues involved in collating attacks on language models, and how to use this information to make model interaction safer.

Table of Contents

1 Scientific Regular Track	1
Can OpenSource beat ChatGPT? - A Comparative Study of Large Language Models for Text-to-Code Generation	2
<i>Luis Mayer, Christian Heumann and Matthias Aßenmacher</i>	
Classifying multilingual party manifestos: Domain transfer across country, time, and genre	22
<i>Matthias Aßenmacher, Nadja Sauter and Christian Heumann</i>	
Data Augmentation for Multi-Class Eating Disorders Text Classification	33
<i>Ghofrane Merhbene and Mascha Kurpicz-Briki</i>	
Fine-tuning the SwissBERT Encoder Model for Embedding Sentences and Documents	42
<i>Juri Grosjean and Jannis Vamvas</i>	
How to make the first move? Analyzing the impact of an acquisition model in deep active learning for a sequence labeling task	51
<i>Ngoc Duyen Tanja Tu</i>	
Multimodal Conversational Interactions for Facial Composites: A Case for Stateful Prompt Orchestration	60
<i>Rico Staedeli, Roman Leu, Jasmin Heierli, Max Meisterhans, Elena Gavagnin and Alexandre de Spindler</i>	
Sentiment- and Keyword-Controllable Text Generation in German with Pre-trained Language Models	69
<i>Paulina Aleksandra Żal, Guang Lu and Nianlong Gu</i>	
The Value of Pre-training for Scientific Text Similarity: Evidence from Matching Grant Proposals to Reviewers	90
<i>Gabriel Okasa and Anne Jorstad</i>	
Tracing Linguistic Footprints of ChatGPT Across Tasks, Domains and Personas in English and German	103
<i>Anastassia Shaitarova, Nikolaj Bauer, Jannis Vamvas and Martin Volk</i>	
Zero-Shot Award Criteria extraction via Large Language Models from German Procurement Data from Switzerland	114
<i>Veton Matoshi, Luca Rolshoven and Matthias Stürmer</i>	
2 Scientific Junior Track	135
Evaluating Labor Market Biases Reflected in German Word Embeddings	136
<i>Leander Rankwiler and Mascha Kurpicz-Briki</i>	
Exploring Personalized Learning Support through Retrieval Augmented Generation: A Feasibility Study	146
<i>Petar Mladenov, Luis Pinheiro, Dino Pelesevic and Jasmin Heierli</i>	
SDG Classification Using Instruction-Tuned LLMs	150
<i>Tobias Fankhauser and Simon Clematide</i>	
Triple Detection in German Verb-based Sentiment Inference: The Case of Novel Verbs	159
<i>Dylan Massey</i>	

3 Applied Track	165
adesso Intelligent Agent - an add-on ecosystem for extending and empowering RAG systems	166
<i>Leon Stolpmann, Rafael Dubach, Stephan Symons and Philipp Kuntschik</i>	
An LLM-based Chatbot for Joint Access to Customer and Corporate data	167
<i>Teo Ferrari, Vincent Coriou, Rastislav Kovac, Vuk Vuković, Daniel Dobos, Fatemeh Borran and Andrei Popescu-Belis</i>	
Annotation Tool for Dataset Creation	168
<i>Patrick Giedemann, Pius von Däniken and Jan Milan Deriu</i>	
Automatic Identification of Swiss German Dialects via Speech-to-Phoneme Transcriptions	169
<i>Laura Bolliger, Safiyya Waldburger, Mark Cieliebak and Jasmina Bogojeska</i>	
Can NLP models and methods be applied to EEG data?	170
<i>Lino Casanova and Andreas Marfurt</i>	
ChaLL - A Chatbot for Language Learners	171
<i>Manuela Hürlimann, Luzia Sauer, Gerold Schneider, Johannes Graën, Jean-Philippe Goldman, Janick Michot, Katsiaryna Mlynchyk, Ahmet Yavuz Uluslu, Irina-Camelia Stroescu, Jan Deriu, Michael Geiss and Mark Cieliebak</i>	
Chatbot Evolution at Swisscom Customer Support	172
<i>Fatemeh Borran</i>	
Climate Change Chatbot	173
<i>Roger Waldvogel and Albert Weichselbraun</i>	
Creating a virtual receptionist to lighten your employees' workload: a case study with the canton of Jura and subsidies for training.	174
<i>Emmanuel de Salis, Jonathan Guerne and Hatem Ghorbel</i>	
Dialect Identifications with Large Language Models	175
<i>Vani Kanjirangat, Ljiljana Dolamic and Fabio Rinaldi</i>	
Harnessing LLM's for generating Patient Discharge Reports	176
<i>Norman Süssstrunk and Albert Weichselbraun</i>	
Interlex –A search engine to explore the interconnectedness of Swiss legal texts	177
<i>Selena Calleri, Michelle Wastl, Bojan Peric and Andreas Abegg</i>	
Kickstarting legal multi-label classification experimentation	178
<i>Claudia Schulz, Martina Forster, Prudhvi Nokku and Stavroula Skylaki</i>	
Language Models Explore the Linguistics of Chess	179
<i>Lars Schmid, Jerome Maag, Mark Cieliebak and Pius von Däniken</i>	
Leveraging LLMs to Enhance Clinical Record Analysis and Retrieval	180
<i>Lorenzo Ruinelli, Amos Colombo, Oscar William Lithgow Serrano, Andrea Franchini, Joseph Cornelius, Sandra Mitrovic and Fabio Rinaldi</i>	
Medicrawl: an ML based system to automate the search of illegal products in ecommerce	181
<i>Nicolas Perez Gonzalez</i>	
Navigating the Commodity Market with Language Models	182
<i>Himmet Kaplan, Albert Weichselbraun and Martin Tschudy</i>	

NLP in support of Pharmacovigilance	183
<i>Fabio Rinaldi, Lorenzo Ruinelli, Roberta Nosedà, Oscar William Lithgow Serrano and Sandra Mitrović</i>	
Online BERT-based Topic Modelling	184
<i>Eric Gericke, Nicolas Jamet, Tian Guo and Martin Schüle</i>	
Orbis2 - A Natural Language Processing Benchmarking Framework that supports Drill Down Analyzes	185
<i>Norman Süsstrunk, Roger Waldvogel, Andreas Murk, André Glatzl and Albert Weichselbraun</i>	
Personalized and Interactive Education in Migraine using Artificial Intelligence	186
<i>Fabio Rinaldi, Oscar William Lithgow Serrano, Andrea Franchini, Chiara Zecca, Giulia Mallucci and Alberto Cordella</i>	
Presenting BUST - A benchmark for the evaluation of system detectors of LLM-Generated Text	187
<i>Joseph Cornelius, Oscar William Lithgow Serrano, Sandra Mitrović, Ljiljana Dolamic and Fabio Rinaldi</i>	
PROMISE: Model-Driven Stateful Prompt Orchestration for Persuasive Conversational Interactions	188
<i>Wenyuan Wu, Jasmin Heierli, Max Meisterhans, Adrian Moser, Andri Färber, Mateusz Dolata, Elena Gavagnin, Alexandre de Spindler and Gerhard Schwabe</i>	
Reducing complexity of Scientific Literature by automated simplification and translation	189
<i>Nico Colic and Fabio Rinaldi</i>	
STT Explorer: A Look Beyond WER	190
<i>Katsiaryna Mlynchyk, Gabriel Hunziker, Lars Mosimann, Ralph Scheu and Mark Cieliebak</i>	
Swiss AI Initiative - Collecting Large Amounts of High-Quality Data for Training Large Language Models	191
<i>Jan Deriu, Maud Ehrmann, Emanuela Boros, Maximilian Böther, Christiane Sibille, Ihor Protsenko, Marta Brucka, Imanol Schlag and Elliott Ash</i>	
Understanding Customer Sentiment with NLP from Sparse Labelled Data	192
<i>Daniel Perruchoud and Joseph Weibel</i>	
Virtual Kids: Using LLMs for Role Playing in Professional Communication Training	193
<i>Don Tuggener</i>	
What can we discover about panic and anxiety from bloggers in Quora and Reddit?	194
<i>Sandra Mitrović and Oscar William Lithgow Serrano</i>	
Whisper Fine-Tuning for Swiss German: A Data Perspective	195
<i>Claudio Paonessa, Vincenzo Timmel, Manfred Vogel and Daniel Perruchoud</i>	
4 Shared Task	196
SwissText 2024 Shared Task: Automatic Classification of the United Nations' Sustainable Development Goals (SDGs) and Their Targets in English Scientific Abstracts	197
<i>Simon Clematide, Martin Volk, Tobias Fankhauser, Lorenz Hilty and Jürgen Bernard</i>	
An Efficient Workflow Towards Improving Classifiers in Low-Resource Settings with Synthetic Data	202
<i>Adrian M.P. Braşoveanu, Albert Weichselbraun, Lyndon J.B. Nixon and Arno Scharl</i>	
MeHuBe at SwissText 2024 Shared Task 1: Ensembling and QLoRA with Retrieved Citations for Fine-Grained Classification of Sustainable Development Goals	207
<i>Fernando de Meer Pardo, Hanna Hubarava and Vera Bernhard</i>	
Scouting out the Border: Leveraging Explainable AI to Generate Synthetic Training Data for SDG Classification	218
<i>Norman Süsstrunk, Albert Weichselbraun, Andreas Murk, Roger Waldvogel and André Glatzl</i>	

Swiss Text - Shared Task (SDG Classification) - Task 1	224
<i>Manuel Bolz, Andreas Loizidis and Kevin Bründler</i>	
5 Workshops	230
AI Support Systems for Academic Research	231
<i>Susie Rao</i>	
RAG: Unveiling the Power of Retrieval-Augmented Generation	233
<i>Jonathoan Guerne</i>	
Grounding Generative AI Models	235
<i>Holger Keibel</i>	
Next-Gen Cleantech Solutions: Mining Insights from Media and Patent Data with Natural Language Processing (NLP) and Large Language Models (LLMs)	237
<i>Guang Lu</i>	
Battle of NLP Ideas	239
<i>Manuela Hürlimann, Katsiaryna Mlynchyk and Philipp Kuntschik</i>	

Chapter 1

Scientific Regular Track

Can OpenSource beat ChatGPT? – A Comparative Study of Large Language Models for Text-to-Code Generation

Luis Mayer¹, Christian Heumann¹, Matthias Aßenmacher^{1,2}

¹Department of Statistics, LMU Munich, Germany,

²Munich Center for Machine Learning (MCML)

Correspondence: luis.mayer@campus.lmu.de, matthias@stat.uni-muenchen.de

Abstract

In recent years, large language models (LLMs) have emerged as powerful tools with potential applications in various fields, including software engineering. Within the scope of this research, we evaluate five different state-of-the-art LLMs - Bard, BingChat, ChatGPT, Llama2, and Code Llama - concerning their capabilities for text-to-code generation. In an empirical study, we feed prompts with textual descriptions of coding problems sourced from the programming website LeetCode to the models with the task of creating solutions in Python. Subsequently, the quality of the generated outputs is assessed using the testing functionalities of LeetCode. The results indicate large differences in performance between the investigated models. ChatGPT can handle these typical programming challenges by far the most effectively, surpassing even code-specialized models like Code Llama. To gain further insights, we measure the runtime as well as the memory usage of the generated outputs and compared them to the other code submissions on LeetCode. A detailed error analysis, encompassing a comparison of the differences concerning correct indentation and form of the generated code as well as an assignment of the incorrectly solved tasks to certain error categories allows us to obtain a more nuanced picture of the results and potential for improvement. The results also show a clear pattern of increasingly incorrect produced code when the models are facing a lot of context in the form of longer prompts.

1 Introduction

Natural Language Processing (NLP) is an interdisciplinary field at the intersection of artificial intelligence, computer science, and linguistics. Its primary goal is to enable computers to understand, interpret, and generate human language in a valuable manner. In the early days of NLP, the focus was primarily on rule-based systems

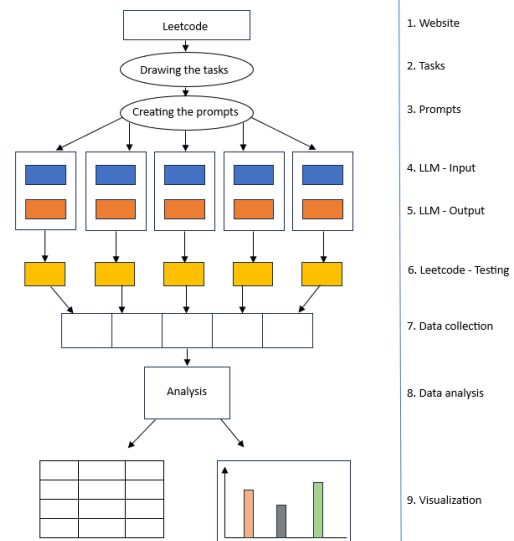


Figure 1: Schematic overview of the evaluation pipeline for the LLMs' performance on text-to-code generation.

and basic statistical models (Zhou et al., 2020). These systems, while innovative for their time, often struggled with constant adjustments and maintenance, linguistic variability, and scalability (Manning and Schütze, 1999; Jurafsky and Martin, 2023). With the advent of Machine Learning, particularly Deep Learning, a paradigm shift occurred. Deep Learning models, especially since the end of the last decade, so-called large-language models (LLMs), based on enormous amounts of data and immense computational power, have demonstrated an unprecedented ability to understand and generate language (Zhang et al., 2023a). The emergence of LLMs is closely linked to the development of fundamental architectures such as the Transformer, introduced by Vaswani et al. (2017), which fostered a new era of language models including well-known LLM-based Chatbots such as ChatGPT (OpenAI,

2022). LLMs have proven themselves very useful for a variety of tasks ranging from answering questions, over various classification tasks to summarizing texts or writing essays. In addition to handling human language, some LLMs are also able to deal with code (Feng et al., 2020). Based on textual descriptions of a programming problem as an input, these models can generate code in different programming languages as an output. The application of translating natural language into syntactically and semantically correct code has far-reaching implications, particularly in software development, where it can bridge the gap between domain experts and developers.

Contribution With this paper, we contribute to a better understanding of recently developed LLMs in terms of their capabilities for code generation by examining them in an empirical study as depicted in Figure 1. Within this study, we investigate the performance of five different LLMs – ChatGPT, BingChat, Bard, Llama2, and Code Llama – in terms of text-to-code generation for the programming language Python and compare their performance with respect to (i) correctness, (ii) runtime, and (iii) memory usage. We draw further conclusions by making use of the different difficulty levels provided in conjunction with the task formulations.

2 Related Work

Studying the abilities of LLMs in coding-related tasks is a dynamically developing field of research at the moment since ongoing developments of LLMs enable their usage for a wide range of coding and programming tasks. Thus, it is crucial to distinguish between the different kinds of tasks, as their applications often differ considerably from each other. The paper of Zhang et al. (2023c) provides an overview of several types of code-processing tasks. In the realm of code correction, Sobania et al. (2023) investigate ChatGPT’s bug-fixing performance by confronting the LLM with 40 faulty Python code snippets. In a similar study Zhang et al. (2023b) examine the proficiency of ChatGPT in repairing flawed Java programs and compare its results to task-specific models like CodeT5 and PLBART. Concerning code summarization, Sun et al. (2023) present an insightful study on the evaluation of ChatGPT’s abilities to create comments for Python code snippets. Likewise to the works mentioned above, also in the area of text-to-code

generation, most studies conducted so far have focused on testing ChatGPT. Geng et al. (2023) evaluate the performance of ChatGPT within the frame of an introductory-level functional language programming course, while Piccolo et al. (2023) explore its capabilities in solving programming tasks from an introductory bioinformatics course. The code generation abilities of ChatGPT were evaluated in another study by Buscemi (2023), where the authors benchmark the model on various tasks in ten different programming languages. In the study by Muennighoff et al. (2023), the focus is on open-source LLMs that are examined in the three task areas of code correction, code explanation, and code synthesis in six different programming languages. The work of Austin et al. (2021) also deals with code synthesis in Python using a wider range of LLMs. In their study, the models are employed both with and without fine-tuning and a comparative analysis of the results was conducted. A study that is somewhat similar in structure to ours, but with a greater emphasis on only whether the code is correct or not, is presented by Deste-fanis et al. (2023). In this work, ChatGPT and Bard are prompted to generate Java code based on a provided code description.

3 Materials and Methods

3.1 Data Set

The tasks for this study are sourced from LeetCode¹, a training website providing a diverse range of programming problems, including (but not limited to) algorithms, dynamic programming, or graphs. LeetCode categorizes the tasks into different topics (e.g. array, math, or sorting) and three difficult levels (easy, medium, and hard), the latter of which we exploit to measure the LLMs’ performance more granularly. Users can pick tasks and try to solve them in different programming languages. LeetCode also provides test cases to check possible solutions and several evaluation metrics for accepted solutions. The focus of this study is to test the performance of LLMs in math and statistics-related tasks. The three task topics in the LeetCode repository best suited for these constraints and most frequently represented are math, matrix, and counting. For the first two subject areas, 30 tasks are selected consisting of 10 tasks from each of the three different difficulty levels. Only 29 tasks were selected for counting, as there were only 9 tasks

¹<https://leetcode.com/>

with a difficulty level of "hard" at the time of data collection. Across all three topics, this results in 89 tasks that are used in this study. After filtering for the appropriate types and difficulty levels, the tasks are randomly drawn using the *pick one* button in LeetCode.

3.2 Models

ChatGPT is an instruction-tuned LLM based on models from the GPT series (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023) developed by OpenAI and initially published in November 2022. The initial and freely accessible variant of ChatGPT was based on the GPT-3.5 turbo (since March 2023), while starting from February 2023 ChatGPT paid access to a newer version based on GPT-4 has been available. For the study conducted in this work, the ChatGPT model based on the GPT-3.5 turbo version was employed, making our performance estimates for ChatGPT somewhat conservative. Further, this ChatGPT was trained on a corpus with a cut-off date in September 2021 and thus does not have access to information newer than October 2021.

BingChat is a conversational LLM-based feature for Microsoft’s search engine Bing that can also be used for programming tasks. So instead of typing a search query into Bing, the user can interact with BingChat (Xuan-Quy et al., 2023). It was launched in February 2023 and is powered by the GPT-4 model from OpenAI. Due to an internet connection, BingChat has access to all the latest information and is not limited in a way like ChatGPT.

Bard² is the instruction-tuned variant of an LLM developed by Google. It was first released in March 2023 on a limited basis, followed by a full release in May 2023. The chatbot was at first powered by LaMDA (Language Model for Dialogue Application; Thoppilan et al., 2022), but since May 2023 Bard has been based on Google’s PaLM 2 (Pathways Language Model 2; Anil et al., 2023). PaLM 2 uses compute-optimal scaling to adjust the model size to the number of tokens in the pre-training corpus. This updated approach makes PaLM 2 more compact than its predecessor PaLM, while still providing higher efficiency, faster inference, and a reduced parameter count (Anil et al., 2023). Similar to BingChat, Bard can access newly appearing information via the internet and is thus able to

²In February 2024, Bard was renamed to Gemini.

also provide links to websites and other online resources.

Llama2 is an open-source LLM released by Meta in July 2023. It has been trained with 40% more data than its predecessor Llama and is capable of handling twice its context length (4096 vs. 2048 tokens; Touvron et al., 2023). Llama2 is offered in three versions with either 7 billion (7B), 13 billion (13B), or 70 billion (70B) parameters. For this study, we choose the largest available version (70B). The chatbot *llama-2-70b-chat*, provided by the website Replicate, was employed for this purpose.

Code Llama is a family of LLMs developed by Meta and published in August 2023. It’s a variant of Llama2 specifically aligned for coding-related tasks, which was fine-tuned on large data sets of programming code. In fact, Code Llama is trained on the same code-specific datasets as Llama2 but using more samples from the same data set for a longer training time (Meta, 2023). In addition to the standard Code Llama model, there are two other versions: a Python-specialized version called Code Llama-Python and Code Llama-Instruct, which is fine-tuned for understanding prompts in natural language. All three types of Code Llama exist in a 7B, a 13B, and a 34B version (Rozière et al., 2023). Since we provide the LLMs with instructions in natural language, the variant Code Llama-Instruct with the largest parameter count (34B) was selected. The chatbot *Code Llama Instruct (34B)* from the website together.ai, which offers several LLMs, is used for this purpose.

4 Experimental Setup

As depicted in Figure 1, the prompts are created based on the pre-selected tasks (cf. Sec. 3). To understand the process, it is important to closely inspect their structure (cf. Appendix A, Fig. 7). Initially, each task consists of three parts. To create a coherent prompt, a fourth part is added and all paragraphs are separated by inserting a blank line. The first part describes the problem to be solved by a function to be created with code. In the subsequent part, one to three examples are presented to show exemplary inputs and outputs of the function. Partially these are supplemented by additional explanations. All this is complemented by a final part listing constraints and conditions that the function is required to fulfill. The fourth

section we added consists of the request for the task to be solved in the programming language Python. We consistently use the expression "Write the code for this task in Python" followed by the name of the Python function with the arguments to be included since the automated tests can only be performed if the generated code uses the function names and arguments specified by LeetCode. Since this information is accessible on LeetCode for each task, we consistently use the prefix "*Start with: <function_name and arguments>*". Some of the tasks in LeetCode have integrated illustrations in the example part, which allows the user a better understanding of the problem. Since the five examined models are not capable of processing visual information, those are not included in the prompt.

After conducting all the mentioned alterations, we also kept track of the number of tokens that constituted the prompts. Then, the prompts are passed to all five models as input. For each task a new chat is started, so that a conversation with the LLM always contains exactly one prompt and one generated output per task. Since the answers often consisted of code blocks interleaved with text descriptions and exemplary applications (cf. Appendix B, Fig. 8), it is crucial to locate the part that contains the function to be implemented. This part is inserted into the code field in LeetCode for the corresponding task before LeetCode's test procedure is executed. Depending on the test result, the feedback from LeetCode differs: If a task is not solved correctly, LeetCode indicates the type of error (e.g. "wrong answer" or "invalid syntax") and the number of correctly passed tests. The different error categories are described in more detail in Section 5. For correctly answered tasks, LeetCode displays a new window featuring runtime, memory usage, and corresponding ranks for the code.

LeetCode measures the runtime in milliseconds and memory usage in megabytes for each successful code submission. Furthermore, these measured values are retained by LeetCode to compare them to all other runtime and memory usage values from submissions by other users, provided that the task and programming language are the same. LeetCode then calculates a quantile ranking indicating the percentage of correct submissions that a given solution surpasses in terms of performance. As an illustration, consider a correct submitted code with runtime and memory usage rankings of 52% and 99%, respectively. In the context of runtime, this implies that the code

is only slightly faster than half of all properly submitted codes. On the other hand, in terms of memory usage, the code outperforms nearly all other submissions, with only one percent exhibiting lower memory usage. The described metrics are stored for each task and model. The findings are presented together with visualizations in the next section.

5 Results

Post-Processing of the generated outputs As already hinted at in the last section, the generated responses of the LLMs often consist of interleaved text and code blocks. However, our experiments show that the code does not necessarily have to be in a code block. Overall, three categories of the code location are found:

- All code in a single code block. Before/after this there may be optional text blocks. This is the standard response type for ChatGPT, Bard, and Llama2.
- The generated code is located in a text block. This text block might also contain descriptions and comments about the code. This case occurs sometimes with BingChat and Code Llama.
- The code is distributed across text and code blocks. This case occurs sometimes with BingChat and Code Llama.

If code situated within a text block (second and third category) is tested within LeetCode, it is immediately assessed as incorrect. This arises from the structure of text blocks, where each line invariably begins with a word. Hence, the code is e.g. not indented, as it should be for instance in if-statements or for-loops as illustrated in Figures 9 and 10 (Appendix B). Since only due to the fact of missing indentations, otherwise correct code may not be recognized as such, we decided to intervene in such cases and to perform the necessary indentations subsequently for all affected models. This was especially crucial for Code Llama, as its code was consistently generated within text blocks. For this reason, post-processing was necessary for all 89 tasks for Code Llama. In four instances, BingChat was also affected. The adjustments were always applied after the code generation and before testing on LeetCode. The approach described

here was consistently employed for all analyses and results presented in the following.

Correctness Table 1 and Figure 2 clearly show that the LLMs’ performance differs notably. The best-performing model is ChatGPT, which is the only model to solve more than 50% of the tasks correctly. The runner-up is the second GPT-based model, BingChat, ahead of Bard. The two Llama models solve by far the fewest tasks correctly, with a share of not even 10%. Code Llama still performs slightly better than Llama2. Although Code Llama is based on Llama2, their output only matches in two (correctly solved) cases.

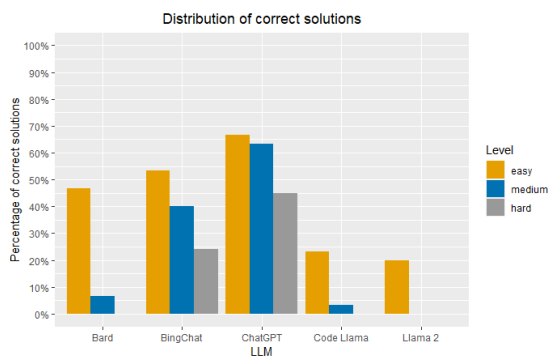


Figure 2: Percentage of correctly solved tasks per LLM.

Across all evaluated LLMs we further observe (cf. Fig. 2) that the share of correctly solved tasks decreases with a higher difficulty level. While all five models are able to complete at least some of the tasks on the easy level, only four of the models achieve the right solutions on the medium level. Further, only the two GPT-based models can successfully solve any task on the highest level. While ChatGPT and BingChat consistently solve more than 20% of tasks correctly, only Bard and Code Llama are above this mark on the easy difficulty level. At the medium difficulty level, these two LLMs drop below 10%, corresponding to two and one correct solution, respectively.

Performance metrics In the next step, we closely inspect the correctly solved tasks. Our focus specifically lies on the runtime and memory usage metrics as explained in Section 4, which LeetCode only returns for each correct submission. As already mentioned, Bard and Code Llama only have one and two right solutions at the medium level which is why it is hardly possible to draw any conclusions and so we omit the combination of these models and difficulty levels in the following figures. It is further important to note that the

following two bar plots consistently depict average values, which are composed of the rank values of the individual correctly solved tasks.

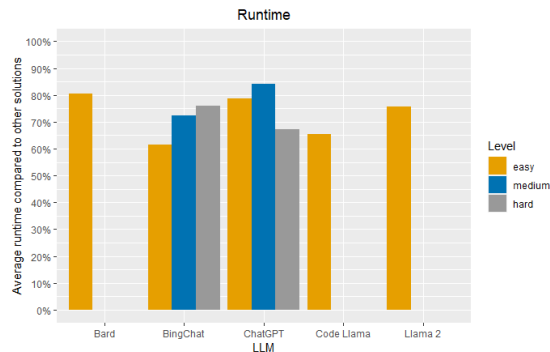


Figure 3: Average runtime ranking (quantiles) of the generated code per LLM and difficulty level.

In Figure 3 the average of the runtime ranks per LLM and difficulty level is depicted. For a better understanding of the plot, consider the bar on the left side: Bard solved 14 tasks correctly for the difficulty level "easy", so Leetcode also calculated 14 ranking values for the runtime, the average of which is displayed here. A value of 70% means that the average runtime of the generated code beats 70% of all other code submissions for this task. Conversely, only 30% of the submitted solutions are quicker. It is noticeable that the (correct) LLM-generated solutions exhibit a comparatively low average runtime, as they are ranked above 50% on average. While concerning the number of correct solutions there are clear differences between the models and the difficulty levels, this is not the case for the runtime: A decrease in runtime performance with higher difficulty levels is not evident. It can be argued that ChatGPT outperforms the other models concerning the number of correctly solved tasks, but not (consistently) in terms of runtime. The ranking values for all models lie between 61% (BingChat, "easy") and 84% (ChatGPT, "medium"). These values suggest that the runtime of the code produced by the models is often lower than that of other code submissions. As faster code is favorable in the context of software development, this is an aspect in which LLMs might be able to contribute effectively.

Concerning memory usage, the average values per model and difficulty level are calculated and displayed in Figure 4 according to the same principles as for the runtime. A higher value signifies that the code generated by the models utilizes less memory than solutions provided by other users.

	Bard	BingChat	ChatGPT	Code Llama	Llama2
correct	18% (16)	39% (35)	58% (52)	9% (8)	7% (6)
incorrect	82% (73)	61% (54)	42% (37)	91% (81)	93% (83)

Table 1: Relative (Absolute) frequencies of correct and incorrect solutions for all evaluated models.

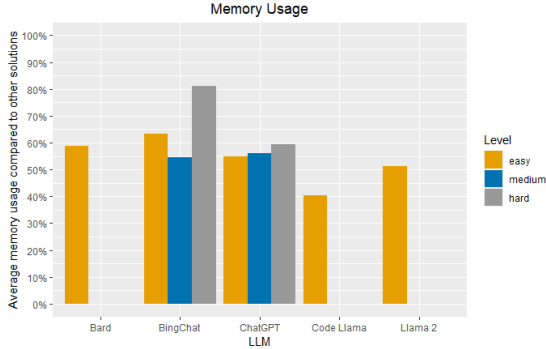


Figure 4: Average memory usage ranking (quantiles) of the generated code per LLM and difficulty level.

Reduced memory usage is advantageous as it allows for easier scalability to handle larger data volumes without additional memory requirements. Besides, code with lower memory usage is often more efficient, especially when required to operate on systems with limited RAM. The main takeaway from taking this angle is that the differences between the models are (again) not as pronounced as observed for correctness. While ChatGPT only achieves similarly high values as the other models, BingChat performs notably better across difficulty levels. Except for Code Llama, all values are again above 50%, although the values are mostly a bit lower than those for the runtime. Exceptional performance can be reported for BingChat at the "hard" level with 81%.

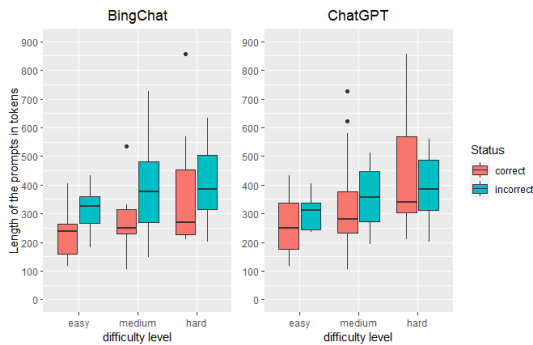


Figure 5: Distribution of prompt lengths of correct and incorrect solutions for BingChat and ChatGPT

Further insights and error analysis Figure 5 presents the distribution of prompt lengths of correct and incorrect solutions, measured by the number of tokens the prompts consist of. For our main analysis, we focus solely on BingChat and ChatGPT since they are the best-performing LLMs regarding the number of correct solutions and at the same time the only ones with correct solutions at all difficulty levels. The boxplots for the remaining three LLMs can be found in Figure 11 in Appendix C. We observe that the lengths of prompts are smaller for correct solutions than for incorrect solutions at all three difficulty levels. This conversely seems to imply that shorter prompts are positively associated with the likelihood of the models producing a correct solution.

Concluding the analysis, Table 2 summarizes the ten most frequent errors across all five models. A complete enumeration of all errors can be found in Table 3 in Appendix D. We learn from this table that the error category "wrong answer" is by far the most frequent one with a share of > 50%. If this error is encountered, this indicates that the code submission has passed only a specific number of tests in LeetCode, which is less than the total number of tests required for a correct solution. Therefore, the generated outputs affected by "wrong answer" do not cover the required functionality expected from the code. The second most common reason why solutions from LeetCode were rejected is "syntax error" with approximately 15%. All other error categories depicted in Table 2 are clearly below 10%. The category "type error" (5.5%) occurs when a function or operation is applied to an object of an inappropriate type, such as attempting to concatenate an object of type integer with an object of type string. If undefined variables are used in the code, the error message falls into the category "name error" (5.2%). The error message "time limit exceeded" (4.6%) means that LeetCode cannot perform all tests, potentially due to an infinite loop in the generated code or simply because the code execution time exceeds LeetCode's time limit. An "index error" (3%) appears when trying to access an index that is outside the bounds of a

	Error type	Count	Share in % of all errors
1	wrong answer	178	54.3%
2	syntax error	48	14.6%
3	type error	18	5.5%
4	name error	17	5.2%
5	time limit exceeded	15	4.6%
6	indentation error	10	3.0%
7	index error	10	3.0%
8	attribute error	9	2.7%
9	no code generated	7	2.1%
10	zero division error	4	1.2%

Table 2: Ten most frequent error categories among all LLMs

sequence type, such as arrays or lists. "Attribute errors" (2.7%) happen when accessing attributes of an object that it is not ascribed. The error category "no code generated", which means that the model did not return any code as output to the prompt, was only the ninth most frequent error with seven cases in total (2.1%). These seven cases are divided among Llama2 (four times), BingChat (two times), and Bard (only once). As the name already implies, a "zero division error" (1.2%) happens when attempting to execute a division operation where the divisor is zero. Figure 6 displays all error categories that occurred per model. The height indicates the share of the error category (in %) of all incorrectly solved tasks per model. For comparison purposes, we also report the results of Code Llama when the missing indentations are not corrected. These cases are denoted as Code Llama* and shown next to the other models.

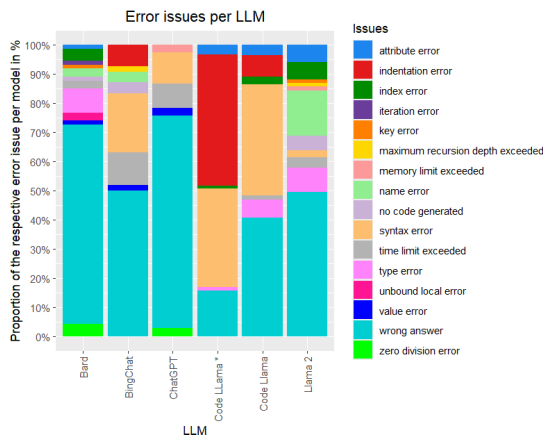


Figure 6: All error categories per LLM.

It is striking that the category "wrong answer" is the only one in the top three for all LLMs. Except for Code Llama*, it is the most common reason for errors as is to be expected from the results pre-

sented in Table 2. For Bard and both GPT-based models, there is the highest margin between this category and the runner-ups. We also observe that BingChat and ChatGPT do not differ concerning the three most frequent error categories and the order in which they appear.

If we now compare the two Code Llama approaches to each other, i.e. the one with added indentations (that was also used in the rest of our analysis) to the unaltered approach (in which we test the generated code without adaptation), clear differences are noticeable in the most frequent error categories. While the "indentation error" is in first place in the unaltered approach, it is only in third place after adaptation. As this error category has decreased, the number of incorrectly solved tasks due to "wrong answer" rises. The error category "syntax error", however, remains almost constant for both approaches.

6 Conclusion

We conducted an empirical study on five different LLMs, namely BingChat, ChatGPT, Bard, Llama2, and Code Llama, aiming at evaluating their potential for text-to-code generation. The mentioned LLMs were examined on 89 Python programming tasks from the coding website LeetCode. The results show clear differences between models and the number of code outputs they could generate correctly. In conclusion, ChatGPT outperformed all other models by a notable margin, followed by the second GPT-based model BingChat. Code Llama and Llama2 exhibited the lowest proficiency in solving tasks correctly, with Code Llama only surpassing Llama2 by correctly solving two additional tasks. Consequently, it cannot be affirmed that Code Llama, despite being a model specialized in coding-related tasks, significantly outperforms

its foundational model Llama2 in text-to-code generation. Moreover, although all the models were able to produce code in Python, a notable drawback of Code Llama is that the generated code is not indented, requiring the user to adjust the format of the code before it can be effectively deployed. Upon examining the correct solutions for all tasks, it can be stated that the code outputs generated by the LLMs are in many cases more efficient in terms of runtime and memory usage than human-written code solutions, assuming that almost all submissions on LeetCode are from humans. Regardless of the model, the length of the prompts seems to have an apparent influence on the likelihood of generating correct code. Since the prompts for correct solutions were shorter, it can be tentatively concluded that the prompts should be designed in a brief and targeted manner to achieve better results.

Moreover, it is worth noting that the models' failures stem to a large extent from code errors, categorized as "*wrong answer*", rather than due to supposedly simpler syntax errors. This is especially the case for Bard, BingChat, and ChatGPT. Considering the definition of the error category "*wrong answer*" as outlined in Section 5, it can be concluded that the three models might even demonstrate partial correctness in solving many tasks. Even these partially correct code outputs can be helpful for the user in subsequent applications, as incorporating LLMs has already advanced them closer to a solution. Users can then continue to work on the correct solution based on the generated code outputs. The outcomes indicate that ChatGPT, in particular, emerges as a useful assistant in terms of text-to-code generation. The impression gained from our study conducted is largely consistent with the findings of Geng et al. (2023). Additionally, the observed strong performance of ChatGPT is also evident in a study by Buscemi (2023), in which ChatGPT had to generate code in ten different programming languages on the same tasks. In the same study, the authors concluded that the model performed well in Python. Buscemi (2023) speculates on a possible reason that ChatGPT performs best in the programming languages in which it has received the most training signals. According to ChatGPT itself, Python is one of the top ten languages on which it has been trained. Nevertheless, considering the large number of wrong solutions, one should always look critically at the generated code, which still requires substantial knowledge of the programming language by the user.

While in the study the conversation between user and model consists only of one input and output, more extensive dialogues should be explored in future experiments. These dialogues could involve instructing the model to identify and fix errors in the previously generated code. Alternatively, they might encompass the execution of several related programming tasks in a single conversation, which requires the model to access previously generated code. Future studies can be extended to other programming languages or task types.

Ethical Considerations and Limitations

It is important to interpret the results with caution, as several limitations need to be considered. First, it is crucial to point out that the study conducted is only a snapshot of the performance of LLMs within the field of code generation at a certain point in time. The field of LLMs continues to move extremely rapidly, and new models and architectures may already outperform previous ones (Zhang et al., 2023a). Second, the study encompasses only 89 tasks and is thus limited in its scope. This has a direct influence on the interpretation and the generalizability of the results. The task volume affects the explanatory power and statistical robustness of the findings, with outcomes derived from a more comprehensive task set naturally having greater evidential weight. Third, we do not claim that our study is exhaustive, so as presented in Section 2 there are several other code-related tasks that LLMs can implement besides text-to-code generation. In addition, the scope of the work was also restricted to Python, which limits the applicability of the results to other programming languages. In terms of content, difficulty, and length, the selected tasks only cover a small subset of the huge spectrum of code-related tasks that can be passed to LLMs. Therefore, the tasks used do not embody the full bandwidth of challenges associated with text-to-code generation tasks, constraining the expressiveness of the study results to a narrow and focused set of tasks.

Concerning the reproducibility of the results, it is essential to acknowledge that the comparison values used in the runtime and memory usage analyses will change over time since users always submit new solutions to LeetCode. Therefore, these findings should rather be considered as a snapshot. When submitting the LLM-generated results to LeetCode, it was not possible to flag them as such

or to prevent them from being incorporated into the rankings. However, given the small number of solutions we uploaded, we do not think this will notably influence the LeetCode statistics. Furthermore, the reproducibility of results in the study is influenced by the fact that LLMs produce different answers even to identical prompts as inputs. This occurs because the models are partially updated and incorporate a certain degree of randomness in their responses. The generated outputs may differ from those obtained in the study conducted.

Acknowledgements

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of BERD@NFDI - grant number 460037581.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav

Petrov, and Yonghui Wu. 2023. [Palm 2 technical report](#).

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. [Program synthesis with large language models](#). *CoRR*, abs/2108.07732.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alessio Buscemi. 2023. [A Comparative Study of Code Generation using ChatGPT 3.5 across 10 Programming Languages](#).

Giuseppe Destefanis, Silvia Bartolucci, and Marco Ortu. 2023. [A preliminary analysis on the code generation capabilities of gpt-3.5 and bard ai models for java functions](#).

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. [CodeBERT: A pre-trained model for programming and natural languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, Online. Association for Computational Linguistics.

Chuqin Geng, Yihan Zhang, Brigitte Pientka, and Xujie Si. 2023. [Can ChatGPT Pass An Introductory Level Functional Language Programming Course?](#)

Daniel Jurafsky and James H Martin. 2023. [Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition](#). <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>. Accessed: 2023-12-03.

Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Meta. 2023. [Introducing CodeLlama, a state-of-the-art large language model for coding](#). <https://ai.meta.com/blog/code-llama-large-language-model-coding/>. Accessed: 2023-12-05.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. 2023. [Octopack: Instruction tuning code large language models](#).

OpenAI. 2022. [Chatgpt: Optimizing language models for dialogue](#).

R OpenAI. 2023. [Gpt-4 technical report](#). *arXiv*, pages 2303–08774.

- Stephen R. Piccolo, Paul Denny, Andrew Luxton-Reilly, Samuel Payne, and Perry G. Ridge. 2023. [Many bioinformatics programming tasks can be automated with chatgpt.](#)
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. [Code llama: Open foundation models for code.](#)
- Dominik Sobania, Martin Briesch, Carol Hanna, and Justyna Petke. 2023. [An analysis of the automatic bug fixing performance of chatgpt.](#)
- Weisong Sun, Chunrong Fang, Yudu You, Yun Miao, Yi Liu, Yuekang Li, Gelei Deng, Shenghan Huang, Yuchen Chen, Quanjun Zhang, Hanwei Qian, Yang Liu, and Zhenyu Chen. 2023. [Automatic code summarization via chatgpt: How far are we?](#)
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open Foundation and Fine-Tuned Chat Models.](#)
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention Is All You Need.](#) *CoRR*, abs/1706.03762.
- Dao Xuan-Quy, Le Ngoc-Bich, Phan Xuan-Dung, Ngo Bac-Bien, and Vo The-Duy. 2023. [Evaluation of chatgpt and microsoft bing ai chat performances on physics exams of vietnamese national high school graduation examination.](#)
- Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2023a. [Dive into Deep Learning.](#) *CoRR*, abs/2106.11342.
- Quanjun Zhang, Tongke Zhang, Juan Zhai, Chunrong Fang, Bowen Yu, Weisong Sun, and Zhenyu Chen. 2023b. [A critical review of large language model on software engineering: An example from chatgpt and automated program repair.](#)
- Ziyin Zhang, Chaoyu Chen, Bingchang Liu, Cong Liao, Zi Gong, Hang Yu, Jianguo Li, and Rui Wang. 2023c. [A survey on language models for code.](#)
- Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. 2020. [Progress in Neural NLP: Modeling, Learning, and Reasoning.](#) *Engineering*, 6(3):275–290.

A Leetcode tasks

```
1 You are given an m x n matrix M initialized with all 0's and an array of operations ops, where ops[i] =
2 [ai, bi] means M[x][y] should be incremented by one for all 0 <= x < ai and 0 <= y < bi.
3 Count and return the number of maximum integers in the matrix after performing all the operations.
4
5 Example 1:
6 Input: m = 3, n = 3, ops = [[2,2],[3,3]]
7 Output: 4
8 Explanation: The maximum integer in M is 2, and there are four of it in M. So return 4.
9
10 Example 2:
11 Input: m = 3, n = 3, ops = [[2,2],[3,3],[3,3],[3,3],[2,2],[3,3],[3,3],[3,3],[2,2],[3,3],[3,3],[3,3]]
12 Output: 4
13
14 Example 3:
15 Input: m = 3, n = 3, ops = []
16 Output: 9
17
18 Constraints:
19 1 <= m, n <= 4 * 10^4
20 0 <= ops.length <= 10^4
21 ops[i].length == 2
22 1 <= a_i <= m
23 1 <= b_i <= n
24
25 Write the code for this task in Python.
26 Start with:
27 class Solution (object):
28     def maxCount(self, m, n, ops):
```

Figure 7: Exemplary prompt based on LeetCode task number 598: Range Addition 2

B Exemplary Output

```
1 Sure, I can help you with that. Here's the code implementation for the given task:
2
3 python
4 class Solution(object):
5     def maxCount(self, m, n, ops):
6         if not ops:
7             return m * n
8
9         min_a = min(op[0] for op in ops)
10        min_b = min(op[1] for op in ops)
11
12        return min_a * min_b
13
14 # Example usage
15 solution = Solution()
16 print(solution.maxCount(3, 3, [[2, 2], [3, 3]])) # Output: 4
17 print(solution.maxCount(3, 3, [[2, 2], [3, 3], [3, 3], [3, 3], [2, 2], [3,
18 3], [3, 3], [3, 3], [2, 2], [3, 3], [3, 3], [3, 3]])) # Output: 4
19 print(solution.maxCount(3, 3, [])) # Output: 9
20
21 This code defines a Solution class with the maxCount method. It first checks if the ops list is
22 empty; if it is, then the entire matrix is incremented, and the maximum number of integers is
23 the total size of the matrix (m * n).
24
25 Otherwise, it calculates the minimum values of ai and bi among the given ops. Since these
26 are the dimensions within which the increments occur, the maximum number of integers will
27 be the product of these minimum values (min_a * min_b).
```

Figure 8: Output generated by ChatGPT to the prompt based on LeetCode task number 598: Range Addition 2

```
1 python
2 class Solution(object):
3     def divisorSubstrings(self, num, k):
4         def is_divisor(sub_num, divisor):
5             return int(sub_num) % divisor == 0
6
7         num_str = str(num)
8         num_length = len(num_str)
9         count = 0
10
11        for i in range(num_length - k + 1):
12            sub_num = num_str[i:i + k]
13            if sub_num[0] != '0': # Avoid leading zeros
14                divisor = int(sub_num)
15                if divisor != 0 and num % divisor == 0:
16                    count += 1
17
18        return count
```

Figure 9: Indented code in a code field generated by ChatGPT (based on LeetCode task number 2269: Find the K-Beauty of a Number)

```
1  [PYTHON]
2  class Solution:
3  def countBalls(self, lowLimit, highLimit):
4  max_count = 0
5  box_count = {}
6  for i in range(lowLimit, highLimit+1):
7  box_num = sum(map(int, str(i)))
8  if box_num in box_count:
9  box_count[box_num] += 1
10 else:
11 box_count[box_num] = 1
12 max_count = max(max_count, box_count[box_num])
13 return max_count
14 [/PYTHON]
```

Figure 10: Non-indented code in a text field generated by Code Llama (based on LeetCode task number 2269: Find the K-Beauty of a Number)

C Prompt Lengths

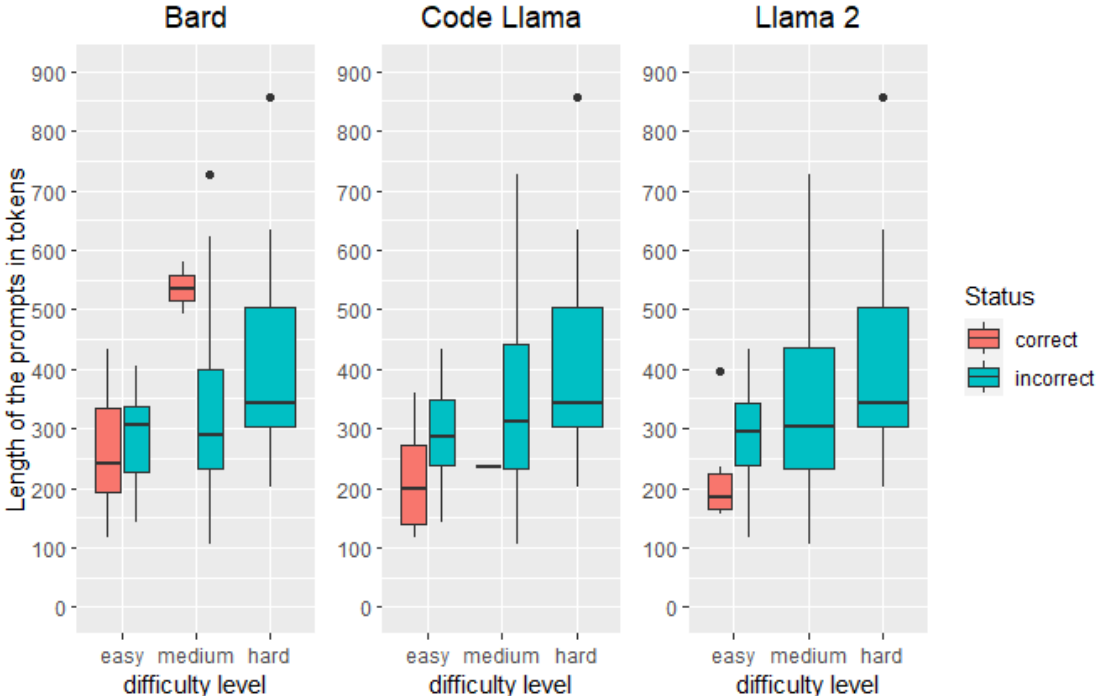


Figure 11: Distribution of prompt lengths of correct and incorrect solutions in Python for Bard, Code Llama, and Llama2

D Error Analysis

	Error type	Count	Share in % of all errors
1	wrong answer	178	54.3%
2	syntax error	48	14.6%
3	type error	18	5.5%
4	name error	17	5.2%
5	time limit exceeded	15	4.6%
6	indentation error	10	3.0%
7	index error	10	3.0%
8	attribute error	9	2.7%
9	no code generated	7	2.1%
10	zero division error	4	1.2%
11	value error	3	0.9%
12	key error	2	0.6%
13	maximum recursion depth exceeded	2	0.6%
14	memory limit exceeded	2	0.6%
15	unbound local error	2	0.6%
16	iteration error	1	0.3%

Table 3: All occurred error categories and their frequencies

E Correctly Solved Tasks

LLM	Level	Math	Matrix	Counting	Total	Total in percent
Bard	easy	3	5	6	14	46.67%
Bard	medium	1	1	0	2	6.67%
Bard	hard	0	0	0	0	0.00%
BingChat	easy	5	5	6	16	53.33%
BingChat	medium	4	5	3	12	40.00%
BingChat	hard	1	5	1	7	24.14%
ChatGPT	easy	7	6	7	20	66.67%
ChatGPT	medium	6	7	6	19	63.33%
ChatGPT	hard	6	5	2	13	44.83%
Code Llama	easy	2	2	3	7	23.33%
Code Llama	medium	1	0	0	1	3.33%
Code Llama	hard	0	0	0	0	0.00%
Llama2	easy	1	2	3	6	20.00%
Llama2	medium	0	0	0	0	0.00%
Llama2	hard	0	0	0	0	0.00%

Table 4: Distribution of correctly solved tasks in the study with Python (the levels easy and difficult consist of 30 tasks each and difficult of 29 tasks)

LLM	Level	Average runtime ranking	Average memory usage
Bard	easy	80.39%	58.87%
Bard	medium	80.46%	66.64%
Bard	hard	0.00%	0.00%
BingChat	easy	61.40%	63.38%
BingChat	medium	72.45%	54.70%
BingChat	hard	75.85%	80.98%
ChatGPT	easy	78.64%	54.86%
ChatGPT	medium	84.25%	56.19%
ChatGPT	hard	67.35%	59.31%
Code Llama	easy	65.28%	40.30%
Code Llama	medium	96.32%	98.42%
Code Llama	hard	0.00%	0.00%
Llama2	easy	75.59%	51.34%
Llama2	medium	0.00%	0.00%
Llama2	hard	0.00%	0.00%

Table 5: Average runtime and memory usage ranking per LLM and Level (corresponds to the plots 3 and 4)

F Absolute values for the runtime and memory usage

easy ID	Bard		BingChat		ChatGPT		Code Llama		Llama 2	
	ms	mb	ms	mb	ms	mb	ms	mb	ms	mb
598	-	-	39	15.48	53	15.44	-	-	-	-
1863	-	-	87	13.27	-	-	-	-	-	-
2409	-	-	-	-	6	13.31	-	-	-	-
2269	-	-	-	-	-	-	-	-	-	-
441	-	-	435	13.06	24	13.22	-	-	-	-
1742	282	16.17	-	-	313	15.94	858	15.94	-	-
2525	14	13.32	-	-	12	13.29	-	-	-	-
2119	-	-	19	13.33	19	13.24	-	-	13	13.14
231	20	13.37	22	13.16	9	13.26	18	13.17	-	-
2591	-	-	-	-	-	-	-	-	-	-
1380	93	13.63	86	13.70	101	13.47	-	-	-	-
1260	-	-	-	-	111	13.64	-	-	-	-
733	48	13.43	47	13.49	52	13.56	-	-	-	-
2500	-	-	-	-	-	-	-	-	-	-
2643	-	-	-	-	-	-	815	13.78	-	-
1030	103	15.85	-	-	113	16.66	-	-	101	15.81
2091	-	-	-	-	-	-	-	-	-	-
1351	-	-	94	14.36	86	14.37	88	14.50	92	14.55
1886	30	13.25	68	24.58	23	13.25	-	-	-	-
2319	198	14.28	197	14.24	-	-	-	-	-	-
1512	16	13.14	26	15.48	14	13.33	-	-	16	13.14
1897	31	13.57	45	13.27	32	13.74	-	-	-	-
1370	-	-	-	-	-	-	-	-	-	-
1356	-	-	-	-	52	13.53	49	13.53	-	-
2423	-	-	-	-	-	-	-	-	-	-
2351	12	13.26	10	13.30	-	-	4	13.36	10	13.36
1876	15	13.27	20	13.24	24	13.35	-	-	-	-
2053	-	-	47	13.33	42	13.67	-	-	-	-
1941	21	13.12	-	-	14	13.44	-	-	24	13.40
169	116	14.9	141	14.89	122	14.87	129	15.02	-	-

Table 6: Absolute values for runtime in milliseconds (ms) and memory usage in megabytes (mb) for each LLM and Level easy (column ID corresponds to the task number in LeetCode)

medium ID	Bard		BingChat		ChatGPT		Code Llama		Llama 2	
	ms	mb	ms	mb	ms	mb	ms	mb	ms	mb
2579	-	-	122	16.42	-	-	-	-	-	-
1017	-	-	-	-	12	13.28	-	-	-	-
319	-	-	16	13.22	16	13.43	6	13.07	-	-
523	-	-	781	33.40	745	33.46	-	-	-	-
2745	-	-	-	-	22	13.41	-	-	-	-
497	154	17.22	-	-	-	-	-	-	-	-
963	-	-	-	-	-	-	-	-	-	-
150	-	-	-	-	29	15.25	-	-	-	-
2063	-	-	-	-	-	-	-	-	-	-
478	-	-	93	24.76	104	24.90	-	-	-	-
861	-	-	-	-	-	-	-	-	-	-
427	-	-	-	-	86	15.34	-	-	-	-
1605	-	-	709	18.52	672	17.74	-	-	-	-
2684	-	-	1034	24.46	1386	22.74	-	-	-	-
2711	-	-	-	-	-	-	-	-	-	-
1895	-	-	3247	13.39	-	-	-	-	-	-
2482	-	-	-	-	1201	56.36	-	-	-	-
1536	-	-	412	14.01	398	13.75	-	-	-	-
1926	581	15.39	-	-	591	16.94	-	-	-	-
934	-	-	292	14.57	310	16.67	-	-	-	-
1497	-	-	-	-	44	25.38	-	-	-	-
1519	-	-	2482	180.85	1600	183.93	-	-	-	-
1010	-	-	242	16.55	205	16.51	-	-	-	-
2182	-	-	-	-	-	-	-	-	-	-
869	-	-	10	13.10	17	13.20	-	-	-	-
945	-	-	-	-	648	24.42	-	-	-	-
811	-	-	-	-	-	-	-	-	-	-
2170	-	-	-	-	-	-	-	-	-	-
1267	-	-	-	-	385	14.55	-	-	-	-
1775	-	-	-	-	-	-	-	-	-	-

Table 7: Absolute values for runtime in milliseconds (ms) and memory usage in megabytes (mb) for each LLM and Level medium (column ID corresponds to the task number in LeetCode)

hard ID	Bard		BingChat		ChatGPT		Code Llama		Llama 2	
	ms	mb	ms	mb	ms	mb	ms	mb	ms	mb
2019	-	-	-	-	1791	15.30	-	-	-	-
2584	-	-	-	-	-	-	-	-	-	-
2338	-	-	-	-	-	-	-	-	-	-
1835	-	-	-	-	681	25.84	-	-	-	-
1735	-	-	-	-	757	19.28	-	-	-	-
964	-	-	-	-	-	-	-	-	-	-
1467	-	-	-	-	-	-	-	-	-	-
2197	-	-	-	-	4131	27.74	-	-	-	-
1510	-	-	699	16.70	696	16.74	-	-	-	-
381	-	-	-	-	352	70.51	-	-	-	-
1074	-	-	-	-	-	-	-	-	-	-
212	-	-	-	-	7963	14.97	-	-	-	-
773	-	-	-	-	47	13.24	-	-	-	-
782	-	-	49	13.21	-	-	-	-	-	-
1970	-	-	-	-	-	-	-	-	-	-
2577	-	-	-	-	-	-	-	-	-	-
980	-	-	33	13.18	40	13.46	-	-	-	-
37	-	-	442	13.31	413	13.36	-	-	-	-
1293	-	-	49	14.74	-	-	-	-	-	-
827	-	-	2420	22.47	1935	22.67	-	-	-	-
2499	-	-	-	-	-	-	-	-	-	-
1857	-	-	-	-	2255	82.90	-	-	-	-
2014	-	-	-	-	-	-	-	-	-	-
2547	-	-	-	-	-	-	-	-	-	-
992	-	-	342	15.17	336	16.36	-	-	-	-
2025	-	-	-	-	-	-	-	-	-	-
2416	-	-	-	-	-	-	-	-	-	-
2514	-	-	-	-	-	-	-	-	-	-
1819	-	-	-	-	-	-	-	-	-	-

Table 8: Absolute values for runtime in milliseconds (ms) and memory usage in megabytes (mb) for each LLM and Level hard (column ID corresponds to the task number in LeetCode)

Classifying multilingual party manifestos: Domain transfer across country, time, and genre

Matthias ABenmacher^{1,2}, Nadja Sauter¹, Christian Heumann¹

¹Department of Statistics, LMU Munich, Germany,

²Munich Center for Machine Learning (MCML)

Correspondence: matthias@stat.uni-muenchen.de

Abstract

Annotating costs of large corpora are still one of the main bottlenecks in empirical social science research. On the one hand, making use of the capabilities of domain transfer allows re-using annotated data sets and trained models. On the other hand, it is not clear how well domain transfer works and how reliable the results are for transfer across different dimensions. We explore the potential of domain transfer across geographical locations, languages, time, and genre in a large-scale database of political manifestos. First, we show the strong within-domain classification performance of fine-tuned transformer models. Second, we vary the genre of the test set across the aforementioned dimensions to test for the fine-tuned models' robustness and transferability. For switching genres, we use an external corpus of transcribed speeches from New Zealand politicians while for the other three dimensions, custom splits of the Manifesto database are used. While BERT achieves the best scores in the initial experiments across modalities, DistilBERT proves to be competitive at a lower computational expense and is thus used for further experiments across time and country. The results of the additional analysis show that (Distil)BERT can be applied to future data with similar performance. Moreover, we observe (partly) notable differences between the political manifestos of different countries of origin, even if these countries share a language or a cultural background.

1 Introduction

Publishing party manifestos in the time frame leading up to an election is a common procedure in most parliamentary democracies around the globe. Summarizing the parties' political agendas for the upcoming electoral period, the published manifestos are intended to serve as guides for voters to reach their decision (Suiter and Farrell, 2011). Since the content of these manifestos also constitutes the foundation for the process of building

government coalitions, analyzing them can be very insightful. Janda et al. (1995), for instance, investigate the common assumption that political parties often try to change their images following a poor election result. Other researchers examine if parties learn from foreign successful parties (Böhmel et al., 2016). Tavits and Letki (2009) and Tsebelis (1999) also investigate their research questions based on political manifestos.

The Manifesto Project¹ covers programs of over 1000 political parties from more than 50 countries over a time frame from 1945 until today (Lehmann, 2022). The database provides access to the raw content of all documents as well as additional annotation for further analysis. Human annotators from over 50 different countries contributed by splitting the documents into quasi-sentences and subsequently classifying each of them according to a coding scheme covering 54 thematic categories. On a more coarse-grained scale, these 54 categories were further summarized into eight topics. Since manual annotation is extremely time and labor-intensive, requiring annotator training reliability, (partial) automation of the process could yield enormous potential for savings.

Our research explores how methods from the field of Natural Language Processing (NLP), which are more and more frequently used in social science research (Wankmüller, 2021), can be used to classify the quasi-sentences of the political manifestos into the eight topics of the Manifesto coding scheme. Therefore, different NLP methods, namely TF-IDF + logistic regression (LR) as a comparative baseline (cf. Osnabrügge et al. (2023)) and different monolingual and multilingual versions of BERT (Devlin et al., 2019) are used to process and subsequently classify the sequences. In the following, first, the related work (cf. Sec. 2.1) and the data extraction process (cf. Sec. 2.2)

¹<https://manifesto-project.wzb.eu/>

will be explained in further detail followed by the experimental setup (cf. Sec. 3), where we delve deeper into the concept of cross-domain classification and motivate the different cross-domain scenarios. The predictive performances of each evaluated model for each of the different scenarios are compared and discussed in Section 4. We conclude the experiments by fine-tuning a multilingual model on the whole corpus.

Contribution: Our main contributions can be summarized as follows: We extend the cross-domain setting introduced by Osnabrügge et al. (2023) along multiple axes. We not only measure transfer across genre (manifestos \rightarrow speeches) but also across time (2018 \rightarrow 2022) and country (leave-one-country-out, LOCO). Instead of relying on simple machine learning classifiers, we fine-tune pre-trained language models (Devlin et al., 2019; Sanh et al., 2019) achieving superior performance to simple models. We don't only rely on English texts, but leverage the whole Manifesto database by employing multilingual pre-trained models. This enables us to train one single model which can be used for all languages and countries. The code for our experiments and the trained models will be publicly available to nurture further research: <https://github.com/slds-lmu/manifesto-domaintransfer> (code) and <https://huggingface.co/assenmacher> (models).

2 Materials and Methods

2.1 Related work

We draw inspiration for our work from the research article "Cross-Domain Topic Classification for Political Texts" (Osnabrügge et al., 2023). The authors employ supervised machine learning (logistic regression, LR) alongside feature engineering techniques for text (TF-IDF w/ n-grams) for the classification of political manifestos and speeches. The analysis was performed on two (labeled) data sets, where each utterance was assigned one of the eight possible categories "freedom and democracy", "fabric of society", "economy", "political system", "welfare and quality of life", "social groups", "external relations" and "no topic". The *source corpus* consists of manifestos, collected between 1984 and 2018, which were extracted from the Manifesto Project (Krause et al., 2018) for the following seven English-speaking countries: Australia, Canada, Ireland, New Zealand, South Africa, the

UK, and the USA. Each document was split into quasi-sentences ($n_{source} = 115,410$) and then labeled by a trained human annotator from the Manifesto Project. In most cases, one quasi-sentence roughly equals one sentence, however, some long sentences containing several statements were split into multiple quasi-sentences. Osnabrügge et al. (2023) use this *source corpus* for training and for measuring the within-domain performance. The *target corpus* ($n_{target} = 4,165$), consists of English speeches held by members of the New Zealand Parliament in the time period from 1987 to 2002. The speeches were extracted from the official record of the New Zealand Parliament (Hansard), and manually annotated according to the same schema by Osnabrügge et al. (2023), who then use it for measuring the cross-domain classification performance.

After the hyperparameter tuning using grid search, they achieved an accuracy of 0.641 on the held-out set of the source corpus and an accuracy of 0.507 on the speeches, showing that cross-domain classification is a reasonable approach. Additionally, the authors create their own, more fine-grained, coding scheme with 44 topic categories for which they report lower performance values for both the within- (0.538) and the cross-domain (0.410) setting. It is important to note, that our performance scores are not perfectly comparable to Osnabrügge et al. (2023), since we download the data ourselves (with slight differences, cf. Sec. 2.2) and thus have a different train/validation/test split.

2.2 Data extraction from Manifesto Project

For conducting the experiments described in Sec. 3, we extract the manifestos ourselves from the Manifesto Project database using its dedicated R-package *manifestoR* (Lewandowski et al., 2020). Thus, as opposed to Osnabrügge et al. (2023), our corpus also includes additional information on the year and country of origin for each utterance. Our data sets include the 2018-2 version of the corpus (Krause et al., 2018), similar to Osnabrügge et al. (2023), as well as the most recent version (2022-1, Lehmann et al., 2022), resulting in $n_{2018,en} = 114,523$ for the seven English-speaking countries mentioned in Sec. 2.1 and $n_{2018,all} = 996,008$ in total. For the 2022 corpus, there are in total 158,601 English observations and 1,504,721 for all languages, respectively. Among those, $n_{2022,en} = 27,764$ observations from the period between 2019 and 2022 constitute our test set for the experiments across time for

the English language. We observe a difference of 887 samples between the data from [Osnabrügge et al. \(2023\)](#) ($n_{source} = 115,410$) and our data set ($n_{2018,en} = 114,523$), which is probably due to potential changes in the 2018 version the database.

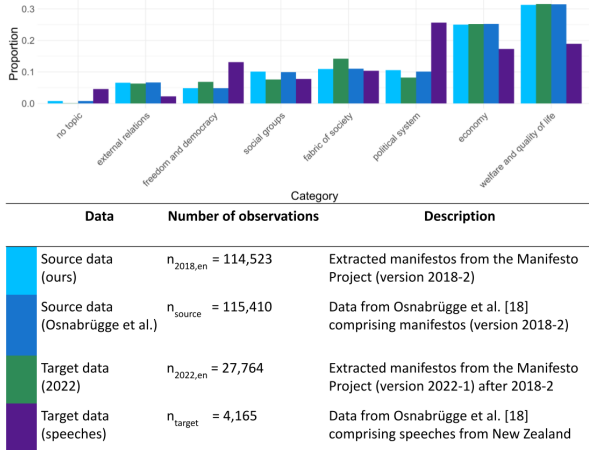


Figure 1: Label distributions for the four different corpora alongside sample sizes and short descriptions.

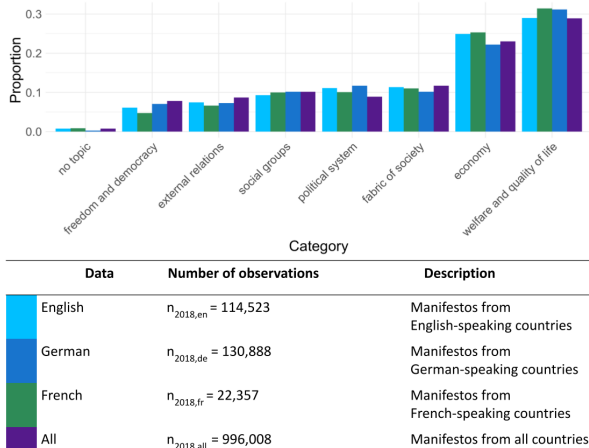


Figure 2: Label distributions for the three most frequent languages and overall in the 2018-2 corpus alongside sample sizes and short descriptions.

Figure 1 visualizes the different label distributions for (a) the source corpus of [Osnabrügge et al. \(2023\)](#), (b) our extraction of the 2018-2 corpus, (c) our extraction of the 2022-1 corpus, and (d) the target corpus of the New Zealand speeches ([Osnabrügge et al., 2023](#)). While the former three roughly follow the same distribution, with about 57% of the observations assigned to either "*welfare and quality of life*" or "*economy*", the most common class of the latter is "*political system*" ($\sim 26\%$) followed by "*welfare and quality of life*" ($\sim 19\%$). Thus, the two main challenges aside from

the domain transfer are the overall class imbalance as well as the differences between the source and target domain with respect to the label distribution. Further Figure 2 shows the distribution of the target classes separated by the language the manifestos are written in. We display the three most frequent languages, which we use for conducting experiments across country (cf. Sec. 3.1), against the distribution in the entire 2018-2 corpus of all manifestos. Here we observe some minor differences, as "*welfare and quality of life*" and "*political system*" are more frequently addressed in German-speaking countries (compared to the overall corpus), "*welfare and quality of life*" and "*economy*" in French-speaking ones, and "*political system*" and "*economy*" in English-speaking ones. Notably, for all three languages, the topics "*freedom and democracy*" and "*external relations*" are addressed less often than in the whole 2018-2 corpus.

3 Experimental Setup

In this section, we introduce the concept of domain transfer in general and in particular the cross-domain classification settings for our application. Further, the methodological background for the employed model architectures will be laid out as follows: First, we briefly review common feature engineering techniques for text data and elaborate on the advantages and disadvantages. These techniques include term-frequency inverse-document-frequency (TF-IDF) weighting, as well as dense word or document embeddings. Second, we introduce two state-of-the-art NLP architectures that we employ in our analysis, namely BERT ([Devlin et al., 2019](#)) and DistilBERT ([Sanh et al., 2019](#)), both of which do not require prior feature engineering steps but accommodate the whole pipeline in one single model. Finally, we briefly sketch the individual experiments which were carried out over the course of this study.

3.1 Cross-Domain Classification

When talking about *classification* in the context of machine learning, researchers commonly implicitly refer to within-domain/within-distribution classification, implying that the trained model is tested on data from the same origin/distribution as the training data (i.e. the *source domain*). Cross-domain classification, on the other hand, explicitly considers a shift in the domain/distribution/source of the

Scenario	Data set characteristic		Data set splitting		Data set sizes	
	Corpus	Language(s)	Training set	Test set	Training set	Test set
within-domain	2018-2	En, De, Fr	random split ^a	random split ^b	91,618 / 104,710 / 17,885	11,452 / 13,089 / 2,236
manifestos → speeches 2018 → 2022	2018-2	En	random split ^a	speeches	91,618	4,165
	2018-2	En, De, Fr	random split ^a	future ^c	91,618 / 104,710 / 17,885	27,764 / 30,542 / 343
across country	2018-2	En, De, Fr	$n - 1$ countries	held-out country	— ^d	— ^d
Multilingual	2018-2	38 languages	random split ^a	random split ^b	796,806	99,601

^a Here: .8/.1/.1, i.e. 80% of the 2018-2 data.

^b Here: .8/.1/.1, i.e. 10% of the 2018-2 data.

^c "future": data from the 2022-1 corpus recorded after the 2018-2 cut-off.

^d Different scenarios, test set contains one single country in each experiment.

Table 1: Overview of the investigated cross-domain scenarios, alongside the used corpora, test sets, and languages.

data, i.e. the data-generating process is assumed to be different. Frequently examined cases of domain shift in NLP include a change in language (i.e. training the model on text from one language and evaluating it in another, cf. [Conneau et al. \(2018, 2019\)](#)), topic (e.g. training the model on reviews on restaurants and evaluation it on reviews on laptops, cf. [Pontiki et al. \(2014\)](#)) or genre (e.g. training on texts and evaluation on transcribed audio data, cf. [Osnabrügge et al. \(2023\)](#)). In our experiments, we contribute to this body of research by considering the following different cross-domain settings:

Transfer across genre: We consider party manifestos from all seven (English-speaking) countries as our source corpus $C_{source} = C_{2018,en}$ and evaluate the trained model on a target corpus C_{target} of transcribed parliamentary speeches from New Zealand. This setting is equivalent to the work of [Osnabrügge et al. \(2023\)](#), yet we rely on more elaborated model architectures.

Transfer across time: We use the party manifestos from all countries for all years up until 2018 as source corpus C_{source}^2 , while the target corpus C_{target} consists of party manifestos from the year 2019 – 2022. This setting is intended to test the temporal robustness of the fine-tuned models.

Transfer across country: This setup comprises three distinct experiments for different languages (English, German, French), for each of which we include data from all³ countries, where manifestos in the given language exist in the 2018-2 corpus. The setting for each language consists again of seven, five, and four, respectively, different individual experiments, since for each language we include all but one country as source corpus C_{source} and evaluate the model on a target corpus C_{target} including

² C_{source} is either $C_{2018,en}$, $C_{2018,de}$ or $C_{2018,fr}$

³For English we excluded countries with a low n , to stay consistent with [Osnabrügge et al. \(2023\)](#).

only the manifestos from the single held-out country. Further, we also inspect a true multilingual model trained on data from all available countries.

Metrics and Training We compare our results, which we measure in terms of Accuracy and Macro-F1 Score, from the cross-domain experiments to the performance we obtain for the within-domain setting. We opt for reporting the macro-averaged version of the F1 Score in order to take into account the class imbalance (cf. Fig. 1). For model training, we conduct a train/validation/test split with proportions .8/.1/.1; all reported performance values are measured on the test set. Note that, depending on the cross-domain setting, also different test sets than the random split are used. Table 1 summarizes the different investigated scenarios in a comprehensive manner, provides an overview of the respectively used corpora for training and evaluation, and specifies with which procedure the respective test sets were created or selected.

3.2 Model architectures

Early feature engineering techniques relying on the bag-of-words (BoW) assumption have in recent years been replaced by more elaborated representation learning algorithms. BoW refers to counting the occurrences of words (or n-grams) in a document and representing it as V -dimensional vector, where V is the vocabulary size. This representation can be enhanced via TF-IDF, as done by [Osnabrügge et al. \(2023\)](#), via a re-weighting using corpus-level occurrence statistics.

With the advent of representation learning, it became possible to represent words ([Mikolov et al., 2013](#); [Pennington et al., 2014](#); [Bojanowski et al., 2016](#)) and documents ([Le and Mikolov, 2014](#)) by dense vectors of a comparably low, fixed dimensionality. These representations were used in a similar fashion in conjunction with a classifier as BoW-based representations. BERT ([Devlin et al.,](#)

	within-domain		manifestos \rightarrow speeches		2018 \rightarrow 2022	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
TF-IDF + LR	0.6413	0.5195	0.5059 (\downarrow 0.1354)	0.4474 (\downarrow 0.0586)	–	–
English BERT	0.6977	0.5841	0.5613 (\downarrow 0.1364)	0.5046 (\downarrow 0.0795)	0.6841 (\downarrow 0.0136)	0.5707 (\downarrow 0.0134)
English DistilBERT	0.6866	0.5694	0.5669 (\downarrow 0.1197)	0.5026 (\downarrow 0.0568)	0.6784 (\downarrow 0.0082)	0.5620 (\downarrow 0.0074)
German DistilBERT	0.6583	0.5628	–	–	0.6559 (\downarrow 0.0024)	0.5485 (\downarrow 0.0143)
FlauBERT	0.6087	0.5159	–	–	0.6093 (\uparrow 0.0006)	0.4783 (\downarrow 0.0376)
Multilingual DistilBERT	0.6748	0.5941	–	–	0.6311 (\downarrow 0.0437)	0.5278 (\downarrow 0.0663)

Table 2: Performance values of TF-IDF + LR (Osnabrügge et al., 2023) versus English BERT and DistilBERT models (upper part) as well as for German DistilBERT and French FlauBERT models (middle part) and the multilingual DistilBERT model (lower part). Absolute change vs. within-domain performance in parentheses.

2019) enabled the coupling of these two steps, i.e. it provided one single end-to-end trainable model for learning (contextual) representations and training the classifier. The commonality of BERT and all subsequent architectures is that they all are relying on the Transformer architecture (Vaswani et al., 2017). Based on BERT, DistilBERT models can be trained using model distillation (Buciluă et al., 2006; Hinton et al., 2015), a training process during which the smaller student model (DistilBERT) is trained to mimic the larger teacher model’s (BERT) behavior. In the case of DistilBERT, the student model, while having half the size of its teacher model, is able to retain approximately 95% of the teacher model’s performance on the GLUE benchmark (Sanh et al., 2019).

We use bert-base-cased as well as distilbert-base-cased for English. For further experiments, we employ distilbert-base-german-cased, flaubert_small_cased (as no French DistilBERT is available) and distilbert-base-multilingual-cased.

3.3 Experiments

In the first step, we stick to the setup from Osnabrügge et al. (2023), extracting similar data, re-running their experiments, and comparing against their LR+TF-IDF baseline. We further compare the performance of BERT against the cheaper DistilBERT for the English within-domain setting and the English cross-domain settings (manifestos \rightarrow speeches, 2018 \rightarrow 2022, and across country) to assess the competitiveness of the latter one. For the cross-domain scenarios in the other languages (German, French) we thereafter conduct all experiments with DistilBERT, since it is the cheaper model. The concluding multilingual experiments on the complete corpus are also conducted us-

ing a DistilBERT model, fine-tuning the model on the train set of a random split of *the whole* 2018-2 data set. The code for all our experiments is available on GitHub: <https://github.com/slds-lmu/manifesto-domaintransfer>.

4 Results

This section will be structured as follows: First, we will show the superior within-domain performance of pre-trained BERT-based models over the simple baseline from Osnabrügge et al. (2023) and will closely inspect the per-class within-domain performances of the different models. In conjunction with this, we also compare our models to Osnabrügge et al. (2023) on the manifestos \rightarrow speeches scenario, since we adopt it from their work. This scenario we can, however, only inspect for the English language as the corpus of speeches is from New Zealand. Second, we will verify if and how well experiments across genre and time work for the different monolingual models and the multilingual one. Third, we inspect closely how well performance can be transferred across different countries speaking the same language. Subsequently, we delve deeper into a truly multilingual by fine-tuning a pre-trained multilingual model on the entirety of the corpus and examining its performance for the different countries and languages.

Within-domain performance The results of our experiments comparing different models for within-domain classification, manifestos \rightarrow speeches, and 2018 \rightarrow 2022 classification are presented in Table 2. For within-domain classification, the TF-IDF + LR model is clearly outperformed by the deep learning models, where the English models perform better than the German, French, and Multilingual ones. It is notable that in general, the French model exhibits

	English			German			French			Multilingual		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
No Topic	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4142	0.1394	0.2086
Freedom / Democracy	0.6258	0.5318	0.5750	0.6631	0.6133	0.6372	0.6533	0.5868	0.6183	0.6165	0.5787	0.5970
External Relations	0.7395	0.7517	<u>0.7456</u>	0.7429	<u>0.7067</u>	0.7243	0.6688	<u>0.6913</u>	0.6799	0.7357	0.7068	<u>0.7209</u>
Social Groups	0.5794	0.5488	0.5637	0.6040	0.5370	0.5685	0.6034	0.4506	0.5160	0.6242	0.5372	0.5774
Political System	0.5629	0.4773	0.5166	0.6088	0.5145	0.5577	0.4407	0.5372	0.4842	0.6012	0.5646	0.5823
Fabric of Society	0.6463	0.6727	0.6592	0.5909	0.6496	0.6189	0.5485	0.4837	0.5140	0.6212	0.6092	0.6151
Economy	0.7269	<u>0.7570</u>	0.7416	<u>0.6882</u>	0.7009	0.6945	0.6270	0.6449	0.6358	0.6934	<u>0.7449</u>	0.7182
Welfare / Quality of Life	<u>0.7293</u>	0.7793	0.7534	0.6686	0.7379	<u>0.7015</u>	<u>0.6604</u>	0.6990	<u>0.6791</u>	<u>0.7151</u>	0.7517	0.7330

Table 3: A detailed performance report for per-class within-domain performance, measured in terms of Precision (P), Recall (R), and F1 Score, for the DistilBERT models in English and German, the French FlauBERT as well as for the multilingual DistilBERT. Best scores (per language) in **bold**, runner-up underlined.

rather low performance values⁴ (within-domain as well as across time) compared to all other models, which may for one reason be caused by the relatively small corpus size for this language compared to all other ones (cf. Tab. 1). We also observe the expectedly higher performance of the English BERT model compared to the English DistilBERT, since it generally outperforms DistilBERT in all scenarios except for the accuracy in *manifesto* → *speeches* transfer. However, the performance gaps between these two models are rather small, which very well justifies the use of DistilBERT for the remainder of the experiments, trading some performance for saving computational expenses.⁵

When further considering the predictive performance separately for each of the eight classes (cf. Tab. 3), we learn that for none of the languages and for none of the investigated scenarios any of the monolingual DistilBERT models was able to predict a single case of the highly underrepresented "no topic" class. The obvious reasons for this are the low number of observations as well as the potential ambiguity, heterogeneity, and fuzziness of the manifestos that could not even by the human annotators be classified into one coherent class but were assigned to this collection basin. This peculiarity of the results should always be taken into account when interpreting them since the macro-averaged F1 Score tends to be a rather conservative performance measure as it weighs the performance of this class similarly to all other classes. This also largely explains the quite notable gap between the Accuracies and Macro-F1 Scores (cf. Tab. 2).

The largest class (in terms of the number of ob-

⁴Note, that cannot be compared to the English TF-IDF + LR baseline due to different training and test sets.

⁵While training BERT for one epoch took roughly 1h 11 minutes, DistilBERT nearly halved this training time per epoch to about 38 minutes. Adding this up over three epochs amounts to time savings of nearly 100 minutes.

servations) was easiest to classify for the DistilBERT models across all languages, i.e. for "*welfare and quality of life*" overall the highest values in *P*, *R*, and *F1* are observed. Interestingly it is not the second largest class ("*economy*") where the models perform next best, but rather one of the smallest classes ("*external relations*"), which is nicely visualized by the highlighting in Table 3. Nevertheless, the models are capable of predicting also the "*economy*" class quite well. Further, it is interesting to observe that for the classes exhibiting high F1 Scores, the gap between recall and precision is (a) rather small and (b) sometimes even in favor of the recall, while for the low-performance classes, the recall often appears to be notably worse than the precision. This is especially consistently observable for the class "*social groups*".

When compared to the monolingual models, the multilingual one stands out due to two distinct reasons (cf. Tab. 3): First, it is the only one of the four models to detect at least *any* true "no topic" observations in its test set. Although the performance for this particular class still is not great, it still seems as if learning from more (and more diverse) data seems to help in this respect. Second, and probably also related to the first advantage, the performance seems to be more stable when comparing the scores across the different classes. While for the other English and French, the ranges (excluding "no topic") of the F1 Score were 0.2290, and 0.1957 respectively, this metric is comparably small, with a value of only 0.1556, similar to 0.1666 for the German language.

Transfer across genre and time Inspecting the two cross-domain settings in Table 2 more closely, we see that transfer across the temporal axis works better than across the genre axis. While for the English DistilBERT model the performance on the

New Zealand speeches drops by quite a margin ($\downarrow 0.1197 / \downarrow 0.0568$), it merely changes when evaluated on the data from a different time period ($\downarrow 0.0082 / \downarrow 0.0074$). Again, comparing BERT to DistilBERT, the latter even seems to be more stable over time since the performance decrease is slightly less pronounced. For the cross-modal transfer scenario, we provide the confusion matrix (cf. Fig. 4 in Appendix A) to enable further error analysis. While the two most frequent classes are still very accurately predicted, the model severely struggles when it comes to distinguishing many of the other classes from the "political system" category. Even for the two largest classes, a notable amount of the instances were misclassified into this category. Further, the model's error of confusing a certain category with "political system" is even worse for the smaller classes, e.g. "freedom and democracy", with fewer samples.

While this comparison of the scenarios across genre and across time can not be made for the other languages and the multilingual scenario, we also observe only very minor drops in performance for the latter scenario there. For the two monolingual models, we record decreases for accuracy of 0.24 percentage points for the German model and even no decrease at all for the accuracy of the French DistilBERT model, as well as decreases of 1.43 (German) and 3.76 (French) percentage points for Macro-F1. The multilingual model, however, exhibits somewhat larger drops in performance of 4.37 percentage points for accuracy and 6.63 percentage points for Macro-F1, respectively.

Transfer across countries The results of our LOCO experiments using the monolingual DistilBERT models for English and German, and a FlauBERT model for French, are presented in Table 4. We support the results by visualizations (cf. Fig. 3) of how the performance on manifestos from a certain country changes depending on whether we (a) evaluate on its portion of the random test split or (b) on all manifestos of this country as a hold-out set. The most important takeaway from these illustrations is the fact that completely withholding data from a certain country hurts model performance on data from this specific country, but not in equal parts for the different languages. For German-speaking countries (cf. Fig. 3, middle) the decrease from left to right is less pronounced than for the other two languages (Fig. 3, top/bottom).

The overall takeaway from the previous experi-

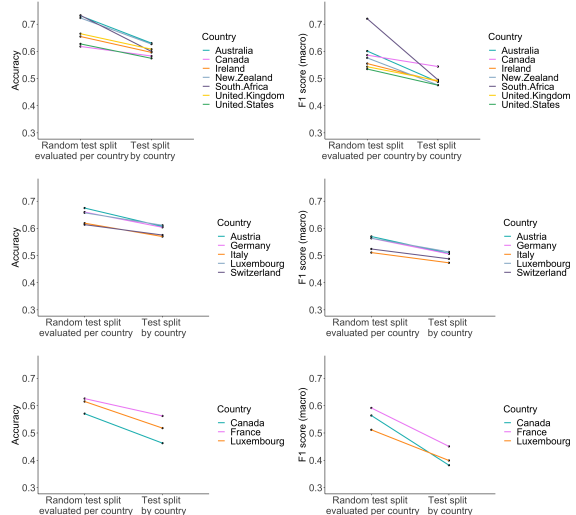


Figure 3: Comparison of the performance on data from specific English- (top), German- (middle), and French-speaking (bottom) countries via the Accuracy (left) and Macro-F1. On the left-hand side of each subfigure, performance is measured on the portion of each country in the random test set, while on the right side, the country-specific LOCO performance is displayed. Lines are drawn between the respective points to visualize the connection within one country. Switzerland is excluded, since there is only one sample in the random test split.

ments (better performance for English) is not entirely confirmed by these results, also showing a much more nuanced picture regarding interesting inter-country differences per language. For the LOCO scenario within the English-speaking countries, Australia and New Zealand exhibit the highest values for accuracy, while South Africa and Canada outperform the other with respect to Macro-F1⁶. The two European countries and the United States overall show the worst performance with respect to both metrics. Further, it is worth noting that there is a rather high variation among these performance values compared to German and French. Excluding the "no topic" class, the values for accuracy exhibit a range of 0.0560, while the Macro-F1 Score has a range of 0.0686. On a final note, it is interesting to see that the performance on New Zealand manifestos is among the top-ranking countries in accuracy, while the domain transfer across modalities (to New Zealand parliamentary speeches) shows a little bit of a performance decrease.

The German LOCO classification experiments using DistilBERT exhibit somewhat different results compared to the English experiments. While

⁶Canada has better Macro-F1 Scores than most other countries (except for the top two), but comparably low accuracy.

			English-LOCO (DistilBERT)		German-LOCO (DistilBERT)		French-LOCO(FlauBERT)	
	n_{random}	$n_{country}$	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Australia	1,861	18,480	0.6304	0.4877	–	–	–	–
Canada	322	3,047	0.5829	0.5441	–	–	–	–
Ireland	2,548	25,357	0.5962	0.4895	–	–	–	–
New Zealand	2,840	28,561	<u>0.6268</u>	0.4761	–	–	–	–
South Africa	628	6,423	0.5997	<u>0.4954</u>	–	–	–	–
United Kingdom	2,182	21,836	0.6080	0.4924	–	–	–	–
United States	1,071	10,819	0.5744	0.4755	–	–	–	–
Austria	3,361	33,818	–	–	<u>0.6071</u>	<u>0.5077</u>	–	–
Germany	6,452	63,413	–	–	0.6039	0.5060	–	–
Italy	63	651	–	–	0.5699	0.4733	–	–
Luxembourg	1,850	19,291	–	–	0.6114	0.5134	–	–
Switzerland	1,390	13,715	–	–	0.5754	0.4878	–	–
Canada	517	5,386	–	–	–	–	0.4629	0.3822
France	850	8,290	–	–	–	–	<u>0.5624</u>	<u>0.4511</u>
Luxembourg	868	8,662	–	–	–	–	0.5179	0.3993
Switzerland	1	19	–	–	–	–	0.7368	0.7288
Average			0.6026	0.4944	0.5935	0.4976	0.5700	0.4904

Table 4: LOCO performance for English (7 countries), German (5 countries), and French (4 countries). Best scores per language in **bold**, runner-up underlined. We report both n_{random} for the number of observations in the random test split and $n_{country}$ for the number of observations when the respective country is used as held-out set.

the overall averages are comparable, the ranges (0.0415 for accuracy and 0.0344 for Macro-F1) indicate that the values for all countries are relatively similar, with Luxembourg having the highest accuracy of 0.6114 as well as the highest Macro-F1 Score of 0.5134. We speculate that the reason for this observation might lie (a) in the similarity of the political systems⁷ of all these countries and (b) in their geographical and cultural closeness. However, being no experts in political science, we would leave the definite interpretation of such matters to those. Regarding the overall performance, the German model performs no worse than the English model(s) which was not necessarily to be expected due to our conclusions drawn from Tables 2 and 3.

A rather distinct picture emerges when inspecting the results for the French LOCO classification (still bearing in mind that the performance estimates for Switzerland, with only 19 observations, might make the interpretations rather unreliable). The range for accuracy is 0.2739 and 0.3466 for Macro-F1, which is notably larger than the ranges for both the English-speaking countries and the German-speaking countries. Switzerland exhibits by far the highest values, but it should again be noted that they are based on only 19 observations.

⁷Despite Luxembourg being a parliamentary monarchy, the country still has a similar landscape of political parties compared to its neighbors, including i.a. social and Christian democrats, liberals, a Green party, as well as different smaller left- and right-wing parties.

The average values are comparable, although a bit lower, to the other two languages, but again strongly influenced by the seemingly strong performance on Swiss manifestos. Regarding the other three countries, France itself stands out from the other two, exhibiting both the highest accuracy as well as the highest Macro-F1 Score among them.

5 Conclusion and Future Work

We showed in a series of extensive experiments that domain transfer along three different axes (genre, time, country) in principle works for this sort of political text. We observed the largest performance drops when attempting to generalize across modalities, however, the models tend to generalize very well across time. While the first finding might be foreseeable, the latter result is insofar exciting since after the time point we chose for splitting the data (2018) quite some new topics, e.g. the global COVID-19 pandemic or the Ukrainian war, emerged. Regarding the generalization across country, even within languages (and hence to some extent also cultural backgrounds), there seem to be notable differences between the political communication in the different countries as observed by the large performance differences. To conclude, we can state that a true multilingual approach towards classifying political text looks promising, yielding good and stable performance across numerous countries with different languages.

Interesting starting points for future work are obviously to examine the capacities of the emerging ever more powerful LLMs to tackle challenging tasks like this and to make use of the continuously extending data pool from the Manifesto project. Since new countries and time points are added constantly, there is definitely the potential to extend our work in future research.

Limitations

The advent of large language models (LLMs), in particular ChatGPT (OpenAI, 2022; Bubeck et al., 2023), resulted in a paradigm change in NLP research. Since then, we can loosely categorize existing and newly introduced classification models into several bins: "pre-train/fine-tune", "prompting", and "chatting". While "pre-train/fine-tune" has been (and still widely is) the pre-dominant research paradigm in applied NLP research since \sim 2018, "prompting" has upon the introduction of GPT-3 (Brown et al., 2020) become an exciting approach for tackling (a) multi-task learning and (b) low-resource scenarios via few-/zero-shot learning. Further, accessing a model via prompting might be considered more "human-like" / "natural" than training a model on class labels via gradient descent.

On the other hand, there are still also numerous reasons not to abandon architectures relying on the "pre-train/fine-tune" paradigm (Yang et al., 2023), several of which we consider fulfilled as far as our research question is concerned. First, given the large, annotated training corpus there is no need to rely on few-shot learning but rather to use all of the available data points to achieve maximum model performance. Prompting models would struggle with this amount of data due to context length constraints. Second, given the very custom-defined label set of political topics for this political corpus, for general-purpose prompting models, this label set would always have to be in some way appended to the prompt for the model to be informed about the granularity in the first place. On the one hand, this would probably lead to the model struggling with learning the underlying concepts, on the other hand, it would lead to better adaptive capabilities in case the granularity changes. Third, for domain-specific research questions like this, it might not always be feasible for researchers to access the computational resources for running or prompting such large models, and hence a task-

specific, parameter-efficient model that does the trick equally well might be preferable.

We further acknowledge that the performance could potentially still be increased using more elaborate models following the "pre-train/fine-tune" paradigm, e.g. variants of the T5 model family (Raffel et al., 2020; Xue et al., 2020). Using these models, however, comes at the cost of a higher computational expense potentially requiring much more VRAM than the average practitioner has access to. The models we employ can, on the other hand, be fine-tuned comfortably using smaller GPUs with around 16GB of VRAM in an acceptable amount of time. Given the ever-increasing model sizes and thus also the computational requirements, this is an important issue to keep an eye on.

Ethical considerations

To the best of our knowledge, no ethical considerations are implied by our work. The only aspect that is affected in a broader sense is the environmental impact of the computationally expensive experiments. This issue naturally comes with pre-training large language models and is obviously a concern that has to be expressed in every work dealing with this sort of model. But on the other hand, our work rather works against increasing the environmental impact, since we "only" focus on reusing existing pre-trained models and performing the cheap(er) fine-tuning step. Further, we also provide access to our fine-tuned models which can be used by other researchers.

Acknowledgements

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of BERD@NFDI - grant number 460037581.

References

- Tobias Böhmelt, Lawrence Ezrow, Roni Lehrer, and Hugh Ward. 2016. Party policy diffusion. *American Political Science Review*, 110(2):397–410.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. arxiv 2016. *arXiv preprint arXiv:1607.04606*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot

- learners. *Advances in neural information processing systems*, 33:1877–1901.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrike, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).
- Kenneth Janda, Robert Harmel, Christine Edens, and Patricia Goff. 1995. **Changes in party identity: Evidence from party manifestos**. *Party Politics*, 1(2):171–196.
- Werner Krause, Pola Lehmann, Jirka Lewandowski, Theres Matthieß, Nicolas Merz, and Sven Regel. 2018. Manifesto Corpus, Version: 2018-2. *Berlin: WZB Berlin Social Science Center*.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Pola Lehmann. 2022. **Manifesto project**. Accessed: 2022-10-01.
- Pola Lehmann, Tobias Burst, Jirka Lewandowski, Theres Matthieß, Sven Regel, and Lisa Zehnter. 2022. Manifesto Corpus. Version: 2022-1. *Berlin: WZB Berlin Social Science Center*.
- Jirka Lewandowski, Nicolas Merz, and Sven Regel. 2020. **manifestoR: Access and Process Data and Documents of the Manifesto Project**. R package version 1.5.0.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- OpenAI. 2022. **Chatgpt: Optimizing language models for dialogue**. Accessed: 2023-01-10.
- Moritz Osnabrügge, Elliott Ash, and Massimo Morelli. 2023. Cross-domain topic classification for political texts. *Political Analysis*, 31(1):59–80.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. **SemEval-2014 task 4: Aspect based sentiment analysis**. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Jane Suiter and David M. Farrell. 2011. *The Parties’ Manifestos*, pages 29–46. Palgrave Macmillan UK, London.
- Margit Tavits and Natalia Letki. 2009. When left is right: Party ideology and policy in post-communist europe. *American Political Science Review*, 103(4):555–569.
- George Tsebelis. 1999. Veto players and law production in parliamentary democracies: An empirical analysis. *American political science review*, 93(3):591–608.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Sandra Wankmüller. 2021. Introduction to neural transfer learning with transformers for social science text analysis. *Sociological Methods & Research*, page 00491241221134527.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Appendix

A Confusion matrix

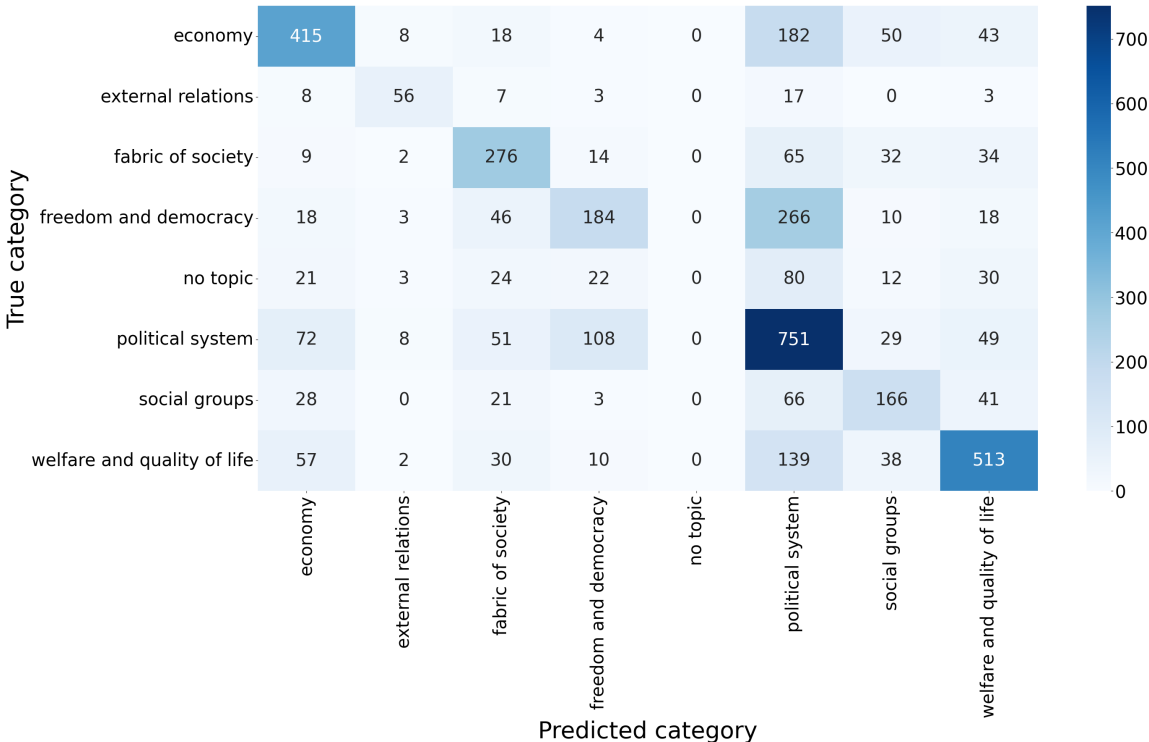


Figure 4: Confusion matrix for the performance of the English DistilBERT model on the test set of the New Zealand parliamentary speeches.

Data Augmentation for Multi-Class Eating Disorders Text Classification

Ghofrane Merhbene and Mascha Kurpicz-Briki

Applied Machine Intelligence, Bern University of Applied Sciences

Biel/Bienne, Switzerland

ghofrane.merhbene@bfh.ch

Abstract

In this study, we tackle the challenge of detecting Eating Disorders (EDs) in German text, a relatively unexplored area in natural language processing (NLP) for mental health. In this project, we developed a manually annotated German dataset from YouTube comments. To address the class distribution imbalance, we employed back translation as a data augmentation technique. This process significantly enhanced the dataset's utility. Through a comprehensive grid search, we identified a Support Vector Machine (SVM) model as the most effective, achieving an average F1-score of 0.83. Our findings not only contribute to the research field of ED detection in German but also demonstrate the effectiveness of innovative data augmentation techniques in managing class imbalances in natural language processing.

1 Introduction

The application of Natural Language Processing (NLP) in mental health diagnosis represents great potential for the future of healthcare (see, e.g., (Rezaii et al., 2022)). Its use may help clinicians in their diagnostic processes, marking a pivotal shift in the treatment and understanding of mental health. Despite this potential, the field faces a significant limitation: the predominant focus on English in research, which hinders the applicability of findings across diverse linguistic contexts. Eating disorders (EDs) typically manifest as detrimental eating habits, disruptions in behaviors, thoughts, and emotions related to food, which can lead to significant weight loss or gain in some cases. These disorders affect not only mental well-being but also physical health. Classified under the F50 category in the ICD-10 (World Health Organization, 1992), EDs encompass various conditions such as anorexia, bulimia, and overeating¹.

¹<https://www.icd10data.com/ICD10CM/Codes/F01-F99/F50-F59/F50->

In response to this research gap, our study introduces an innovative method focused on the identification of various types of EDs in German text. This approach not only widens the linguistic scope of current research, but also contributes to a more global understanding of EDs in different cultures and languages.

Our paper presents several substantial advancements to the current state of the art. Primarily, we have trained and tested a classifier capable of detecting a range of Eating Disorders, thereby pushing the boundaries of automated diagnostic tools in NLP for mental health research. Additionally, we provide insights on the effectiveness of data augmentation with back translation for a task like ED detection in German with few training data being available.

The structure of this paper is as follows: Section 2 offers a comprehensive review of related work, laying the groundwork for our research. Section 3 presents our proposed methodology and the materials used. Section 4 is dedicated to the presentation and discussion of our experimental results, showcasing the efficacy and insights derived from our model. Section 5 concludes the paper with a summary of our findings and potential directions for future research. Lastly, Section 6 reflects on the limitations of our study, ensuring a balanced and critical understanding of our work.

2 Related Work

The current state-of-the-art on the application of NLP in detecting EDs in languages other than English remains relatively sparse. To date, very few efforts in this area have been primarily focused on specific languages, with only one study addressing Spanish (López Úbeda et al., 2019) and two dedicated to Polish (Spinczyk et al., 2020; Rojewska et al., 2022). Furthermore, the majority of the English focused studies have predominantly used data from social media platforms such as Red-

dit (Yan et al., 2019), Twitter (Benítez-Andrades et al., 2021; López Úbeda et al., 2019; Zhou et al., 2020; Benítez-Andrades et al., 2022; Wang et al., 2017; He and Luo, 2016), and Tumblr (He and Luo, 2016; De Choudhury, 2015). A significant number of these investigations have leveraged the CLEF eRisk dataset (Wang et al., 2018; Ragheb et al., 2018; Aguilera et al., 2021; Aragon et al., 2021; Mohammadi et al., 2019; Ramiandrisoa and Mothe, 2020; Paul et al., 2018; Trozcek et al., 2018; Ramiandrisoa et al., 2018; Ortega-Mendoza et al., 2018; Liu et al., 2018; Merhbene et al., 2023), a fundamental resource provided by the Conference and Labs of the Evaluation Forum (CLEF).

CLEF eRisk (Parapar et al., 2023), an annual event in the research community, is designed to evaluate and benchmark the capabilities of various NLP systems in identifying and analyzing high-risk and harmful content on social media, including language patterns indicative of EDs, self-harm, and suicidal tendencies. This initiative offers a repository of social media posts, primarily from Reddit, and facilitates a competitive environment for teams to develop and assess their models for early detection of such critical issues. The overarching goal is to propel advancements in NLP, specifically in the context of recognizing and interpreting high-risk content on social media platforms.

Using a common dataset like CLEF eRisk enables researchers to strive for optimal outcomes in a competitive setting. However, this approach is not without its limitations. A key concern is the potential for leading NLP technologies to become overly adapted to patterns unique to this dataset, possibly leading to reduced efficacy when applied to varied types of data. This underscores the necessity of integrating greater diversity in training and evaluation datasets for NLP models targeting ED detection.

Despite these challenges, the field has witnessed some promising developments. For instance, López Úbeda et al. (2019) reported an impressive F1 score of 0.91 using supervised machine learning models. Similarly, Wang et al. (2017) achieved an accuracy of 0.97 using a Support Vector Machine classifier, leveraging user-based metrics encompassing social status, behavior, and psychometry. These successes indicate the potential of NLP in the realm of automatic ED detection, pointing towards a promising direction for future research.

3 Materials and Methods

3.1 Dataset

German served as the principal language for this work. A high-quality dataset is pivotal in such research; therefore, we used the YouTube API² to compile a robust dataset of anonymized German YouTube comments. To methodically identify videos relevant to Eating Disorders (EDs), we developed search queries incorporating specific keywords indicative of various EDs. These keywords included Essstörungen (eating disorders), Anorexie (anorexia), Bulimie (bulimia), among others that are closely associated with eating behaviors and body image issues. We also formulated query phrases to capture a broad spectrum of personal and informative content, such as "Meine Erfahrung mit Essstörungen" (My experience with eating disorders), "Leben mit einer Essstörung" (Living with an eating disorder), and "Magersucht OR Bulimie: Wie ich es geschafft habe" (Anorexia OR Bulimia: How I overcame it). These queries were designed to ensure the inclusion of a diverse range of video content related to the spectrum of eating disorders.

After identifying relevant videos, we extracted all comments under each video, ensuring comprehensive coverage of public discourse on these topics. The collected comments underwent a meticulous annotation process. Three domain-specific annotators manually labeled the data, adhering to a detailed set of annotation guidelines we developed. We employed a majority agreement method to finalize the annotations. Given the multifaceted nature of EDs, we used a multi-label annotation framework with six labels based on the ICD-10 (World Health Organization, 1992), as detailed in Table 1.

To address potential inconsistencies in manual annotations and ensure the reliability of our dataset, we computed inter-annotator agreement using pairwise Cohen's kappa coefficient for each label. The kappa values ranged from fair to substantial agreement, highlighting a generally reliable annotation process despite the subjective complexities involved in interpreting comments related to EDs.

Table 2 gives some insights into the dataset, highlighting key statistics such as the total number of samples and the average and standard deviation of text lengths measured in characters. To further char-

²<https://developers.google.com/youtube/v3>

Label	Description
A (Anorexia)	For cases where individuals engage in extreme calorie restriction, excessive exercise, or purging to control their weight or body shape.
O (Overeating)	For cases where individuals engage in behaviors like eating large amounts of food rapidly, feeling a loss of control over their eating, or eating when not physically hungry.
B (Binge Eating)	For cases where individuals engage in binge eating followed by purging behaviors such as vomiting, using laxatives, or excessive exercise. Common features of bulimia include weight fluctuations, tooth decay, and dehydration.
N (No ED)	For individuals who do not show any eating disorder behaviors or symptoms. An example is someone with a healthy relationship with food and their body, displaying no signs of disordered eating.
P (Previous ED)	For individuals with a history of eating disorders but are currently in recovery, showing no symptoms. An example is someone who had anorexia but is now in remission, identified by their use of past tense when describing their experience with an eating disorder.
K (No label)	When none of the other labels applies.

Table 1: Labels Description.

No. of Samples	Mean Text Length (chars)	Std. Dev. of Text Length (chars)
743	561	726.7

Table 2: Dataset Statistics.

acterize the dataset, we analyzed the lexical diversity and the average number of words per comment. The results are summarized in Table 3.

Lexical Diversity	Avg. nb. of words per comment
0.18	92.14

Table 3: Dataset Characteristics.

The lexical diversity score is calculated by dividing the number of unique words by the total number of words (over all datapoints). The resulting score of this metric is 0.18. Additionally, the average word count of 92.14 per comment demonstrates that the comments are detailed enough to provide substantial textual content for analysis, allowing for the expression of personal experiences and insights crucial for understanding public perceptions and misconceptions related to eating disorders.

A significant challenge associated with working with data derived from social media is ensuring a balanced representation of the targeted classes within a study. Our dataset notably exhibited this issue of imbalance, as depicted in Figure 1.

Label Distribution

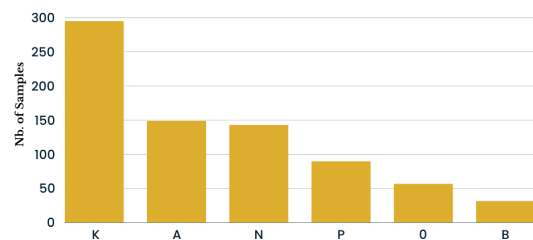


Figure 1: Label Distribution (A= Anorexia, O= Overeating, B= Binge Eating, N= No ED, P= Previous ED, K= No Label)

3.2 Textual Data Augmentation

In the realm of NLP, data augmentation is a crucial technique in machine learning, especially in the context of limited or imbalanced datasets. It involves artificially increasing the diversity of data, without actually collecting new data. This is achieved by creating modified versions of existing data points using techniques such as synonym replacement, sentence shuffling, and back translation (Pellicer et al., 2023).

Translation is a paraphrasing technique that has surged in popularity with advancements in machine translation technologies and the widespread availability of online translation APIs (Li et al., 2022).

Back translation involves translating text from the original language to a secondary language and then back to the original language again (see example in Table 4). This technique has demonstrated its effectiveness in a range of applications. For example, Beddiar et al. (2021) relied on back translation to augment their dataset and improve the performance of their model. This method effectively generates additional, syntactically accurate data points while preserving the original semantic content. This ensures the enrichment of datasets without the risk of introducing low-quality, noisy data, thereby upholding the overall data integrity.

In this work, we use *MarianMT*³ (Tiedemann and Thottingal, 2020) a neural machine translation framework from Hugging Face. The back translation process was executed using three different languages: English, Dutch, and Luxembourgish.

However, it was not uniformly applied across all labels. The variation in label distribution, as shown in Figure 1, inspired this selective approach. For instance, label B had approximately 34 data points, whereas label O had about twice as many. Figure 3 illustrates the specific back translation augmentation applied to each label. Some labels, like A and N, underwent the process once, others like P twice, and labels O and P three times. It is important to note that label K, being the predominant class, was excluded from this augmentation process. The implementation of back translation in these languages effectively contributed to creating a larger more balanced dataset with 1377 entries, thus enhancing the diversity of the data used in our analysis. Table 5 displays the lexical diversity and average word count per comment in the augmented dataset.

Figure 2 shows the new label distribution after augmenting the dataset using back translation. This technique improved the label distribution in the dataset, especially for the categories with previously moderate to high sample counts such as Anorexia (A) and No ED (N). This technique has effectively increased the representation of most labels, helping to reduce the initial imbalance.

3.3 Performance Metrics

To assess the performance of our proposal we rely mainly on F1-score and Balanced Accuracy.

- **F1-score:** The harmonic mean of precision

³https://huggingface.co/docs/transformers/model_doc/marian

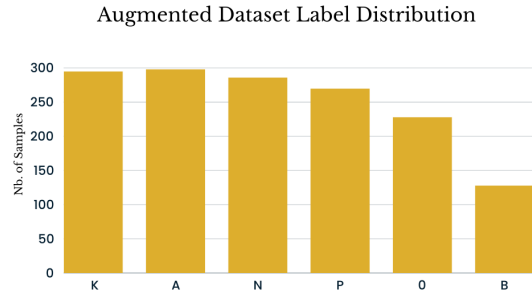


Figure 2: Augmented Dataset Label Distribution (A= Anorexia, O= Overeating, B= Binge Eating, N= No ED, P= Previous ED, K= No Label).

and recall. Precision is the ratio of true positive predictions to the total number of positive predictions, while recall is the ratio of true positive predictions to the total number of actual positives.

$$F1\text{-score} = 2 \times \frac{\left(\frac{TP}{TP+FP}\right) \times \left(\frac{TP}{TP+FN}\right)}{\left(\frac{TP}{TP+FP}\right) + \left(\frac{TP}{TP+FN}\right)} \quad (1)$$

- **Balanced Accuracy:** It is particularly useful for evaluating classification performance on datasets with imbalanced class distributions. It is defined as the average of recall obtained on each class.

$$Balanced\ Accuracy = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \quad (2)$$

Where TP, FN, TN, and FP correspond to true positive, false negative, true negative and false positive respectively.

3.4 Classification

A critical aspect to consider, particularly when working with data derived from social media platforms, is text pre-processing. This step is instrumental in refining and standardizing the input text, thereby significantly enhancing its quality and consistency. Our dataset underwent a thorough pre-processing to prepare it for effective model training. This involved the following operations: URL removal, HTML tag removal, removal of special

Process Step	Text
Original German Text	Die Essstörungen haben mein Leben kaputt gemacht.
Translated to Dutch	Eetstoornissen hebben mijn leven geruïneerd.
Back Translated to German	Essstörungen haben mein Leben ruiniert.

Table 4: Example of the back translation process used in data augmentation. The original German text translates to "Eating disorders have ruined my life." in English.

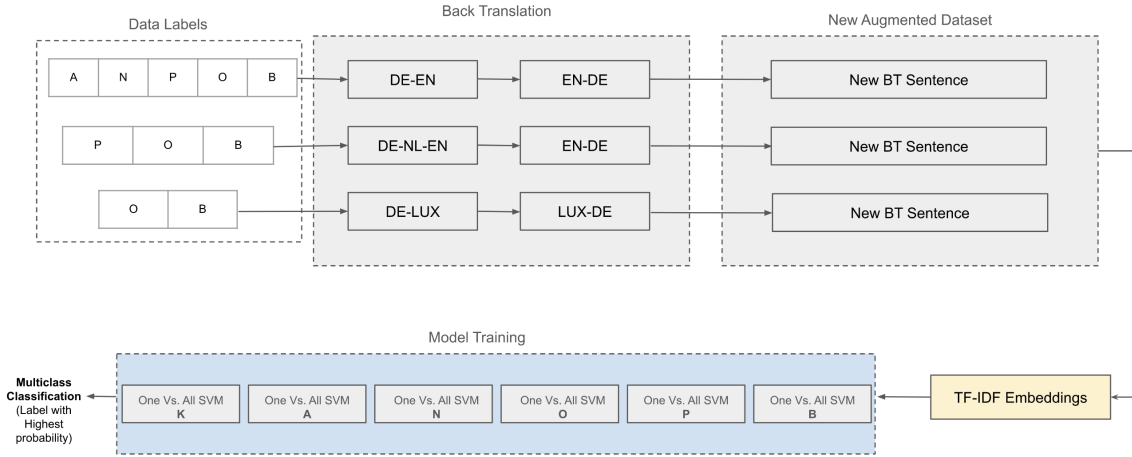


Figure 3: General scheme of the proposed architecture.

Lexical Diversity	Avg. nb. of words per comment
0.11	101.67

Table 5: Augmented Dataset Characteristics.

characters and numbers, case normalization, tokenization, and finally stop words removal. Following this and to ensure robust model evaluation, 30% of our original dataset was designated as the test set to assess model performance. The remaining 70% of the original data, supplemented with synthetic data, was allocated for training purposes. To compute the word embeddings, we rely on the Term Frequency-Inverse Document Frequency (TF-IDF) with a vector size of 5000. TF-IDF is a statistical measure that assesses the relevance of a word within a document set, or corpus. The TF component of TF-IDF increases proportionally with the number of times a word appears in a document, reflecting its importance. Conversely, the IDF component inversely scales the weight of the word based on its frequency across the entire corpus. This adjustment is crucial as it diminishes the influence of words that occur commonly across all documents, thereby helping to highlight more distinctive terms within each document.

To ensure that our classifier can accurately identify sentences with multiple labels, a *One-vs-All*

model training approach is applied. Thus, a separate model is trained for each label. This method is especially relevant given the overlapping and multifaceted nature of eating disorders, allowing for a more nuanced and comprehensive classification, as demonstrated in Figure 3.

To identify the most effective model for each label, we conducted a grid search across various models, including Multinomial Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, Logistic Regression, SVM, and Multilayer Perceptron (MLP) Classifier. For the SVM, we used a linear kernel to maintain computational efficiency while capturing linear relationships, set the regularization parameter C to 10 to allow for some misclassification but with a firm margin, and chose 'scale' for the gamma parameter to automatically adjust it according to the number of features, ensuring the model's adaptability. The MLP classifier, which is a type of neural network, was allowed 1000 iterations, giving the network ample opportunity to converge on a solution and learn from the data effectively. The other classifiers were used with their default parameters to establish a baseline performance.

The selection of the best-performing model for each label was based on the F1-score. This evaluation process was applied to both the original and the augmented datasets to ensure the most accu-

rate and effective model selection. Additionally, to avoid overfitting, we ensured that the synthetic data was used solely for training purposes, while deliberately excluding it from the testing phase.

4 Results and Discussion

4.1 Results

Tables 6 and 7 present the F1-score results from our experiments on the original and the augmented datasets, respectively. In these tables, we've accentuated the best results corresponding to each label. The Decision Tree algorithm achieved better results for the labels B, and O. Meanwhile, the Random Forest algorithm was the frontrunner for label K, the gradient-boosting classifier took the lead for label K, B, surpassing the performance of all other models. The MLP classifier was distinguished as the best for label A, while the SVM stood out for labels A and N, indicating its robustness across these particular categories.

The enhancement in performance after dataset augmentation is clear. Notably, we observed a substantial uptick in results, with an approximate 70% improvement for label B and a significant 50% boost for label A.

When looking at the average performance across all labels, it becomes clear that SVM classifier outperforms all others, achieving an F1-score of 0.41 before data augmentation and 0.83 after. Additionally, the implementation of back translation as a data augmentation technique significantly enhanced the average performance, yielding an approximate improvement of 40%.

Figure 4 shows the results in terms of balanced accuracy for each label. From the graph, it is observable that SVM when trained on the augmented dataset outperforms all other classifiers, while MLP follows close behind. We can also notice that most classifiers benefited from data augmentation, although the Random Forest and Gradient Boosting models did not show improvement on label K when comparing the augmented dataset to the original.

4.2 Discussion

Back translation for data augmentation has garnered increasing recognition for its potential to enhance datasets across various fields. In this study, applying this method to our dataset and integrating it with different machine learning models resulted in a notable improvement in both F1-score and Balanced Accuracy. Such performance enhancements

are likely linked to the introduction of linguistic variations by back translation, which contribute to a more robust and varied dataset. This aspect is particularly important in the realm of Eating Disorders, where the subtle nuances of language and expression are key to accurately identifying and categorizing the different types of EDs.

It is also important to note that the efficiency of this method has been further emphasized by its successful application in other research contexts. For example, [Corbeil and Ghadivel \(2020\)](#) have demonstrated the efficiency of back-translation's paraphrasing capability and its ability to generate robust and diverse new data points, and [Bédi et al. \(2022\)](#) found that using back-translation to augment a dataset on hate speech was beneficial for their machine learning model. Furthermore, ([Beddiar et al., 2021](#)) reported a significant enhancement in their study, where the application of back translation on a novel cyberbullying detection dataset using a convolutional neural network (CNN) architecture led to a 42% improvement in the F1-score. This correlation between our results and those of other studies reinforces the broad applicability and effectiveness of this method.

It is important to note that the only label that showed little to no improvement of both F1-score and Balanced Accuracy metrics was label K. This was likely because the augmentation was omitted for this label as it is already the dominant class.

5 Conclusion

In this study, we addressed a notable research gap by focusing on the automatic detection of Eating Disorders (EDs) in German text, and thus contributing to the state-of-the-art of NLP for mental health. Our work led to developing a specialized, manually annotated dataset tailored for ED detection in German. Despite facing challenges with a significant class imbalance within the dataset, we successfully implemented back translation for data augmentation to tackle this challenge. This approach not only helped in balancing the dataset but also significantly enhanced the model performance. It resulted in a remarkable 40% overall improvement in F1-score and a notable increase in the Balanced Accuracy score when used with SVM for classification over our 6 data classes. Our findings underscore the potential of language-specific resources and targeted augmentation techniques in improving the accuracy of automatic ED detection systems.

Label	MNB	DT	RF	GB	LR	SVM	MLP
A	0.00	0.38	0.00	0.10	0.00	0.42	0.50
B	0.00	0.33	0.00	0.33	0.00	0.00	0.00
K	0.28	0.56	0.73	0.73	0.66	0.64	0.62
N	0.00	0.29	0.00	0.24	0.00	0.46	0.29
O	0.00	0.55	0.00	0.35	0.00	0.47	0.00
P	0.00	0.25	0.00	0.23	0.00	0.50	0.18
Average	0.05	0.39	0.12	0.33	0.11	0.41	0.27

Table 6: F1-scores for each label using different models on the Original dataset (MNB: MultinomialNB, DT: DecisionTree, RF: RandomForest, GB: GradientBoosting, LR: LogisticRegression, SVM: SVC, MLP: MLPClassifier)

Label	MNB	DT	RF	GB	LR	SVM	MLP
A	0.00	0.59	0.62	0.57	0.42	0.86	0.86
B	0.0	0.77	0.80	0.80	0.00	0.96	0.96
K	0.03	0.59	0.33	0.64	0.44	0.73	0.60
N	0.00	0.47	0.31	0.30	0.04	0.72	0.59
O	0.0	0.61	0.76	0.81	0.20	0.84	0.80
P	0.0	0.54	0.67	0.67	0.39	0.88	0.83
Average	0.005	0.60	0.58	0.63	0.25	0.83	0.77

Table 7: F1-scores for each label using different models on the Augmented dataset (MNB: MultinomialNB, DT: DecisionTree, RF: RandomForest, GB: GradientBoosting, LR: LogisticRegression, SVM: SVC, MLP: MLPClassifier)

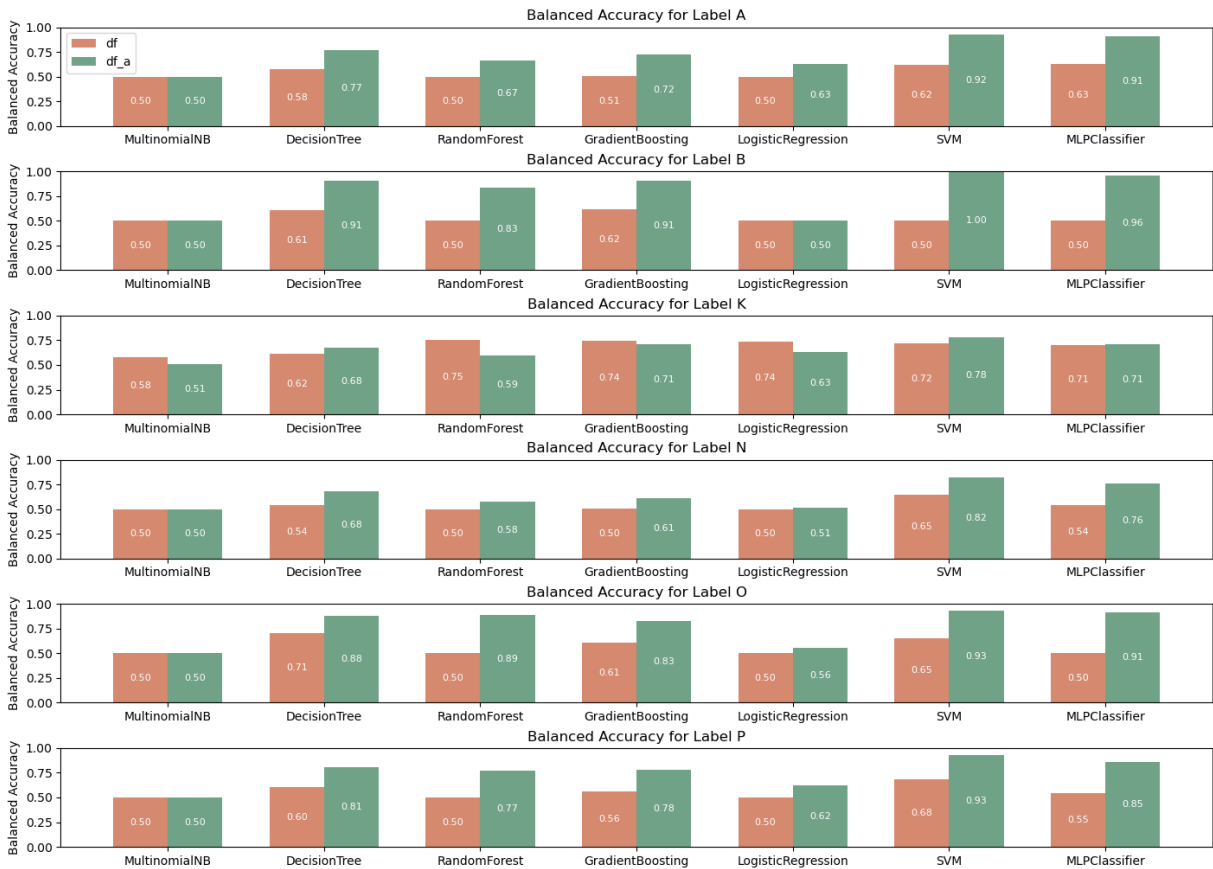


Figure 4: Balanced Accuracy for each label using different models with the original and the augmented dataset (df corresponds to the original dataset and df_a to the augmented dataset).

6 Limitations

In considering the limitations of this study, it is important to acknowledge the constraints associated with the dataset. While we have made significant strides in data augmentation through back translation, the severe initial imbalance in class distribution may still have residual effects on the generalizability of our findings. Notably, some labels, such as label B with only 34 examples, required the use of the entire dataset for augmentation to ensure their inclusion in the testing process. This approach was essential for maintaining a balanced representation across different classes, albeit potentially limiting the variety of testing scenarios.

Moreover, the manual annotation process, despite being thorough, is subject to human error and interpretative variability, which could influence the reliability of the dataset. Additionally, the reliance on text from YouTube comments presents a limitation in terms of linguistic variety and depth, as it still remains a social media platform and it may not fully represent the broader spectrum of language use associated with eating disorders, or the general population. Another limitation is the diversity in the available training data. As the used comments were anonymous, no information about gender or other characteristics of the authors of the texts were available. This needs to be addressed in future work to ensure that the classifier works with the same efficiency for different groups of the population. Finally, the performance of the SVM model, while promising, was evaluated within the context of this specific dataset, and its applicability to other datasets or in a real-world scenario requires further validation.

Ethics Statement

The work presented in this paper is part of a research project investigating NLP for mental health. The data collection and processing followed an internal guideline that was established in collaboration with a legal advisor. All data was anonymized. The annotation was done by domain experts who were hired at adequate local conditions and who are familiar with the sensitivity of the texts provided. The targeted tools derived from this research aim to provide further insights to clinical professionals, not to replace them. Given the potential limitations of such methods and datasets, the authors consider it highly relevant to keep human experts in the loop.

Acknowledgements

The authors gratefully acknowledge the support of the Inventus Bern Foundation for our research in the field of augmented intelligence for the detection of eating disorders.

References

- Juan Aguilera, Delia Irazú, Irazú Hernández Farías, María Ortega-Mendoza, and Manuel Montes-Y-Gomez. 2021. Depression and anorexia detection in social media as a one-class classification problem.
- Mario Ezra Aragon, Adrian Pastor Lopez-Monroy, Luis-Carlos Gonzalez-Gurrola, and Manuel Montes. 2021. Detecting mental disorders in social media through emotional patterns - the case of anorexia and depression. *IEEE Transactions on Affective Computing*, pages 1–1.
- Djamila Romaiissa Beddiar, Md Saroar Jahan, and Mourad Oussalah. 2021. [Data expansion using back translation and paraphrasing for hate speech detection](#). *Online Social Networks and Media*, 24:100153.
- Branislav Bédi, Hakeem Beedar, Belinda Chiera, Nedelina Ivanova, Christèle Maizonniaux, Neasa Ní Chiaráin, Manny Rayner, John Sloan, and Ghil’ad Zuckermann. 2022. [Using LARA to create image-based and phonetically annotated multimodal texts for endangered languages](#). In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 68–77, Dublin, Ireland. Association for Computational Linguistics.
- José Alberto Benítez-Andrades, José Manuel Alija-Pérez, Isaías García-Rodríguez, Carmen Benavides, Héctor Alaiz-Moretón, Rafael Pastor Vargas, and María Teresa García-Ordás. 2021. Bert model-based approach for detecting categories of tweets in the field of eating disorders (ed). In *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 586–590.
- José Alberto Benítez-Andrades, José-Manuel Alija-Pérez, Maria-Esther Vidal, Rafael Pastor-Vargas, and María Teresa García-Ordás. 2022. Traditional machine learning models and bidirectional encoder representations from transformer (bert)-based automatic classification of tweets about eating disorders: Algorithm development and validation study.
- Jean-Philippe Corbeil and Hadi Abdi Ghadivel. 2020. [Bet: A backtranslation approach for easy data augmentation in transformer-based paraphrase identification context](#).
- Munmun De Choudhury. 2015. Anorexia on tumblr: A characterization study. *Digital Health*.
- Ling He and Jiebo Luo. 2016. What makes a pro eating disorder hashtag: Using hashtags to identify pro eating disorder tumblr posts and twitter users.

- Bohan Li, Yutai Hou, and Wanxiang Che. 2022. [Data augmentation approaches in natural language processing: A survey](#). *AI Open*, 3:71–90.
- Ning Liu, Zheng Zhou, Kang Xin, and Fuji Ren. 2018. Tual at erisk 2018. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018*, volume 2125 of *CEUR Workshop Proceedings*.
- Pilar López Úbeda, Flor Miriam Plaza del Arco, Manuel Carlos Díaz Galiano, L. Alfonso Urena Lopez, and Maite Martin. 2019. Detecting anorexia in spanish tweets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 655–663. INCOMA Ltd.
- Ghofrane Merhbene, Alexandre R. Puttick, and Mascha Kurpicz-Briki. 2023. [BFH-AMI at erisk@clef 2023](#). In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 727–735. CEUR-WS.org.
- Elham Mohammadi, Hessam Amini, and Leila Kosseim. 2019. Quick and (maybe not so) easy detection of anorexia in social media posts.
- Rosa M Ortega-Mendoza, A Pastor López-Monroy, Anilu Franco-Arcega, and Manuel Montes-Y-Gómez. 2018. Peimex at erisk2018: Emphasizing personal information for depression and anorexia detection.
- Javier Parapar, Patricia Martín-Rodilla, David E Losada, and Fabio Crestani. 2023. Overview of erisk 2023: Early risk prediction on the internet. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 294–315. Springer.
- Sayanta Paul, Jandhyala Sree Kalyani, and Tanmay Basu. 2018. Early detection of signs of anorexia and depression over social media using effective machine learning frameworks.
- Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. 2023. [Data augmentation techniques in natural language processing](#). *Applied Soft Computing*, 132:109803.
- Waleed Ragheb, Bilel Moulahi, Jérôme Azé, Sandra Bringay, and Maximilien Servajean. 2018. Temporal mood variation: at the clef erisk-2018 tasks for early risk detection on the internet.
- Faneva Ramiandrisoa and Josiane Mothe. 2020. Early detection of depression and anorexia from social media: A machine learning approach. In *CEUR-WS*, volume 2621 of *Proceedings of the Conference CIR-CLE 2020*, Samatan, France.
- Faneva Ramiandrisoa, Josiane Mothe, Farah Benamara, and Véronique Moriceau. 2018. Irit at e-risk 2018. In *9th Conference and Labs of the Evaluation Forum, Living Labs (CLEF 2018)*, pages 1–12, Avignon, France.
- Neguine Rezaii, Phillip Wolff, and Bruce H Price. 2022. Natural language processing in psychiatry: the promises and perils of a transformative approach. *The British Journal of Psychiatry*, 220(5):251–253.
- Katarzyna Rojewska, Stella Maćkowska, Michał Maćkowski, Agnieszka Różańska, Klaudia Barańska, Mariusz Dzieciatko, and Dominik Spinczyk. 2022. [Natural language processing and machine learning supporting the work of a psychologist and its evaluation on the example of support for psychological diagnosis of anorexia](#). *Applied Sciences*, 12(9).
- Dominik Spinczyk, Maciej Bas, Marcin Dzieciatko, Mateusz Maćkowski, Katarzyna , and Sylwia Maćkowska. 2020. [Computer-aided therapeutic diagnosis for anorexia](#). *BMC Medical Informatics and Decision Making*, 20(1):251.
- Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.
- Marcel Trotzek, Sven Koitka, and Christoph M Friedrich. 2018. Word embeddings and linguistic metadata at the clef 2018 tasks for early detection of depression and anorexia fhdo biomedical computer science group (bcsg).
- Tao Wang, Markus Brede, Antonella Ianni, and Emmanouil Mentzakis. 2017. Detecting and characterizing eating-disorder communities on social media.
- Yu-Tseng Wang, Hen-Hsen Huang, and Hsin-Hsi Chen. 2018. A neural network approach to early risk detection of depression and anorexia on social media text. In *CLEF*.
- World Health Organization. 1992. *The ICD-10 Classification of Mental and Behavioural Disorders: Clinical Descriptions and Diagnostic Guidelines*. World Health Organization.
- Hao Yan, Ellen Fitzsimmons-Craft, Micah Goodman, Melissa Krauss, Sanmay Das, and Patty Cavazos-Rehg. 2019. Automatic detection of eating disorder-related social media posts that could benefit from a mental health intervention. *International Journal of Eating Disorders*, 52.
- Sicheng Zhou, Yunpeng Zhao, Jiang Bian, Ann F Haynos, Rui Zhang, and Rui Zhang. 2020. Exploring eating disorder topics on twitter: Machine learning approach.

Fine-tuning the SwissBERT Encoder Model for Embedding Sentences and Documents

Juri Grosjean and Jannis Vamvas

Department of Computational Linguistics, University of Zurich
jurileander.grosjean@uzh.ch, vamvas@cl.uzh.ch

Abstract

Encoder models trained for the embedding of sentences or short documents have proven useful for tasks such as semantic search and topic modeling. In this paper, we present a version of the SwissBERT encoder model that we specifically fine-tuned for this purpose. SwissBERT contains language adapters for the four national languages of Switzerland – German, French, Italian, and Romansh – and has been pre-trained on a large number of news articles in those languages. Using contrastive learning based on a subset of these articles, we trained a fine-tuned version, which we call SentenceSwissBERT. Multilingual experiments on document retrieval and text classification in a Switzerland-specific setting show that SentenceSwissBERT surpasses the accuracy of the original SwissBERT model and of a comparable baseline. The model is openly available for research use.¹

1 Introduction

Sentence embeddings have become a valuable tool in natural language processing. Neural models are fed with sequence strings and convert them into embeddings, i.e. a numeric representation of the input text. These can be applied in a variety of contexts, e.g. information retrieval, semantic similarity, text classification and topic modeling.

SwissBERT (Vamvas et al., 2023) is a modular encoder model based on X-MOD (Pfeiffer et al., 2022), which was specifically designed for multilingual representation learning. SwissBERT has been trained via masked language modeling on more than 21 million Swiss news articles in Swiss Standard German, French, Italian, and Romansh Grischun. The model is designed for processing Switzerland-related text, e.g. for named entity recognition, part-of-speech tagging, text categorization, or word embeddings.

¹<https://huggingface.co/jgrosjean-mathesis/sentence-swissbert>

The aim of this work is to fine-tune the existing SwissBERT model for the embedding of sentences and short documents. Specifically, our hypothesis is that using a contrastive learning technique such as SimCSE (Gao et al., 2021) to fine-tune SwissBERT will yield a model that outperforms the base model as well as generic multilingual sentence encoders in the context of processing news articles from Switzerland.

This is evaluated on two natural language processing tasks that utilize sentence embeddings, namely document retrieval and nearest-neighbor text classification, both from a monolingual and cross-lingual perspective. Indeed, the experiments show that the fine-tuned SwissBERT, which we call SentenceSwissBERT, has a higher accuracy than baseline models. An especially strong effect was observed for the Romansh language, with an absolute improvement in accuracy of up to 55 percentage points over the original SwissBERT model, and up to 29 percentage points over the best SentenceBERT baseline.

2 Related Work

Sentence-BERT This approach introduced by Reimers and Gurevych (2019) enhances BERT and RoBERTa for generating fixed-size sentence embeddings. It investigated using the CLS-token, the mean of all output vectors (MEAN-strategy), or the max-over-time of output vectors (MAX-strategy) as sentence embeddings and found the MEAN-strategy to perform best. The method applies siamese and triplet network architectures to finetune pre-trained models, which enables them to learn high-quality sentence embeddings, e.g. for comparison via cosine similarity. The training approach entails three objective functions: classification, regression, and triplet, each with specific training structures. Data from SNLI (Bowman et al., 2015) and MultiNLI datasets (Williams et al., 2018) was used for training. Sentence-BERT has given

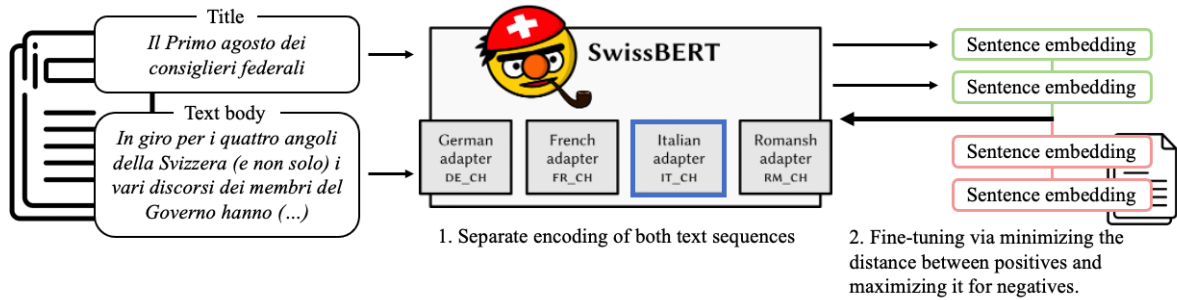


Figure 1: Visualisation of the supervised SimCSE training approach.

rise to a family of popular open-source encoder models.²

Multilingual Sentence Embeddings There are multiple approaches for training BERT-based encoder models for cross-lingual transfer. Reimers and Gurevych (2020) propose utilizing knowledge distillation to enhance mono-lingual models for multilingual use. Feng et al. (2022) found that harnessing pre-trained language models and fine-tuning them for cross-lingual tasks yields promising results while requiring less training data than training encoder models from scratch via multilingual language data like translations.

Contrastive Learning This technique originally surged in training neural models to perform vision tasks, e.g. image recognition. However, it has also been shown to deliver promising results with NLP tasks. The goal is for the model to learn an embedding space in which similar data is mapped closely to each other and unlike data stays far apart. For a mini-batch of N sentences, where (h_i, h_i^+) represent a pair of semantically-related sequences, h_j a random in-batch negative, and τ the temperature hyperparameter, the training objective looks as follows:

$$-\log \frac{e^{\cos_sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^N e^{\cos_sim(h_i, h_j^+)/\tau}} \quad (1)$$

Introduced by Gao et al. (2021), the SimCSE (simple contrastive sentence embedding) framework has been found highly effective when used in conjunction with pre-trained language models. This technique can be applied using an unsupervised or a supervised training.

For the unsupervised approach, the sequences in the training data are matched with themselves to create positive matches, i.e. the cosine similarity

between both outputs (MEAN pooling or CLS) is maximized. Thanks to the dropout masks, the embeddings of identical sequences still differ slightly.

The supervised approach uses a dataset of sentence pairs with similar meanings, and an optional third entry that is contradictory in meaning to the other two (hard negative). The similarity computation is maximized for the similar sentence pairs and minimized between the positives and the negatives.

3 Fine-tuning

To fine-tune SwissBERT for sentence embeddings, we opted for a (weakly) supervised SimCSE approach without hard negatives. Analogous to the original SwissBERT, Swiss news articles serve as the training data for this. The documents are split into sequence pairs, where one sequence consists of the article’s title and – if available – its lead concatenated, while the other contains the text body (see Figure 1). The title-body pairs represent (h_i, h_i^+) in the contrastive loss training objective 1.

3.1 Dataset

The fine-tuning data consists of over 1.5 million Swiss news articles obtained through the Swissdox@LiRI database³ in German, French, Italian, and Romansh (see Table 1). All German and French articles selected from the corpus have been published between 2020 and 2023, while the Italian and Romansh media date back to 2000, because the database contains fewer articles in these languages. The news articles are pre-processed analogous to SwissBERT’s original training data (Vamvas et al., 2023).

²<https://www.sbert.net/>

³<https://swissdox.linguistik.uzh.ch/>

Language	Documents	Tokens
German	760 350	621 107 750
French	644 416	567 688 406
Italian	63 666	35 109 282
Romansh	39 732	16 376 397
Total	1 508 414	1 240 281 835

Table 1: Composition of the dataset used for fine-tuning SwissBERT. We report the number of documents and tokens in the four languages.

3.2 Hyperparameters

The structure of the SimCSE train script provided by Gao et al. (2021)⁴ was updated and adapted according to SwissBERT, i.e. adding the X-MOD model architecture configuration as well as a language switch component, so that the model would continuously adjust its adapter according to the training data language during the training process. During fine-tuning on SimCSE, we froze the language adapters and updated all the other parameters. The training data was padded / truncated to 512 tokens, so that it fits the input limit. The model was fine-tuned in one single epoch, using a learning rate of 1e-5 and the AdamW optimizer (Loshchilov and Hutter, 2019), a batch size of 512 and a temperature of 0.05, which has been recommended for SimCSE (Gao et al., 2021). We used MEAN pooling, following the findings by Reimers and Gurevych (2019).

4 Evaluation

We evaluate SentenceSwissBERT on two custom, Switzerland-related NLP tasks in German, French, Italian, and Romansh. It is measured against the original SwissBERT and a multilingual SentenceBERT model that showed the strongest performance in the given evaluation tasks.

4.1 Dataset

For evaluation, we make use of the *20 Minuten* dataset (Kew et al., 2023), based on *20 Minuten*, one of the most widely circulated German-language newspapers in Switzerland. The articles tend to be relatively short and cover a variety of topics. Most of the documents in the dataset include a short article summary and topic tags

Given its format and features, the *20 Minuten* dataset is especially suitable for assessing SentenceSwissBERT’s performance. For the evalu-

⁴<https://github.com/princeton-nlp/SimCSE>

Task	Language	Documents
Document retrieval	German	499
	French	499
	Italian	499
	Romansh	499
Text classification: train set	German	4 986
	French	1 240
	Italian	1 240
Text classification: test set	German	1 240
	French	1 240
	Romansh	1 240

Table 2: Composition of the documents sourced from the *20 Minuten* dataset (Kew et al., 2023) that were employed for both evaluation tasks.

Category	Train articles	Test articles
accident	244	60
corona	1 468	367
economy	768	192
film	247	61
football	627	156
germany	250	62
social media	288	71
switzerland	300	743
ukraine war	268	66
usa	526	131
Total	4 986	1 240

Table 3: Composition of the test set of the text classification task, including the respective counts per category.

ation, all articles present in the *20 Minuten* corpus were removed from the original fine-tuning data in all languages, so that there is no overlap.

In order to expand the evaluation to French, Italian, and Romansh, the relevant parts of the articles were machine-translated via Google Cloud API (FR, IT) and Textshuttle API (RM). Using machine translation allows for a controlled comparison across languages when evaluating, since all documents share the same structure and content. Moreover, manual annotations can be automatically projected to the other languages without a need for additional annotation. A potential downside of machine translation is that the distribution of the test data does not reflect the diversity of human-written text. Tables 2 and 3 report statistics of the data we use for evaluation.

4.2 Tasks

Document retrieval For this task, the embedding of each article’s summary is compared to all the articles’ content embeddings and then matched by choosing the pair with the highest cosine similarity score. The performance is reported via the accuracy score, which is based on how many summaries were matched with the correct content in relation to the total number of articles processed. There is no train-test split performed for this task. It is performed monolingually (where the summary is written in the same language as the article) and cross-lingually.

Text Classification Ten categories are manually mapped from certain topic tags in the dataset. All documents without these (or overlapping) chosen topic tags are disregarded. Then, a random train-test split with a 80/20 ratio is performed once on the remaining data for every category respectively. The exact number of files per category are displayed in Table 3. Next, the text classification is carried out utilizing a nearest neighbors approach: The text body of each test article is compared to every embedding from the training data via cosine similarity. Subsequently, the topic tag of its one nearest neighbor from the training set (highest similarity) is assigned to it.

To assess cross-lingual transfer, the training data is kept in German for the assessment of each of the four languages, while the test data is machine-translated to French, Italian and Romansh. As the categories vary in frequency, the weighted average of all categories’ F1-scores is reported.

4.3 Baseline Models

SwissBERT While not specifically trained for this, sentence embeddings can already be extracted from the last hidden layer of the original SwissBERT encoder model via MEAN pooling. The input language is specified, just like in its newly fine-tuned version. This comparison demonstrates whether there is value in fine-tuning the model specifically for sentence embeddings.

Sentence-BERT Reimers and Gurevych (2019) propose several multilingual sentence embedding models.⁵ In this work, the *distiluse-base-multilingual-cased-v1* model is opted for as a baseline, as it shows the strongest performance for the

⁵<https://www.sbert.net/examples/training/multilingual/README.html>

given evaluation tasks (see Appendix B). It has originally been trained following the multilingual knowledge distillation approach introduced in Section 2, using mUSE (Chidambaram et al., 2019) as teacher model and a version of the multilingual Universal Sentence Encoder (Yang et al., 2020) as the student model. This version of Sentence-BERT supports various languages, among them French, German, and Italian, but not Romansh. Unlike with SwissBERT, the input language does not need to be specified. This model has a similar number of parameters as SwissBERT (see Table 4). However, it maps to a 512-dimensional embedding space and, hence, is computationally more efficient than SwissBERT.

The other multilingual Sentence-Transformer (*paraphrase-multilingual-mpnet-base-v2*) tested is much larger (278 043 648 parameters). Although this model maps to a 768-dimensional space, analogous to SwissBERT, it performed worse in the evaluation tasks than *distiluse-base-multilingual-cased-v1* (see Appendix B). Thus, it was disregarded.

Model	Vocabulary	Parameters
Sentence-BERT	119 547	135 127 808
SwissBERT	50 262	160 101 888

Table 4: Vocabulary sizes and parameter counts of the two baseline models. The fine-tuned SentenceSwissBERT has the same size as the original model.

5 Results

Document Retrieval Results for this evaluation task are reported in Table 5. SentenceSwissBERT outperforms its base model SwissBERT, demonstrating a clear improvement compared to the original model. The largest difference is noticeable in the processing of Romansh text.

SentenceSwissBERT also obtains better results than the Sentence-BERT baseline *distiluse-base-multilingual-cased*, except for two cases. Both models achieve high accuracy in both the monolingual and cross-lingual tasks. The clearest difference can be seen for German and especially Romansh, which Sentence-BERT was not trained on.

Text classification Table 6 presents the results of this evaluation task. Again, SentenceSwissBERT tends to improve over the baselines, with the excep-

Encoder Model	Summary Language	Article Language			
		DE	FR	IT	RM
SwissBERT (Vamvas et al., 2023)	DE	87.20	78.36	72.95	40.68
	FR	86.52	84.97	78.96	40.84
	IT	83.17	80.17	84.17	33.41
	RM	46.08	39.10	43.39	83.17
Sentence-BERT (Reimers and Gurevych, 2019)	DE	91.80	90.98	90.38	62.53
	FR	90.78	93.19	90.78	63.36
	IT	88.12	91.29	91.58	65.71
	RM	70.59	73.48	73.55	73.35
SentenceSwissBERT	DE	93.40	92.79	90.18	91.58
	FR	94.33	93.99	90.98	90.07
	IT	92.08	90.85	92.18	88.50
	RM	92.16	89.44	88.43	91.58

Table 5: Results for the document retrieval task using the *20 Minuten* dataset (Kew et al., 2023). The accuracy score is reported. The best results per language pair are marked in bold print.

Encoder Model	Training Language	Test Language			
		DE	FR	IT	RM
SwissBERT (Vamvas et al., 2023)	DE	77.93	69.62	67.09	43.79
Sentence-BERT (Reimers and Gurevych, 2019)	DE	77.23	76.83	76.90	65.35
SentenceSwissBERT	DE	78.49	77.18	76.65	77.20

Table 6: Results for the nearest-neighbor classification task using the *20 Minuten* dataset (Kew et al., 2023). A weighted F1-score is reported and the best results are marked in bold print.

tion of Italian, where the Sentence-BERT model is slightly more accurate.

6 Discussion and Conclusion

The results confirm that contrastive learning with title–body pairs is an effective fine-tuning approach for a masked language model. Using just a subset of 1.5 million articles from the original pre-training dataset, a clear improvement on the two sentence-level tasks has been achieved.

On the one hand, we observed an effect in monolingual tasks, e.g., by matching French summaries with French articles, or by performing nearest-neighbor topic classification of German articles using German examples. On the other hand, we also evaluated cross-lingual variations of those tasks, and found a clear benefit in the cross-lingual setting as well, even though we did not use cross-lingual examples in our fine-tuning. This suggests that modular deep learning with language adapters can be combined effectively with contrastive learning.

We expect that SentenceSwissBERT will be a

useful model variant for other Switzerland-related tasks that require sentence or document embeddings. For example, SentenceSwissBERT might be used for semantic search, or topic modeling based on document embeddings (e.g. BERTopic; Grootendorst, 2022). Future work could also explore whether including training data from other domains than news articles could further improve the generality of the model.

Limitations

The SentenceSwissBERT model has been trained on news articles only. Hence, it might not perform as well on other text domains. Additionally, the model input during training was limited to a maximum of 512 tokens. Thus, it may not be useful for processing longer texts. Finally, we note that we used machine-translated test data for evaluation in languages other than German.

Acknowledgements

The authors acknowledge funding by the Swiss National Science Foundation (project MUTAMUR; no. 213976). For this publication, use was made of media data made available via Swissdox@LiRI by the Linguistic Research Infrastructure of the University of Zurich (see <https://t.uzh.ch/1hI> for more information). The authors are indebted to Gerold Schneider for helpful guidance, and to Textshuttle for providing access to their Romansh machine translation API.

References

- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Muthu Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. 2019. [Learning cross-lingual sentence representations via a multi-task dual-encoder model](#). In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 250–259, Florence, Italy. Association for Computational Linguistics.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Maarten R. Grootendorst. 2022. [BERTopic: Neural topic modeling with a class-based tf-idf procedure](#). *ArXiv*, abs/2203.05794.
- Tannon Kew, Marek Kostrzewa, and Sarah Ebling. 2023. [20 minuten: A multi-task news summarisation dataset for German](#). In *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, pages 1–13, Neuchatel, Switzerland. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. [Lifting the curse of multilinguality by pre-training modular transformers](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Jannis Vamvas, Johannes Graën, and Rico Sennrich. 2023. [SwissBERT: The multilingual language model for Switzerland](#). In *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, pages 54–69, Neuchatel, Switzerland. Association for Computational Linguistics.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. [Multilingual universal sentence encoder for semantic retrieval](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

A Pre-training dataset media composition

Medium	Articles	Language	Medium	Articles	Language
lematin.ch	99 939	FR	Berner Zeitung	7 777	DE
24heures.ch	73 385	FR	Appenzeller Zeitung	7 548	DE
tdg.ch	69 498	FR	Zürcher Unterländer	7 425	DE
Le Temps	63 130	FR	Oltner Tagblatt / MLZ	7 420	DE
24 heures	62 004	FR	badenertagblatt.ch	7 140	DE
Tribune de Genève	57 604	FR	Berner Oberländer	7 138	DE
blick.ch	51 556	DE	Femina	7 106	FR
rsi.ch	51 526	IT	Toggenburger Tagblatt	7 032	DE
letemps.ch	48 353	FR	Thuner Tagblatt	6 982	DE
rts.ch	47 397	FR	solothurnerzeitung.ch	6 120	DE
cash.ch	46 750	DE	bzbasel.ch	5 921	DE
blick.ch	43 178	FR	RTS.ch	5 914	FR
rtr.ch	39 732	RM	Obwaldner Zeitung	5 854	DE
srf.ch	29 536	DE	Nidwaldner Zeitung	5 844	DE
nzz.ch	28 091	DE	TV 8	5 677	FR
tagblatt.ch	27 279	DE	Sonntagsblick	5 606	DE
luzernerzeitung.ch	23 855	DE	Grenchner Tagblatt	5 530	DE
Aargauer Zeitung / MLZ	21 868	DE	Solothurner Zeitung / MLZ	5 450	DE
Neue Zürcher Zeitung	18 408	DE	BZ - Langenthaler Tagblatt	5 277	DE
Le Matin Dimanche	18 352	FR	Sonntagszeitung	5 228	DE
Thurgauer Zeitung	17 335	DE	Limmataler Zeitung / MLZ	5 042	DE
Blick	14 636	DE	NZZ am Sonntag	4 991	DE
landbote.ch	13 089	DE	Finanz und Wirtschaft	4 962	DE
Tages-Anzeiger	13 040	DE	SWI swissinfo.ch	4 855	IT
bazonline.ch	12 709	DE	Glückspost	4 621	DE
aargauerzeitung.ch	12 309	DE	Limmataler Zeitung	4 513	DE
bernerzeitung.ch	12 207	DE	limmattalerzeitung.ch	4 488	DE
Zofinger Tagblatt / MLZ	11 888	DE	rts Vidéo	4 092	FR
tagesanzeiger.ch	11 612	DE	Die Weltwoche	4 011	DE
berneroberlaender.ch	11 603	DE	Bilan	3 979	FR
thunertagblatt.ch	11 581	DE	oltnertagblatt.ch	3 958	DE
zsz.ch	11 517	DE	grenchnertagblatt.ch	3 857	DE
L'illustré	11 231	FR	swissinfo.ch	3 575	IT
langenthalertagblatt.ch	11 184	DE	www.swissinfo.ch	3 541	IT
zuonline.ch	11 120	DE	swissinfo.ch	3 525	FR
Basler Zeitung	10 895	DE	PME Magazine	3 244	FR
derbund.ch	10 748	DE	illustre.ch	3 077	FR
schweizer-illustrierte.ch	10 620	DE	Schweizer Illustrierte	3 068	DE
Zuger Zeitung	10 557	DE	Handelszeitung	2 917	DE
bz - Zeitung für die Region Basel	10 528	DE	srf Video	2 558	DE
handelszeitung.ch	9 790	DE	Die Wochenzeitung	1 953	DE
pme.ch	9 491	FR	bellevue.nzz.ch	1 919	DE
Der Bund	9 396	DE	Thalwiler Anzeiger/Sihltaler	1 826	DE
Werdenberger & Obertoggenburger	9 214	DE	Zuger Presse	1 781	DE
Der Landbote	9 122	DE	HZ Insurance	1 617	DE
Zürichsee-Zeitung	9 019	DE	Schweizer Familie	1 570	DE
fuw.ch	8 791	DE	weltwoche.ch	1 466	DE
Luzerner Zeitung	8 651	DE	Beobachter	1 446	DE
Badener Tagblatt	8 435	DE	Zugbieter	1 409	DE
Urner Zeitung	8 284	DE	Guide TV Cinéma	1 384	FR
St. Galler Tagblatt	8 117	DE	weltwoche.de	1 267	DE
Wiler Zeitung	8 003	DE	Tele	1 176	DE
			Bilanz	1 085	DE
			swissinfo.ch	1 004	DE
			encore!	986	FR

Medium	Articles	Language	Medium	Articles	Language
Beobachter.ch	984	DE	Neue Zürcher Zeitung	3	IT
Das Magazin	982	DE	cash.ch	2	FR
züritipp (Tages-Anzeiger)	882	DE	Blick	2	IT
NZZ am Sonntag Magazin	823	DE	Berner Zeitung	2	IT
TV Star	764	DE	srf.ch	2	IT
weltwoche-daily.ch	719	DE	weltwoche.de	2	IT
bilanz.ch	596	DE	Blick	1	FR
SWI swissinfo.ch	587	FR	bernerzeitung.ch	1	FR
Streaming	535	DE	fuw.ch	1	FR
HZ Insurance	529	FR	Sonntagsblick	1	FR
NZZ PRO Global	446	DE	Basler Zeitung	1	FR
Schweizer LandLiebe	441	DE	weltwoche.ch	1	FR
glueckspost.ch	399	DE	weltwoche.de	1	FR
encore! (dt)	274	DE	srf.ch	1	FR
Newsnet / 24 heures	227	FR	bazonline.ch	1	FR
TV Land & Lüt	215	DE	rtr.ch	1	IT
NZZ Geschichte	151	DE	derbund.ch	1	IT
SI Sport	143	DE	St. Galler Tagblatt	1	IT
Newsnet / Berner Zeitung	143	DE	Die Weltwoche	1	IT
Bolero	142	DE	Das Magazin	1	IT
boleromagazin.ch	118	DE	nzz.ch	1	IT
NZZ Folio	109	DE	Basler Zeitung	1	IT
beobachter.ch	107	DE	Schweiz am Sonntag / MLZ	1	IT
Aargauer Zeitung / MLZ	91	FR	blick.ch	1	IT
HZ Insurance	77	IT	Cash	1	IT
SI Gruen	70	DE	bazonline.ch	1	IT
L'Illustré Sport	70	FR			
Newsnet / Basler Zeitung	69	DE			
Newsnet / Der Bund	58	DE			
Bolero F	56	FR			
Schweiz am Wochenende	47	FR			
Badener Tagblatt	34	FR			
Schweizer Versicherung	31	FR			
Newsnet / Le Matin	28	FR			
Newsnet / Tribune de Genève	25	FR			
Schweizer Illustrierte Style	23	IT			
Grenchner Tagblatt	22	FR			
Oltner Tagblatt / MLZ	21	FR			
Werdenberger & Obertoggenburger	21	IT			
Solothurner Zeitung / MLZ	20	FR			
Limmattaler Zeitung / MLZ	20	FR			
Finanz und Wirtschaft	18	FR			
NZZ Online	16	DE			
Schweizer Versicherung	16	IT			
TV4	12	DE			
Limmattaler Zeitung	9	FR			
rts Video	7	FR			
SWI swissinfo.ch	6	DE			
Newsnet / Tages-Anzeiger	6	DE			
Handelszeitung	6	IT			
Berner Oberländer	5	FR			
Thuner Tagblatt	5	FR			
berneroberlaender.ch	4	FR			
Beobachter.ch	4	IT			
thunertagblatt.ch	3	FR			

Table 8: Composition of the dataset used to fine-tune the SwissBERT model according to medium and language.

B Evaluation results of Sentence-BERT baselines

Encoder Model	Summary Language	Article Language			
		DE	FR	IT	RM
<i>paraphrase-multilingual-mpnet-base-v2</i> (Reimers and Gurevych, 2019)	DE	75.01	81.76	79.56	18.44
	FR	75.18	83.57	81.56	19.87
	IT	72.28	78.87	79.56	19.25
	RM	53.64	53.91	57.11	19.44
<i>distiluse-base-multilingual-cased-v1</i> (Reimers and Gurevych, 2019)	DE	91.80	90.98	90.38	62.53
	FR	90.78	93.19	90.78	63.36
	IT	88.12	91.29	91.58	65.71
	RM	70.59	73.48	73.55	73.35

Table 9: Results for the document retrieval task using two multilingual Sentence-BERT models. The accuracy score is reported. The best results per language pair are marked in bold print.

Encoder Model	Training Language	Test Language			
		DE	FR	IT	RM
<i>paraphrase-multilingual-mpnet-base-v2</i> (Reimers and Gurevych, 2019)	DE	75.42	75.64	73.88	39.38
<i>distiluse-base-multilingual-cased-v1</i> (Reimers and Gurevych, 2019)	DE	77.23	76.83	76.90	65.35

Table 10: Results for the nearest-neighbor classification task using the two multilingual Sentence-BERT models. A weighted F1-score is reported and the best results are marked in bold print.

How to make the first move? Analyzing the impact of an acquisition model in deep active learning for a sequence labeling task

Ngoc Duyen Tanja Tu

Leibniz Institute for the German Language

Mannheim, Germany

tu@ids-mannheim.de

Abstract

Deep active learning is a promising method for training a tagger when resources are low. In this paper, we systematically analyse the impact of differently performing initial acquisition models on the performance of each successor model in a custom sequence labeling binary classification task. For this purpose, we compare the performance of 5 successor models. They are trained on data queried by a random sampler and two differently performing initial acquisition models. In this way, we get an impression of the impact of an initial acquisition model with a certain performance on the performance of the successor models.

1 Introduction

Deep active learning (DAL) is used in sequence labeling tasks like named entity recognition (NER; e.g. Kim, 2020) or part-of-speech-tagging (POS-tagging; e.g. Chaudhary et al., 2021). DAL is a combination of deep learning (DL; first model proposed by McCulloch and Pitts, 1943) and active learning (AL; first introduced by Lewis and Gale, 1994). In DAL a DL model is initially trained on a small amount of labeled data. This model is called an initial acquisition model (Tsvigun et al., 2022). The acquisition model is used to select data points (samples) to be annotated from an unlabeled data pool based on an algorithm, called a query strategy. The samples are selected by the acquisition model in a way that intends to increase the performance of the model. The selected samples are then manually annotated and added to the training data. A new model, called a successor model is then trained on the enlarged training data. The successor model is used as an acquisition model in the next DAL iteration. The iteration process is repeated until a predefined termination condition is reached, for example the model achieves a certain f-score.

We use DAL for the development of a monitor corpus that consists of questions on the German

language, so-called language inquiries (Lang et al., 2023a) as in (1) *Heißt es **der Bayerische Ministerpräsident** oder **der bayrische Ministerpräsident**?* [‘Is it the **bavarian Minister-President** or the **Bavarian Minister-President**?’]¹. Language inquiries serve as a primary source of authentic language data for a variety of research questions (Breindl, 2016). The core of the corpus is a collection of approx. 50,000 language inquiries sent by email from (supposed) laypeople to a language consulting service between 1999 and 2019. To ensure that researchers have access to the data points from the corpus that are relevant to their research question, we tag the corpus in different ways (Lang et al., 2023b). Annotating a subset of 500 randomly extracted language inquiries from our corpus showed that 98% of the data contain examples. In most cases, the examples have the pattern as the bold print in (1) (see Section 3.1). For this reason, in this paper, we focus on this category of examples, that we call "list of alternatives". We train a tagger that detects the span of a language inquiry that is a list of alternatives.

The annotations obtained provide an interesting insight for linguists: Extracting the most frequent list of alternatives in language inquiries can help to identify standardization gaps in grammar. Moreover, the list of alternatives can be used in grammatical online resources to make it easier for users to find information about grammatical phenomena: Rather than having to be familiar with the terminology to look up grammatical information, users could click on list of alternatives that matched their question.

As we work with copyright-protected data, we cannot rely on crowd-sourcing to annotate the data. However, the spans to be tagged have a specific pattern: The alternatives listed in the language

¹Note that some translations are not in correct English but given in a way that the semantic meaning of the question becomes clear.

inquiries are very similar to each other (see (1)). Since the samples to be annotated show a specific pattern that is visible on the surface structure, it can be assumed that a model does not need a large amount of training data to learn this pattern. For this reason, DAL seems to be a suitable method. Although the spans to be annotated seem easy to tag because they all show a specific pattern, in most cases it is not possible to detect them with a rule-based approach (see Section 4.1). One reason is that the occurrences of the spans are not limited to certain positions in the language inquiries. Additionally, the length of the spans to be tagged can vary greatly: An annotation can span characters, words, multiword phrases or whole sentences. We therefore use DAL for our task for two key reasons: firstly, a rule-based approach is of limited use and secondly, DAL does not require a huge amount of labeled data to train a model.

The main contribution of this paper is a systematic analysis of the impact of the acquisition model on the performance of each successor model. We identified for our task that the performance of the successor model depends on the initial acquisition model. The insights of this study could be transferred to similar sequence labeling tasks like automatic tagging of paronym candidates. Moreover, unlike many other studies on DAL (Zhang et al., 2022), this task is not a simulation, so it is possible for us to address aspects like real annotation cost and time.

The paper is structured as follows: In Section 2 we give an overview of work relevant to our task. After that, in Section 3 we explain our experimental setup and present our results in Section 4. Finally in Section 5 we then summarise our findings and identify open questions.

2 Related work

As we are using DAL for a sequence labeling task, we only focus on papers that take a similar approach, in order to apply the results to our work. Studies with a textual data basis have shown that DAL models perform just as well as DL models, although significantly less data is used for training: Mirbostani et al. (2023) trained a DAL model for morphological inflection and morphophonological processing in Cairene Egyptian Arabic. Although the model is only trained on 30% of the data in comparison to the state-of-the-art model it performs just as well. Kim (2020) demonstrated for slot filling

that a DAL model trained only on 15% of the data compared to the best-performing DL model still achieved 98-99% performance. These results indicate that DAL sounds promising for sequence labeling tasks.

Studies on the ideal implementation of DAL focus mainly on the selection of the best query strategy. Query strategies can be divided into uncertainty-based algorithms, diversity-based algorithms and hybrid algorithms. Uncertainty-based algorithms use the acquisition model to predict the labels for the data points of the unlabeled data pool. The data points are then ranked by the confidence score of the model and the ones with the lowest confidence scores are selected for manual annotation. Diversity-based algorithms sort the data points of the unlabeled data pool by their similarity according to different strategies. The most different data points are then selected for manual annotation. Hybrid algorithms are a combination of an uncertainty-based and a diversity-based algorithm.

Chaudhary et al. (2021) developed a suitable query strategy for a DAL model trained for POS-tagging. However, the proposed strategy cannot be applied to our task because it considers the confusion caused by different annotations for the same token. For example, the German word *die* ('the') could either be a pronoun or an article. In our task the probability is the same that a given token is part of a list of alternatives or not. Mirbostani et al. (2023) showed that using a suitable query strategy, in their case: an uncertainty-based algorithm, outperforms the model which is trained on randomly selected samples from the unlabeled data pool. They could not demonstrate that combining uncertainty and diversity query strategies achieve better results. In contrast, Kim (2020) showed that a combination of uncertainty and diversity query strategies yield better results in DAL for NER and slot filling than pure uncertainty or pure diversity query strategies as well as random sampling.

Radmard et al. (2021) focused on reducing the cost of annotating the selected samples from the unlabeled data pool by only querying subsequences of each unlabeled sentence for NER. Annotated labels are then propagated to the unlabeled data pool if the subsequence is the same. Using subsequences in our task is not feasible because in our case the probability is the same that a given subsequence is part of an annotation or not, unlike in NER.

While the works listed above have trained DL

models from scratch, [Shelmanov et al. \(2021\)](#) used deep pre-trained models and showed that the latter outperforms the former.

None of these works analysed the impact of the initial acquisition model on the performance of the successor models. This shortcoming in the research to date will be addressed in this paper.

3 Experimental setup

3.1 Data preparation

The manually labeled data set consists of 500 language inquiries. They have been randomly extracted from our corpus of approx. 50,000 language inquiries. The vast majority of questions contain explicit examples (see for example: (1)-(14)). The data set was annotated in three steps:

(i) First, two linguists annotated the examples found in the data with the following four categories:

- "list of alternatives", e.g. (1),
- "sentence", e.g. (2) *Wie schreibe ich diesen Satz richtig: **Er ist am Witze erzählen***. [...] [‘How do I write this sentence correctly: "**He is telling jokes**".’],
- "expression", e.g. (3) *Eigentlich sind Wörter wie "kein" und "einzig" nicht stiegerbar [sic]. Allerdings frage ich mich, wie dann der Ausdruck "**in keinsten Weise**" zu erklären ist.* [...] [‘Actually, words like "none" and "only" are not gradable. However, I wonder how the expression "**in *nonest manner**" can be explained.’] and
- "other" if an example does not fit in one of the categories.

The classification is done in order to develop a strategy based on the examples per category with which the examples can be tagged automatically. The distribution of the categories is as follows: About 50% of the examples can be assigned to the category "list of alternatives", about 23% to "sentence", about 18% to "expression", about 6% fall in the category "other" and about 2% of the language inquiries do not contain an example at all. As "list of alternatives" is the most common category, we have opted for this paper to automatically tag it.

(ii) The language inquiries were tokenized by splitting at white space. Each token of a language inquiry tagged with "list of alternatives" in (i) were annotated by the two annotators. Every token was

Set	Token total	Token alternative
Train	12,799	1,248
Val	2,180	168
Test	2,753	345

Table 1: The distribution of all tokens and tokens that are part of a list of alternatives in the sets.

labeled according to the BIO format with "B-part-of-a-list" for the beginning of a span, "I-part-of-a-list" if the token is inside a span or "O" if the token is not part of a span. The tokens of language inquiries tagged with other categories than "list of alternatives" were tagged with "O". 90% of the annotated list of alternatives consists of two alternatives, the remaining 10% of 3 to 6.

The inter-annotator-agreement is very high with a Fleiss’ Kappa of 0.99. The two annotators discussed differing annotations and determined the final annotation.

500 language inquiries comprising 17,732 tokens were tagged and randomly split in training (approx. 70%), validation (approx. 15%) and test set (approx. 15%) in a way, so that no question is split in more than one data set (see table 1). Quotation marks are removed from the questions to prevent overfitting because 18% of the lists of alternatives are written in between quotation marks. Additionally, 36% of the language inquiries containing a list of alternatives either do not have the alternatives in quotation marks but other tokens or closing quotation marks are missing. In 46% of the data quotation marks are not used at all.

The remaining approx. 49,500 language inquiries from the corpus form the unlabeled data pool and will be shown to the model during each DAL iteration. 40 language inquiries are selected in each iteration increasing the manually labeled data set by approx. 10% per iteration. The unlabeled data pool was split into four due to the RAM size. Thus, per iteration four unlabeled data pools were shown in succession to the acquisition model and 10 language inquiries were selected from each. Due to low annotation power, only one linguist annotated the selected language inquiries per iteration.

3.2 Task

Our model was trained to tag spans of list of alternatives proposed in language inquiries. The alternatives listed can be:

- different spellings, e.g. (4) *...sie spiegeln den besonderen Charme des Schlosses wider. Oder? ...sie spiegeln den besonderen Charme des Schlosses wieder.*
[‘...they mirror the special charm of the castle. Or? ...they *mirror the special charm of the castle ’],
- syntactic alternatives, e.g. (5) [...] *heiße Himbeeren mit Vanilleeis, heiße Himbeeren auf Vanilleeis.* [...]
[‘Hot raspberries with vanilla ice cream, hot raspberries on vanilla ice cream. [...]’],
- (potential) synonyms, e.g. (6) *gibt es das Wort konzeptionalisieren/Konzeptionalisierung als Alternative zu konzeptualisieren/Konzeptualisierung.* [...]
[‘Does the word *conceptionalise/*conceptionalisation as an alternative to conceptualise/conceptualisation exist. [...]’],
- a list of several possibilities, e.g. (7) [...] *Heißen derartige Läden nun Asiashop, Asienladen oder gar Asialaden?*
[‘[...] Are such stores now called Asia shop, Asian store or even Asia store?’].

The alternatives can be presented in detail as in (4) or shortened as in (8) [...] *Ich arbeite in einer Firma, deren spannende(n) [...] Aufgaben sehr reizvoll sind.* [‘I work for a company whose *excit(e)ing [...] tasks are very appealing.’]. Note that in (8) the alternative is given within the parenthesis. In some cases, the alternatives are connected by *oder* (‘or’) as in (1) and (4) but in other cases, there is no connector at all.

The task is modeled as a sequence labeling, binary classification task. Every token will be labeled according to the BIO format (see Section 3.1). The length of spans can be whole sentences as in (4), words as in (6) or characters as in (9) *im Duden steht Schmant mit t, auf den Sahnetöpfchen [...] mit d.* [...] [‘in the dictionary Schmant is written with a t, on the cream pots [...] with a d.’]

The test set (see Section 3.1) contains 57 lists of alternatives, including 30 language inquiries on spellings, 11 on syntactic alternatives, 13 on synonyms and 3 lists of several possibilities.

3.3 Model

We use Flair sequence tagger (Akbik et al., 2019) as according to their GitHub page² many trained Flair sequence tagger models for different NLP tasks are state-of-the-art.³

The sequence tagger consists of the following layers: an embedding layer, an LSTM layer⁴ and a CRF-layer.

We have left the default values of the hyperparameters (see table 2) except for the embeddings: We test two different pre-trained embedding settings for the initial acquisition model:

- The pre-trained German BERT model⁵ dbmdz/bert-base-german-cased⁶ which achieves better performance than google-bert/bert-base-german-cased on NER⁷ as well as in our study.
- German non-contextualized FastText embeddings⁸ stacked on German contextualized Flair forward and Flair backward embeddings⁹ as suggested in Akbik et al. (2019).

BERT (Devlin et al., 2019) and Flair embeddings (Akbik et al., 2018) are trained differently: while BERT learns structures on token-level, Flair embeddings learn structures on character-level. On the one hand character-level models are better in handling typing errors (Gao et al., 2021), which is a big advantage for our non-normalized data. On the other hand, the span to be tagged is mostly a sequence of tokens. For these reasons, either BERT or Flair embeddings could be beneficial for our task.

²<https://github.com/flairNLP/flair>

³We also finetuned the models <https://huggingface.co/google-bert/bert-base-german-cased>, <https://huggingface.co/dbmdz/bert-base-german-cased> and <https://huggingface.co/distilbert/distilbert-base-german-cased> for our task to use as initial acquisition models but the f-scores are quite low with 0.29, 0.36 and 0.08.

⁴We also tested the performance of the models with a BiLSTM layer but the f-scores are lower than 0.15.

⁵We are aware that BERT only processes 512 subword tokens. Since only about 0.28% of our data basis is longer than this threshold, we accept the loss.

⁶<https://huggingface.co/dbmdz/bert-base-german-cased>

⁷<https://github.com/stefan-it/fine-tuned-berts-seq>

⁸<https://flairnlp.github.io/docs/tutorial-embeddings/classic-word-embeddings>

⁹<https://flairnlp.github.io/docs/tutorial-embeddings/flair-embeddings>

Hyperparameter	Value
Hidden size	256
Number of LSTM layers	1
Dropout	0.0
Classifier	Softmax + CRF
Mini batch size	32
Activation function	tanh
Max epoch	100
Initial learning rate	0.1
Patience	3
Annealing factor	0.5
Optimizer	SGD
Learning rate	Learning rate decay

Table 2: The default values of the hyperparameters.

Many aspects during training are set randomly, e.g. weight initialization. This can affect the performance of a model significantly (Reimers and Gurevych, 2017). For this reason, we trained the initial acquisition model five times.¹⁰

3.4 Active learning

We use the SeqAL framework¹¹ to perform active learning with the Flair sequence tagger. Various considerations were taken into account when selecting the query strategy: Choosing a pure uncertainty-based query strategy could result in samples that represent rare edge cases, that would make it difficult for the model to generalize from them. In addition to that, the model could select almost only similar samples that would result in redundant cases being annotated. Hence, a hybrid method could result in better performance from the model (Ren et al., 2022). Based on these arguments, we chose a hybrid query strategy.¹² As uncertainty-based method we chose maximum normalized log-probability (MNLP) which performs better than least confidence because it has no bias towards choosing longer sentences (Shen et al., 2017). The confidence score for each prediction, calculated by Viterbi loss, is used for this strategy. As diversity sampling (DS) method, we chose distribute similarity as it operates on token-level to compare the

¹⁰We only trained the initial acquisition model five times and not every successor model as we wanted to keep the initial random weights constant for each acquisition model per iteration. Whether this actually has an effect on the performance of the successor models has to be clarified.

¹¹<https://github.com/tech-sketch/SeqAL>

¹²We also experimented with selecting the samples only with MNLP and only with DS but these models performed worse than with a combination of MNLP and DS.

similarity of samples with each other, which is fitting for our task. For this method, the tokens of a sample are first embedded and the similarity is then calculated based on the cosine similarity. In contrast, cluster-based strategies are not suitable because we only tag one class. For comparison, we also selected samples randomly.

4 Results

4.1 Initial acquisition model

We report the performance of each model as f-scores only for the positive spans so the imbalance between the positive and negative class does not distort the results. We implemented a simple rule-based tagger as a baseline which detects the most frequent kind of list of alternatives: spelling variants (see Section 3.2). The rule-based tagger is only able to detect three types of spelling variants listed in language inquiries containing the token *oder* ('or'):

(i) the tagger checks if the token preceding *oder* and the token following *oder* in the language inquiry have a Jaro-Winkler similarity greater or equal 0.80.¹³ If this is the case the tokens are tagged as "part-of-a-list". This way, cases like (10) *Essensmarke oder Essenmarke. Und warum? [Foodstamp or foodstamps. And why?]* are tagged.

(ii) The tagger checks if Token A preceding *oder* is a substring of Token B following *oder* and if the length of Token A is shorter than Token B. If this is the case, Token C preceding Token A is concatenated with Token A. If the concatenated token equals Token B Token A, B and C are tagged as "part-of-a-list". This way, cases like (11) *hervorlugen oder hervorlugen [to peek out or to peekout]* are tagged.

(iii) Analogous to (ii) except for the following token, to tag cases like (12) *hinunterfallen oder hinunter fallen [to falldown or to fall down]*.

In addition, a random baseline was implemented, which randomly assigns "part-of-a-list" to 13% of the tokens and "not-part-of-a-list" to 87% as the test data show this distribution.

Five random seeds were set for the training of the sequence tagger with two different embedding configurations as described in section 3.3. The average performance of the models are reported in table 3.

¹³We tested different thresholds and learned that 0.80 works best.

Embedding	Prec	Rec	F-score
flair_emb	0.39	0.05	0.09
BERT	0.51	0.38	0.44
rule_based	0.91	0.06	0.11
random_baseline	0.15	0.15	0.15

Table 3: The performance of a rule-based baseline, of a random baseline and of sequence taggers with different embeddings on average of five random seeds.

Model	Prec	Rec	F-score
BERT_0	0.46	0.35	0.40
BERT_1	0.46	0.38	0.42
BERT_2	0.56	0.36	0.44
BERT_3	0.51	0.41	0.45
BERT_4	0.55	0.41	0.47

Table 4: The performance of each initial acquisition model trained with BERT and five different random seeds.

The sequence tagger trained with Flair embeddings performed worst, even worse than the random baseline and the simple rule-based baseline. It can be deduced that character-level embeddings are not suitable for our task, so we must use token-level embeddings. The sequence tagger trained with BERT performed by far the best. For this reason, we decided to use the sequence tagger trained with BERT and different random seeds for the analysis of finding the ideal acquisition model. As shown in table 4 the f-score of the models trained with different random seeds range from 0.40 to 0.47. We decided to use BERT_0, the worst performing initial acquisition model, and BERT_4, the best performing initial acquisition model, in our DAL approach. The hypothesis is that the worst initial acquisition model is not as confident as the best and therefore selects more helpful samples.

4.2 Performance of the successor model

The overlap of selected samples of each acquisition model initialized with BERT_0 and BERT_4 averages only 33% per iteration. There is no overlap of selected samples between each acquisition model and the random sampler. The differently performing acquisition models therefore actually select different samples from the unlabeled data pool per iteration. Thus, in a next step it was possible to analyse if one of the two differently performing acquisition models selected better samples leading to a better successor model.

This was indeed the case: The performance of the successor models per iteration (see figure 1 and 2) are different depending on the model that selected the training data. Figure 1 shows the f-score for each successor model initialized with BERT_0 with different training data selected by its subsequent acquisition models, the acquisition models initialized with BERT_4 (i.e. the best initial acquisition model) and a random sampler. Figure 1 shows that the model achieves the best f-score of 0.65 (precision: 0.67, recall: 0.64) in iteration 19 when trained on the samples selected by its subsequent acquisition models. This specific successor model from iteration 19 is referred to BERT_0_best in the following. In contrast, the model only achieves 0.61 as the highest f-score in iteration 20 when trained on the samples selected by the acquisition models initialized with BERT_4. Figure 2 seems to confirm our hypothesis from Section 4.1: trained on data selected by the acquisition models initialized with BERT_0, i.e. the worst initial model, it achieves the overall highest f-score of 0.67 (precision: 0.70, recall: 0.64) in iteration 20. This specific successor model is referred to BERT_4_best in the following. However, the difference in performance is not as pronounced: trained on the samples selected by its subsequent acquisition models, the model achieves an f-score of 0.66 (precision: 0.66, recall: 0.66) in iteration 20. Still, it performed worse.

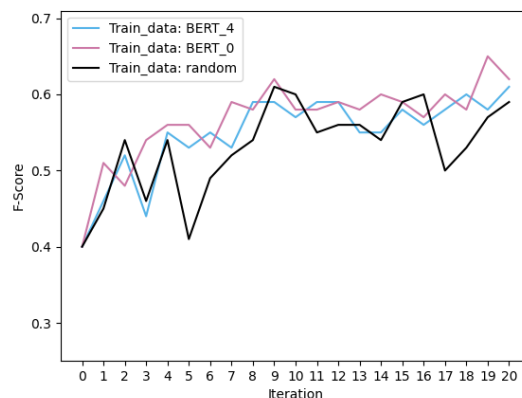


Figure 1: The f-score for all successor models initialized with BERT_0 and trained on data sampled by three different models per iteration.

To summarize: The best successor model of BERT_0 is achieved when the model is trained on data selected by its subsequent acquisition models in iteration 19 (BERT_0_best). The best suc-

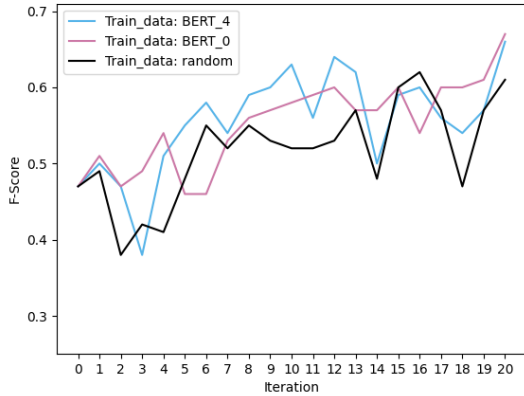


Figure 2: The f-score for all successor models initialized with BERT_4 and trained on data sampled by three different models per iteration.

cessor model of BERT_4 is achieved when the model is also trained on data selected by the acquisition models initialized with BERT_0 in iteration 20 (BERT_4_best). An error analysis of these two models shows that both models tend to overfit. Both models tend to tag spans that match a pattern similar to a list of alternatives, for example: (13) *Zusammenstellung einer 14-tägigen all inclusive Gruppenreise. Wie schreibt man all inclusive Gruppenreise nach den neuen Regeln, [...]*

[‘Compilation of a 14-day all inclusive group tour. How to write all inclusive group tour according to the new rules, [...]’].

Conjunctions are in most cases not part of a list of alternatives, that is the reason why BERT_0_best overfits and tends to not tag a conjunction even if it is part of an alternative, e.g. *oder* (‘or’) in the first sentence in (14) *..., danach das Fleisch medium oder durchbraten. Oder: ..., danach das Fleisch medium- oder durchbraten.*

[‘..., then roast the meat medium or well done. Or: ..., then roast the meat medium- or well done.’]. This is reflected in the slightly lower precision value of this model compared to the other. In addition to that, BERT_0_best tags tokens that are not alternatives more often than BERT_4_best. However, we cannot make out a rule why the tokens are falsely tagged by BERT_0_best.

The f-scores seem low, but if we look at the false positives like in (13) we can see that in many cases examples in the language inquiries were tagged that are not list of alternatives but fall into other categories (see section 3.1): For BERT_0_best it is 109 out of 127 false positives and for BERT_4_best

Model	0_train	4_train	random
BERT_0	0.65	0.61	0.61
BERT_1	0.64	0.61	0.60
BERT_2	0.63	0.65	0.63
BERT_3	0.70	0.62	0.62
BERT_4	0.67	0.66	0.62

Table 5: The best f-scores through all iterations for each random seed and each training data set sampled by the acquisition models initialized with BERT_0 (0_train), BERT_4 (4_train) and a random sampler (random).

it is 95 out of 109 false positives. These spans are of great interest for us, even if they do not correspond to the task at hand because we want to extract all examples from the language inquiries for the corpus development.

4.3 Train other models on the sampled training data

We see that the best performance results from training the best initial model with the samples selected by the acquisition models initialized with the worst initial model. The fact that the training samples of the worst initial model lead to a better performance is not limited to the two models evaluated in detail, BERT_0_best and BERT_4_best. Table 5 shows the best f-scores of the successor models of all five initial models differentiated according to the two training sets. Except for the best performing successor model initialized with BERT_2 the models achieve a better performance with the training data sampled by the acquisition models initialized with BERT_0, i.e. the worst initial model. It can also be seen, that the best f-score achieved by the models trained on the data sampled by the acquisition model initialized with BERT_4, i.e. the best initial model, is in most cases as high or similarly high as the models trained on the data randomly sampled. Although the models trained on the data sampled by the acquisition models initialized with BERT_0 achieve mostly the highest f-score, the models require an average of 18 iterations to achieve it. The models trained on the data sampled by the acquisition models initialized with BERT_4 achieve their highest score on average after 16 iterations and the models trained on the data sampled by the random sampler need 13 iterations to achieve their highest score. It is therefore a question of time resources as to whether performance can be lost.

4.4 Time investment

The annotation of the initial data set (all three steps) took about 14 hours in total for both annotators. Overall annotating the selected samples for each iteration in three DAL approaches took about 13 hours. Note that only one person annotated the selected samples. The training of each model took about 6.5 hours (approximately 20 minutes per iteration for each model) with 1 NVIDIA Tesla V100 GPU. We set a cut at iteration 20 because by then we reached the time we planned to invest in AL.

5 Conclusion

In our contribution, we analysed the impact of the acquisition model to the performance of each successor model. We have found that the performance of the successor models in our task differ depending on which model has selected the data to be trained with. Based on this finding, we suggest the following approach:

- (i) train several initial models with different random seeds,
- (ii) select the worst performing one as acquisition model,
- (iii) train the other initial models from (i) on the data selected by the subsequent acquisition models defined in (ii) and
- (iv) use the best performing model out of all trained successor models from any iteration to solve your task.

However, our analysis is restricted to only one specific task using German language data and five random seeds. For this reason, the experiment has to be repeated with additional random seeds as well as on other tasks in different languages to check whether our findings can be confirmed. There is still a lot to investigate, but this work provides the initial basis for fruitful further research and for the first time focuses on the initial acquisition model in a DAL setting.

As for our task, we will expand it to examples of different categories (listed in 3.1) as in the error analysis (see Section 4.2) we learnt that the trained model detects them. As we are interested in these anyway, this is very practical. With the aid of a text classification model (that is yet to be trained) or a rule-based approach we can then tag the examples with the corresponding categories.

Limitations

There is no state-of-the-art approach and we cannot (afford) to annotate the whole unlabeled data pool: therefore we do not know how the model would perform on the whole dataset.

As we only analysed the impact of the acquisition models for one task and one language it remains to be seen how scalable the findings are.

As only a limited number of random seeds are considered for the initial acquisition model, it is impossible to determine if a model exists that performs worse than the already identified worst initial acquisition model.

Acknowledgement

This work was performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC.

References

- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. [FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.
- Eva Breindl. 2016. [Sprachberatung im interaktiven Web](#). In Staffeldt Sven Klein Wolf Peter, editor, *Die Kodifizierung der Sprache. Strukturen, Funktionen, Konsequenzen*, volume 17 of *WespA - Würzburger elektronische sprachwissenschaftliche Arbeiten*, pages 85–109.
- Aditi Chaudhary, Antonios Anastasopoulos, Zaid Sheikh, and Graham Neubig. 2021. [Reducing confusion in active learning for part-of-speech tagging](#). *Transactions of the Association for Computational Linguistics*, 9:1–16.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- Mengyi Gao, Canran Xu, and Peng Shi. 2021. [Hierarchical Character Tagger for Short Text Spelling Error Correction](#). In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 106–113, Online. Association for Computational Linguistics.
- Yekyung Kim. 2020. [Deep Active Learning for Sequence Labeling Based on Diversity and Uncertainty in Gradient](#). In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 1–8, Suzhou, China. Association for Computational Linguistics.
- Christian Lang, Ngoc Duyen Tanja Tu, Roman Schneider, and Anna Volodina. 2023a. [Projektvorstellung – Sprachanfragen. Empirisch gestützte Erforschung von Zweifelsfällen](#). In *9. Tagung des Verbands "Digital Humanities im deutschsprachigen Raum" DHd 2023*, Trier, Luxemburg.
- Christian Lang, Ngoc Duyen Tanja Tu, and Laura Zeidler. 2023b. [Making Non-Normalized Content Retrievable – A Tagging Pipeline for a Corpus of Expert–Layperson Texts](#). In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 239–244, Portugal. NOVA CLUNL.
- David D. Lewis and William A. Gale. 1994. [A Sequential Algorithm for Training Text Classifiers](#). Publisher: arXiv Version Number: 2.
- Warren S. McCulloch and Walter Pitts. 1943. [A logical calculus of the ideas immanent in nervous activity](#). *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- Seyed Morteza Mirbostani, Yasaman Boreshban, Salam Khalifa, SeyedAbolghasem Mirroshandel, and Owen Rambow. 2023. [Deep Active Learning for Morphophonological Processing](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 793–803, Toronto, Canada. Association for Computational Linguistics.
- Puria Radmard, Yassir Fathullah, and Aldo Lipani. 2021. [Subsequence Based Deep Active Learning for Named Entity Recognition](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4310–4321, Online. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2017. [Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.
- Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2022. [A Survey of Deep Active Learning](#). *ACM Computing Surveys*, 54(9):1–40.
- Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. [Active Learning for Sequence Tagging with Deep Pre-trained Models and Bayesian Uncertainty Estimates](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1698–1712, Online. Association for Computational Linguistics.
- Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. [Deep Active Learning for Named Entity Recognition](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.
- Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian, and Leonid Zhukov. 2022. [Towards Computationally Feasible Deep Active Learning](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1198–1218, Seattle, United States. Association for Computational Linguistics.
- Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. [A Survey of Active Learning for Natural Language Processing](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Multimodal Conversational Interactions for Facial Composites: A Case for Stateful Prompt Orchestration

Rico Städeli and Roman Leu and Jasmin Heierli and Max Meisterhans
and Elena Gavagnin and Alexandre de Spindler

Zurich University of Applied Sciences, Winterthur, Switzerland

{staedric, leurom01}@students.zhaw.ch, {heej, meix, gava, desa}@zhaw.ch

Abstract

The advent of advanced language models has raised expectations for conversational interactions with information systems, but effectively controlling these models to achieve nuanced conversational behavior remains a challenge. This paper introduces PROMISE, a framework that uses state machine modeling to enable dynamic prompt orchestration to facilitate complex interactions. We illustrate PROMISE’s application using a multimodal dialogue scenario designed to assist witnesses in generating facial composites of suspects. Our demonstration validates the framework’s feasibility and its utility to enable meaningful and complex conversational interactions.

1 Introduction

While the development of increasingly powerful language models (LM) raises expectations for more sophisticated conversational behaviours, directing LM behaviour remains challenging (Hadi et al., 2023). There is therefore a growing need to investigate the feasibility and utility of LM applications.

The capabilities of LMs in extracting structured information from text and generating text from structured information are key enablers for designing and specifying conversational interactions. For example, the extraction of user intents and associated parameters, and the generation of responses from intent-specific, structured data, have long been fundamental in developing conversation flows for assistive conversational agents.

While this dialog management approach is effective for answering questions and executing simple commands, existing platforms such as Google DialogFlow and IBM Watson faced limitations when developing more complex interactions. In DialogFlow, the intertwining of numerous contexts and intents can make it difficult to avoid inadequate intent matches. In Watson, the multitude of logical conditions attached to each step can lead

to an extended, often unwieldy tree of conversation possibilities. As the complexity of the dialogs increases, the adaptability of both platforms decreases, making it more difficult to integrate new requirements without great effort.

Consequently, the advent of LMs not only raises the question of how they can be utilized to support designing conversational flows, but also how to harness their advanced capabilities for more complex conversational interactions. Training an LM from scratch to serve a specific purpose is resource-intensive and often impractical for typical development projects. Although fine-tuning can tailor LM responses, it also demands meticulous data preparation, making fast, iterative experimentation difficult. Ultimately, neither approach fully addresses the inherent challenges arising when complex interactions are designed, implemented in variants, and improved iteratively.

Leveraging LMs’ zero- and few-shot learning abilities, more efficient approaches for LM control were developed which are commonly referred to as *prompt engineering* (Korzynski et al., 2023; White et al., 2023). While many prompt articulation strategies have been developed (Wei et al., 2022; Fernando et al., 2023; Chu et al., 2023), this alone cannot ensure consistent LM behaviour in complex interactions. Overly detailed prompts that cover the entire interaction may lead to confusion in sequences or levels of partial interactions. Conversely, overly broad prompts risk missing expected responses, may induce erroneous responses, and introduce vulnerabilities (Mozes et al., 2023).

To mitigate the challenges met when using prompts for complex interactions, Wu et al. 2024 proposed a framework (PROMISE) that follows the idea of segmenting complex tasks into sequences of simpler tasks. This was shown to enhance control and predictability while harnessing LM capabilities (Helland et al., 2023). Complex prompts are thus broken down into separate, more specific

prompts, increasing the predictability of LM behaviour while leveraging conversational skills using existing prompting techniques.

PROMISE uses concepts of state machine modeling as a means to orchestrate partial prompts. Conversational behaviour in interaction states, triggers and guards of state transitions, and actions performed when transitioning, can all be implemented with prompts.

In this paper we present a use case for complex and multimodal conversational interactions involving image generation. The idea is to assist witnesses of a crime in recalling and describing the appearance of a person they encountered. Such descriptions are then used to generate visual representations of that person, potentially aiding in the identification of suspects.

In the following Sect. 2, we highlight the challenges of this application domain that necessitate a multimodal approach. We then identify specific requirements for conversational behaviour in Sect. 3. Section 4 details how the PROMISE framework facilitates the realisation of these requirements. Our validation is twofold: Section 5 focuses on the demonstration of the framework-enabled feasibility of the witness assistance application, which serves as a proof of concept. Following this, Sect. 6 shifts focus to the utility of the application, showcasing its effectiveness in facilitating complex conversational interactions. We conclude with final thoughts in Sect. 7.

2 Background

Conversations aimed at eliciting specific information can be considered examples of complex interactions. These usually require an actively managed dialog that dynamically guides the user to obtain the necessary information. An example of this type of interaction is the creation of facial composite images, for which detailed descriptions of a person’s appearance must be captured. In such cases, specific characteristics of a person must be recorded during a conversation, whereby the conversation may deviate from the course or contain irrelevant details. As the witness is confronted with resulting images, the description obtained so far may have to be supplemented or corrected.

At present, the collaborative elaboration of facial composites is a highly intricate process involving various technical and psychological aspects. It faces challenges at each stage of translating a text-

based description based on human memory into a visual representation of a person (Wells and Hasel, 2007). Although the problem is inherently multimodal and requires the interplay of textual and visual components, current research and efforts to make progress have largely focused on these two elements separately, resulting in unsatisfactory solutions from both visual and linguistic perspectives (Jalal et al., 2023).

From a visual standpoint, the process currently involves either a forensic artist creating a sketch, or an eyewitness iteratively composing a face using specialized software. In both cases, the process is suboptimal as it often leads to a static and unrealistic representation of the suspect (Jalal et al., 2023). Recently, generative deep learning has been introduced, primarily focusing on translating sketches into photographs, predominantly employing generative adversarial networks such as Pix2Pix and CycleGans (Wang et al., 2018a,b; Zhu et al., 2017).

From a linguistic standpoint, the challenge becomes more significant because facial descriptions are frequently affected by noisy information from eyewitnesses. As a matter of fact, sketch-based facial recognition relies primarily on a static textual description given by the witness, which could be inaccurate in the first place and, secondly, does not offer any confidence estimate for each provided feature. In this respect, much linguistic research in the field has concentrated on developing hierarchical analytical methods and frameworks that leverage linguistic theory such as part-of-speech and attribute ontology to effectively extract relevant facial attributes from given descriptions (Karczmarek et al., 2017; Khan and Jalal, 2020).

While both types of approaches address relevant challenges, neither of them seeks to explore how their combination can be used to better assist witnesses in recalling information. In the following, we therefore propose an innovative multimodal conversational interaction that dynamically iterates through linguistic utterances and visual representations to help witnesses recall relevant information.

3 Requirements

We now identify requirements to the conversational behaviour of an assistant supporting witnesses of a crime in recalling and describing the appearance of a person they encountered.

The example conversation in Fig. 1 showcases how descriptive information can be elicited from

a witness. The assistant (light and dark green) uses open-ended questions and empathetic dialogue to create a supportive environment. In a first phase (light green), the witness is guided to provide enough descriptions to generate an initial image, which is then presented (indicated in 1st squared brackets). This initiates a second phase (dark green), where the witness is asked for feedback on the image presented. A new image is then generated based on this feedback, and presented again (indicated in 2nd squared brackets). While this second phase may be repeated, the interaction shall conclude when the witness cannot recall any further information. (light green at the end).

This interaction comprises three phases, which must transition into one another if certain conditions are met. The starting phase needs to conclude when the witness has provided enough descriptive information for an initial image to be generated. The image is generated as part of the transition and presented to the witness to initiate the second phase. This second phase can be left for two reasons. One reason is that the witness provides additional, corrective information, which triggers the generation of a new image. When this new image is presented to the witness, the interaction transitions back to the second phase. The other reason is that the witness no longer suggests any changes. At this point, the interaction transitions to a final phase, in which the witness is thanked and bid goodbye.

To support the generation of an image, transitions need to include an extraction of all descriptive information provided throughout a defined segment of the conversation. For example, the following JSON object should result from such an extraction out of the starting phase of the conversation in Fig. 1.

```
{ "Person": {
  "gender": "Female",
  "height": "5'6\"",
  "build": "Medium",
  "demeanor": "Relaxed, cheerful",
  "hair": {
    "color": "Light brown",
    "style": "Long, loose, wavy"
  },
  "eyes": "Light-colored, possibly
  blue or green",
  "facialFeatures": "Friendly face
  with a big smile",
  "distinctiveMarks": "None"
},
"Environment": {
  "Location": "City, downtown area",
  "Time": "Day"
}}
```

Accordingly, the following JSON object should be extracted from the second phase.

```
{ "hairColor": "caramel brown",
  "eyeColor": "clear blue",
  "earrings": "none" }
```

While the utterances produced by the assistant shown in Fig. 1 demonstrate the benefit of using an LM to generate them, several challenges arise when controlling the LM with a single prompt: Consistently distinguishing different interaction phases, appropriately transitioning between these phases according to specified characteristics of the conversation, comprehensively extracting information from specific conversation segments, all the while promptly following user requests to end the interaction at any time. We are therefore presenting the use of PROMISE as a means of overcoming these challenges while enabling the beneficial use of LMs in the following Sect. 4.

4 Implementation

With PROMISE, the conversational interaction exemplified in Fig. 1 is modelled by a state machine such as in Fig. 2. The state machine consists of an outer state in light gray, an initial node to its left, and a final node to its right. Transitions lead from the initial node to the outer state, and from the outer state to the final node. Transitions depend on triggers and guards, and actions are executed when followed. The outer state contains a sequence of inner states which also starts with an initial node and has its own final node. In between, there are two states shown in green and blue, each specifying the conversational behaviour in the first two interaction phases identified in the previous Sect. 3. The third phase thanking the user and bidding goodbye is specified by the inner final node.

Using PROMISE, states are annotated with a *state prompt* such as "You are having a conversation with a user who ..." for the outer state, "Start by asking open-ended questions about ..." for the green inner state, and "Continue the conversation to refine ..." for the blue inner state. Such state prompts will be used to control the LM while the interaction is in one of these states. Similarly, transitions are also annotated with prompts, for example as indicated by "Details provided", "[]", and "(Extract details, Generate image)" in the case of the transition between the two inner states. These prompts control the LM when analysing the conversation in terms of transition triggers, guards, and

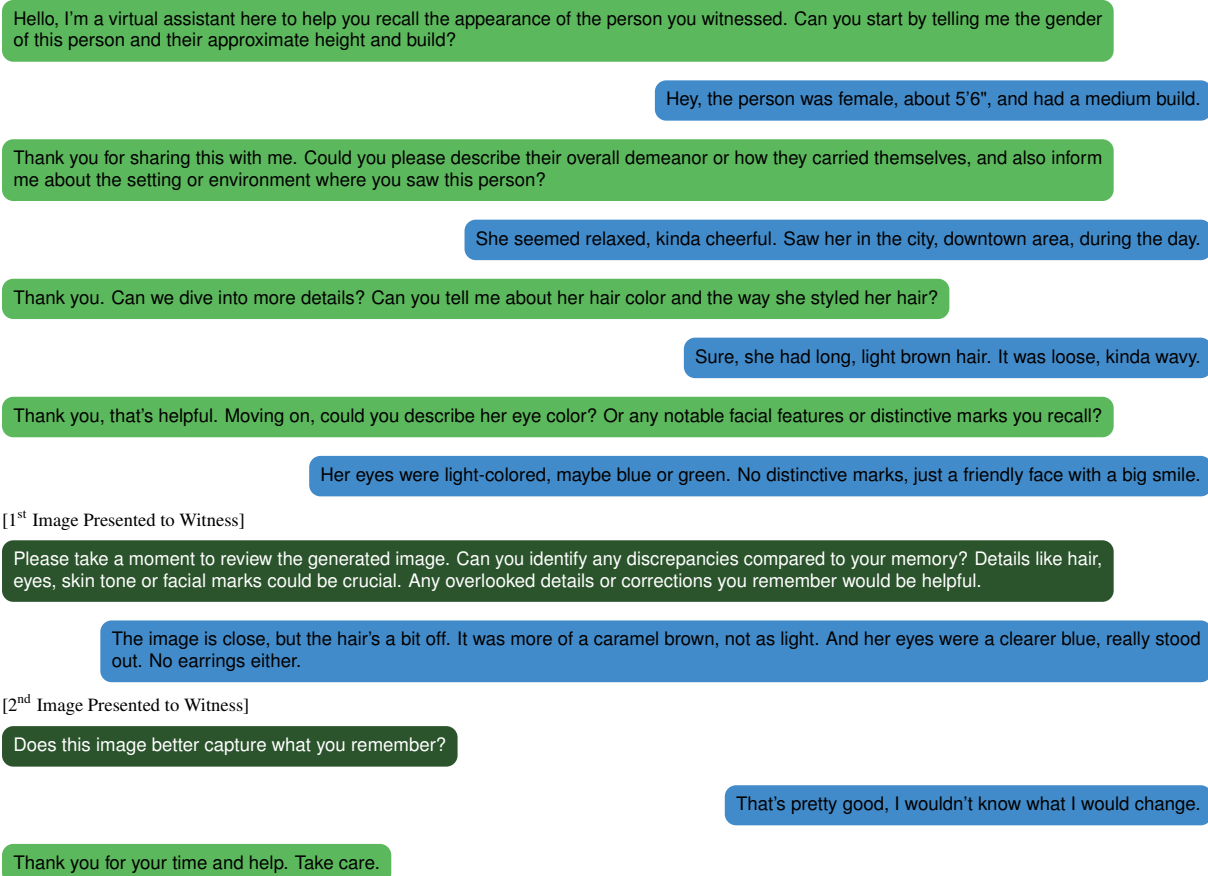


Figure 1: Multi-Phase Elicitation: Assistant (1st: Light Green, 2nd: Dark Green, 3rd: Goodbye) & Witness (Blue)

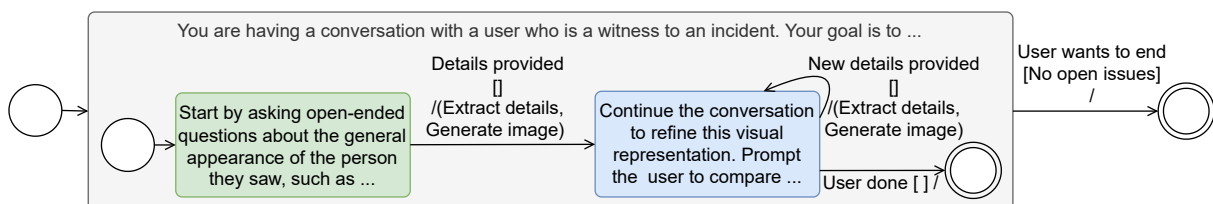


Figure 2: Conversational Interaction Design for Multi-State Elicitation

actions, respectively. As shown next, PROMISE transparently composes more complex prompts by dynamically assembling the prompts associated with states and transitions.

When the interaction is started, the initial node is used to identify the first interaction state. In this case, the initial node points to the outer state, in which the inner initial node points to the green inner state. In this green state, PROMISE will compose a prompt P_c by concatenating the outer state prompt P_{os}

$$P_{os} = \text{"You are having a conversation with ..."}$$

with the inner state prompt P_{is}

$$P_{is} = \text{"Start by asking open-ended ..."}$$

and the *state opening prompt* P_{iso}

$$P_{iso} = \text{"Begin the conversation by ..."}$$

such as to obtain the composed prompt P_c

$$\begin{aligned} P_c &= P_{os} + P_{is} + P_{iso} \\ &= \text{"You are having ... Start ... Begin the ..."} \end{aligned}$$

The state opening prompt, an optional extension of the state prompt, is used when the state is configured to start the conversation, as demonstrated in this example.

The composed prompt P_c is then used to instruct the LM. In the example conversation above, the LM completion returned the utterance "Hello, I'm

here to help you ...", which opens the conversation with the witness.

Upon every utterance from the witness, such as the first utterance "Hey, the person was female ...", all outgoing transitions are checked before the LM is used to generate a response to the witness. First, the list of utterances U_{is}^t , which represents the conversation held in this state so far, is extended with the incoming user utterance u_u as follows.

$$\begin{aligned} U_{is}^{t+1} &= U_{is}^t + u_u \\ &= ["Hello, ...", "Hey, the person ..."] \end{aligned}$$

Then, to check a transition, its trigger prompt P_t and guard prompt P_g are used to obtain decisions from the LM about whether the transition should be followed. If it is followed, the action prompt P_a is used to execute the action. The trigger, guard and action prompts are automatically extended with the utterances such as to support decisions based on the conversation so far.

For example, in the case of a transition trigger, the composed prompt

$$\begin{aligned} P_c &= P_t + U_{is}^{t+1} \\ &= "Review the conversation... Determine..." \\ &+ ["Hello, ...", "Hey, the person ..."] \end{aligned}$$

is created to let the LM decide whether the conversation so far contains the information required to generate a prompt for the image generation model. While the first witness response mentions a height, build and how the suspect carried themselves, no information about their hair and facial features has been provided so far. Consequently, this transition trigger does not pass, and the interaction stays in the current state.

Multiple decisions may be attached to a single transition. Each decision may contain a prompt for LM-based evaluation, optionally containing placeholders for data injection. Alternatively, decisions may also be specified with code that implements any other evaluation. In our example, a second decision serves as a transition guard, and instructs the LM to decide whether there are no open questions from the witness that should prevent the current interaction from transitioning unexpectedly.

If the interaction stays in the current state, the state prompt and accumulated utterances are in-

cluded in the newly composed prompt

$$\begin{aligned} P_c &= P_{os} + P_{is} + U_{is}^{t+1} \\ &= "You are having ... Begin the ..." \\ &+ ["Hello, ...", "Hey, the person ..."] \end{aligned}$$

which is used to obtain the subsequent response to the witness from the LM. This response is also appended to the state utterances. As seen in the example interaction in Fig. 1, the conversation therefore stays in the same state as long as the expected information is incomplete. When all the information is provided, the transition decisions pass, and the conversation transitions to the subsequent state attached to the transition. As shown in Fig. 2, the blue state follows the green state. In this blue state, the LM is controlled as described for the green state, but using the partial prompts associated with the blue state.

Transitions may include multiple actions that contain a prompt or code. In our example, there are two actions. The first is to extract the details provided by the witness. The second action will generate an image based on these details extracted. In both cases, the action is a prompt with which the composed prompt

$$\begin{aligned} P_c &= P_a + U_{is} \\ &= "Review the conversation... Extract..." \\ &+ ["Hello, ...", "Hey, the person ..."] \end{aligned}$$

is created and used to instruct the LM such as for extracting details or generating an image. In most cases, the result of an action is an object that is stored in an interaction storage, making it accessible to other states, transition decisions and actions, or surrounding system components.

As opposed to the green state, the blue state has two outgoing transitions. One of them is triggered by the condition "User done", which will be true if the witness has nothing to add to the latest image shown to them. The other transition is triggered by the user providing additional details that can be used to update the image. If this is the case, the additional details will be extracted and used to generate the next image to be shown to the witness as part of this recursive transition.

The outer state simplifies the development of conversational interactions in three ways. First, the outer state prompt P_{os} is transparently prepended to all its inner states. Consequently, developers can avoid redundancies by factoring out common parts

of inner state prompts. Second, the outer state may have its own outgoing transitions which enables multilayered interactions. For example, this outer state has a transition triggered by the user wishing to stop the interaction. This trigger decision is tested with each incoming user utterance in all the inner states. As a result, this transition may be triggered at any point in the whole interaction. Third, the outer state maintains its own list of utterances U_{os} containing all utterances of all its inner states. This enables decisions and actions to be made in the scope of larger conversational contexts.

In summary, PROMISE promotes a separation of concerns when LMs are controlled using prompts. One means of separation results from the support of state-specific prompts and separate, individual prompts for transition decisions and actions. Another means is provided with the ability to factor out recurring prompt parts and reuse them in a common outer state. As a result, it costs less effort to distinguish interaction phases and control their transitions more consistently. The maintenance of state-specific utterances better supports transitions and actions playing out in different interaction segments. Finally, the ability to nest states supports multilayered interactions where different conversation flows are controlled in parallel.

Note that all prompts used in a PROMISE application may feature placeholders in which any text or data may be injected. While this supports the use of prompt engineering for retrieval augmented generation (RAG), a more detailed description of this mechanism is beyond the scope of this paper.

5 Proof-of-Concept Application

This section demonstrates the PROMISE framework’s practical applicability by detailing the resulting witness assistant application. Our aim is to showcase the feasibility of bringing nuanced requirements of multithreaded and multimodal conversational flows to reality. The main functionality of the application is outlined in Fig. 3.

A chat frontend used by witnesses is indicated on the left. While the conversation depicted is the one previously shown in Fig. 1, this front-end showcases the idea of iterative questioning, where the witness is repeatedly presented with images generated from the information they provide, and subsequently updated with the additional information they provide when an image is presented. The extractions in JSON format used for the generation

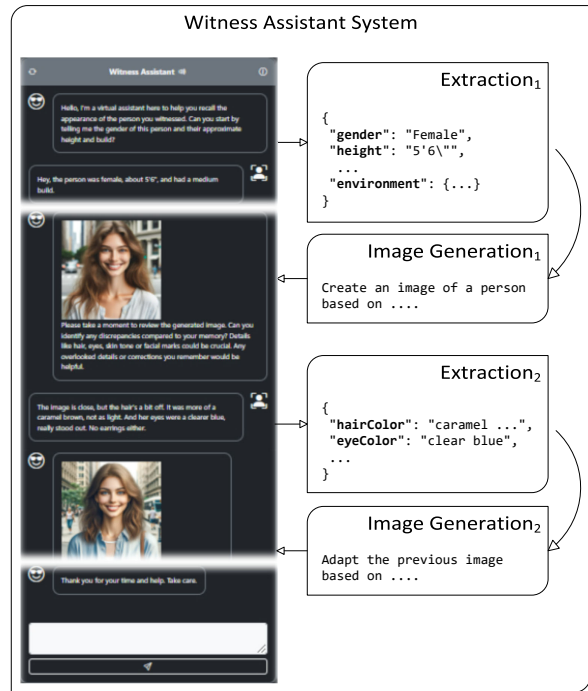


Figure 3: Witness Assistant System

of the images are shown on the right.

The conversational interaction was modeled as shown in Fig. 2 and implemented using the PROMISE API. All prompts for the outer and inner states, as well as the transition triggers and actions were generated using ChatGPT with GPT-4 (OpenAI, 2023) using the following meta prompt.

Assist me in crafting prompts for guiding GPT to function as a chatbot. This chatbot is designed to (...)

For example, the meta prompt was then extended with the specific instruction

I need a prompt that instructs GPT to have a conversation with the witness in order to elicit some key features of the person they've met. (...)

in order to obtain the witness assistant outer state prompt

You are having a conversation with a user who is a witness to an incident. Your goal is to (...)

The fact that this application could be realised with PROMISE demonstrates that the framework effectively supports the development of the required conversational interaction, including the different phases, phase transitions, extractions, image generation and multilayering.

6 Application Validation

Building on the witness assistant application developed using the PROMISE framework, we now turn our attention to the utility of this application. This section is dedicated to evaluating the application's effectiveness in achieving its intended goals, particularly in improving the accuracy and efficiency of information gathering. Through a simulated interaction, we demonstrate how the application not only operationalizes the framework but also enhances the user experience and outcome quality.

The application was tested by using a fictitious witness simulated using ChatGPT with GPT-4 (OpenAI, 2023), who is asked to describe a person visible in a provided picture. For this purpose, we used the following prompt.

```
Generate responses as if you're a user
interacting with a chatbot (...)
I'll provide the chatbot's messages,
and you'll respond as the user (...)
When asked for descriptions, your
answers should reflect what is visible
about the person in this picture.
[Picture of Person Here]
```

The picture shown in Fig. 4 was obtained from Unsplash¹ and added to the previous prompt at the location indicated with the squared brackets. From then on, the assistant's utterances such as

```
Hello, I'm a virtual assistant here to
help you recall the appearance of (...)
```

were provided to obtain the utterance of the simulated witness such as

```
Hey, the person was female, about 5'6",
and had a medium build.
```

The witness utterances were passed back to the assistant, whose responses were in turn passed on to the simulated witness, and so forth.

The resulting interaction shown in Fig. 1 demonstrates that the assistant behaves according to the combination of the outer state and respective inner state prompts. Furthermore, the different green-coloured utterances, each representing a specific state, adequately triggered state transitions due to the transition decision prompts. Finally, the transition action prompts facilitated the extractions of descriptive information resulting in JSON objects shown in Sect. 3. As can be observed, these JSON objects fully capture all the information provided by the simulated witness.

Based on the JSON object extracted from the first phase of the interaction, the following prompt

¹Guilherme Stecanella on Unsplash



Figure 4: Person to be described by simulated witness

was used with DALL-E 3 to generate an initial possible image of the person to be identified:

```
Create visual representations based on
witness descriptions. I'll supply you
with JSON objects detailing the
characteristics of an individual, and
your task is to produce images that
match these descriptions.
[JSON Object Here]
```

The image generated using the first JSON object obtained from the first phase of the interaction is shown in Fig. 5.

The generated image was then passed on to the simulated witness, in order to correct or add previously collected attributes and to obtain another JSON object capturing these changes. The following prompt was then used with DALL-E 3 to generate new versions of a previous image based on the JSON object extracted from subsequent phases of the interaction.

```
The witness was shown an image and
queried about any modifications that
could enhance its similarity to their
recollection. Here is the JSON object
with the witness's suggested
adjustments:
[JSON Object Here]
```

The image generated using the second JSON object obtained from the second phase of the interaction is



Figure 5: Image generated after first phase

shown in Fig. 6. A simple quantitative comparison



Figure 6: Image generated after second phase

using the cosine-similarity of the two generated images using Dino2 (Oquab et al., 2024) embeddings confirms the high-level visual matching (cosine = 0.89) and consistent general agreement with the original picture (cosine = 0.75, 0.79).

One limitation we encountered in using image generation models is that it is difficult to generate multiple images in which only prescribed aspects differ. As can be observed when comparing the images in Fig. 5 and 6, despite the JSON object suggesting no more changes than to the hair and eye colors, and not having earrings (which wasn't present in the first image), the second picture shows a collar, a handbag, earrings, and the background was altered. This is a known fact in generative

computer vision, and it has its origin in the intrinsic diffusion model at the base.

While the ability to generate such images proves the utility of our application, the validity of the individual images as well as their stepwise adaptation largely depends on the capability of the image generation model. Consequently, any in-depth validation of the differences among the generated images is out of the scope of this paper, as it would rather serve as a validation of the specific image generative model employed instead of validating the witness assistant application or the PROMISE framework.

7 Conclusion

The use of language models (LM) to support conversational interactions is promising but challenging. As the complexity of the expected behaviour grows, so does the prompt specifying the behaviour, which increases the likelihood of misbehaviour. We therefore introduce the notion of stateful prompt orchestration which follows the idea of segmenting complex prompts into smaller ones, which can then be combined dynamically, depending on the state of the interaction.

Given the requirements from an application scenario, we presented the use of the PROMISE framework to design and implement a multimodal conversational interaction. PROMISE supports this by leveraging state machine modelling concepts. This enables developers of interactions to orchestrate prompts, not only to harness the LMs capabilities for open-ended conversations, but also to enable complex conversational interactions including conversation flows, flow transitions, extractions, recursive flows, and multilayered interactions.

With the successful development and simulated use of a proof-of-concept application, we demonstrate that PROMISE effectively supports the design and implementation of useful conversational interactions. The resulting application proves the ability of PROMISE to manage different prompts and effectively orchestrate these prompts to enable complex multimodal interactions beyond what is feasible with single-prompt LM applications.

Our next steps involve augmenting the PROMISE framework with persuasive conversational capabilities by providing the means to dynamically select and apply persuasion strategies. These extensions will enable persuasive conversational interactions that deliver demonstrable

benefits in the healthcare sector, for example. As PROMISE is able to respond to defined conversation segments, this will allow the use of different persuasion strategies during the conversation, further enhancing the framework's support for more sophisticated interactions.

References

- Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.
- Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*.
- Solveig Helland, Elena Gavagnin, and Alexandre de Spindler. 2023. [Divide et impera: Multi-transformer architectures for complex NLP-tasks](#). In *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, pages 70–75, Neuchatel, Switzerland. Association for Computational Linguistics.
- Anand Singh Jalal, Dilip Kumar Sharma, and Bilal Sikander. 2023. [Suspect face retrieval using visual and linguistic information](#). *The Visual Computer*, 39(7):2609–2635.
- Paweł Karczmarek, Witold Pedrycz, Adam Kiersztyn, and Przemysław Rutka. 2017. [A study in facial features saliency in face recognition: An analytic hierarchy process approach](#). *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 21(24):7503–7517.
- Mohd. Aamir Khan and Anand Singh Jalal. 2020. [A framework for suspect face retrieval using linguistic descriptions](#). *Expert Systems with Applications*, 141:112925.
- Pawel Korzynski, Grzegorz Mazurek, Pamela Krzykowska, and Artur Kurasinski. 2023. Artificial intelligence prompt engineering as a new digital competence: Analysis of generative ai technologies such as chatgpt. *Entrepreneurial Business and Economics Review*, 11(3):25–37.
- Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D Griffin. 2023. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv preprint arXiv:2308.12833*.
- OpenAI. 2023. [GPT-4 Technical Report](#).
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. [DINOv2: Learning Robust Visual Features without Supervision](#).
- Lidan Wang, Vishwanath Sindagi, and Vishal Patel. 2018a. [High-Quality Facial Photo-Sketch Synthesis Using Multi-Adversarial Networks](#). In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 83–90, Xi'an, China. IEEE Press.
- Nannan Wang, Wenjin Zha, Jie Li, and Xinbo Gao. 2018b. [Back projection: An effective postprocessing method for GAN-based face sketch synthesis](#). *Pattern Recognition Letters*, 107:59–65.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Gary L. Wells and Lisa E. Hasel. 2007. [Facial Composite Production by Eyewitnesses](#). *Current Directions in Psychological Science*, 16(1):6–10.
- Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. [A prompt pattern catalog to enhance prompt engineering with chatgpt](#). *arXiv preprint arXiv:2302.11382*.
- Wenyuan Wu, Jasmin Heierli, Max Meisterhans, Adrian Moser, Andri Färber, Mateusz Dolata, Elena Gavagnin, Alexandre de Spindler, and Gerhard Schwabe. 2024. Promise: A framework for model-driven stateful prompt orchestration. In *Intelligent Information Systems: CAiSE Forum 2024, Limassol, Cyprus, June 3–7, 2024, Proceedings*, volume 520 of *Lecture Notes in Business Information Processing*. Springer International Publishing.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. [Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks](#). In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251.

Sentiment- and Keyword-Controllable Text Generation in German with Pre-trained Language Models

Paulina Aleksandra Żal and Guang Lu
Lucerne University of Applied Sciences and Arts
paulina.zal@hslu.ch
guang.lu@hslu.ch

Nianlong Gu
University of Zurich
nianlong.gu@uzh.ch

Abstract

Controllable Text Generation (CTG) aims to enhance the controllability of language models by providing the user with mechanisms to control attributes such as sentiment, topic, and style in the generated text. Despite the progress in English text generation, the potential challenges of CTG in other languages such as German remain largely underexplored. This study investigates CTG with Pre-trained Language Models (PLMs), i.e. GPT-2 in German, to achieve both sentiment and keyword control. To this end, Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) were first used for sentiment control and then the Logit Modification Mechanism for keyword forcing. Our initial study has shown that the combination of these techniques is promising for CTG in German. More specifically, text generation works well with respect to the given sentiment, while improvements are still needed to ensure coherence and fluency in the generated text with specific keywords. Compared to forcing nouns, forcing sentiment-based keywords degrades the quality of the texts. This means that sentiment and keyword control need to be combined in a single training process to achieve better concurrent control. The code is publicly available to reproduce the results¹.

1 Introduction

The rise of Pre-trained Language Models (PLMs) has revolutionized text generation, making a substantial advancement in fields like e-commerce and marketing. These models, trained on extensive text corpora, are capable of generating new content with a deep understanding of language (Zhang et al., 2022). Despite their remarkable abilities, the probabilistic nature of these models means the generated texts may not always align perfectly with the intents of users.

¹<https://github.com/polie94/SwissText2024>

In response to this challenge, various methods for Controllable Text Generation (CTG) have emerged, including but not limited to fine-tuning, retraining, and post-processing (Zhang et al., 2022). These strategies, while effective to some extent, often fall short of granting users full control over the sentiment or topic attributes of the generated text. In contrast, the “plug-and-play” approach described by Zhu et al. (2022) offers a more interactive approach by incorporating external inputs such as keywords or topic labels to intuitively control the generated text. This approach not only enhances control over the final output but also streamlines text creation, enhancing efficiency and aligning language models with human preferences.

Recent CTG research has mainly concentrated on English, leaving a gap to other languages, like German. German CTG presents unique challenges due to its complex grammar and compound words, raising questions about how existing techniques generalize to German and the interplay between sentiment and keyword control.

To address these challenges, this study focuses on adapting the German GPT-2 model (Minixhofer et al., 2022) for “plug-and-play” manner. Given a sentiment token, a set of keywords and a prompt, we expect the model to complete the given prompt with respect to the specified sentiment and utilizing the given keywords. Our approach involves developing a sentiment classifier, conducting Supervised Fine-Tuning (SFT), and utilizing Reinforcement Learning (RL) for further refinement. Additionally, we adapted a Logit Modification Mechanism from Pascual et al. (2020) to incorporate specified keywords into the generated content. The contributions of this work are twofold:

- We evaluate the synergy between the fine-tuning and post-processing in CTG, showing their mutual influence on the generated text.
- We identify the gap in a performance evaluation when combining the sentiment and key-

word control methods, shedding light on the challenges of CTG in German.

2 Related Work

Since the introduction of the *Transformer* architecture and subsequent PLMs like GPT (Radford et al., 2018) and T5 (Raffel et al., 2020), researchers have explored various methods for CTG. The generation can be controlled at different stages of the model lifecycle, e.g. during the fine-tuning or post-processing stage (Zhang et al., 2022).

Recent advancements in CTG have leveraged various techniques to enhance the quality, fluency, and safety of generated text. Ziegler et al. (2019) fine-tuned a model with RL from human preferences, incorporating a penalty to maintain text fluency. Schulman et al. (2017) proposed a Proximal Policy Optimization (PPO) algorithm for CTG, as demonstrated by von Werra (2023a). The approach exemplified by Krause et al. (2020) offers efficient guidance for text generation, promoting safer and less biased content. Pascual et al. (2021) introduced the Keyword2Text method for creating text with constraints, favoring words similar in meaning to specified targets. Pascual et al. (2020) addressed topic control by stipulating the presence of specific words in the generated text. Additionally, Dathathri et al. (2019) combined PLMs with attribute classifiers for topic and sentiment control without additional training.

3 Methodology

Figure 1 outlines our proposed approach, starting with data cleaning and the creation of the BERT-based sentiment classifier. We then fine-tune the German GPT-2 model on instruction-based data, integrating the sentiment classifier into the RL process using PPO. We then introduce keyword control to the fine-tuned language models through logit modification during decoding and we experiment with various decoding strategies.

3.1 Data

Guhr et al. (2020b) introduced a sentiment classifier tailored for analyzing German text. They compiled a fresh dataset specifically for sentiment analysis in German. Our study uses specific segments from their dataset (Guhr et al., 2020a), including hotel reviews², movie reviews³, and data

²holidaycheck.de

³filmstarts.de

on emotions.

Similarly to the approach of Guhr et al. (2020b), the data for holiday and movie reviews are categorised as positive if the corresponding ratings were greater than three and as negative if they were less than three. Neutral reviews are those with a rating of three. The analysis of the used data is shown in Tab. 1. The example of data is shown in Appendix B.

3.2 Sentiment Classifier

We fine-tune a German BERT (Chaumond, 2020) as a sentiment classifier. To assess model performance and prevent overfitting, evaluation is conducted on the unseen test data. The model encodes input text and feeds the classify token (CLS) embedding into a linear layer, producing a scalar sentiment score that distinguishes between positive and negative sentiments. After fine-tuning the BERT model using cross-entropy loss for sentiment prediction, the classifier is employed in RL optimization and serves as a sentiment quality evaluator for generated texts.

3.3 Fine-Tuning

To address potential challenges faced by the German GPT-2 model in adapting to new, specific domains and tasks, utilizing an SFT approach can improve its performance.

The SFT is performed on texts starting with a sentiment-specific instruction in the form: *[sentiment token] text*, where the sentiment is labeled as: *[positive]* or *[negative]*. The *text* is the review from the dataset. In our experiments we remove the neutral sentiment to reduce the number of cases.

3.4 Proximal Policy Optimization

We use PPO (Schulman et al., 2017) to improve sentiment control in our Language Model (LM). Using sentiment scores from the classifier as rewards, we fine-tune the LM, previously adjusted through SFT with the same dataset, for optimized sentiment generation. Our approach introduces two reward functions for positive and negative sentiment generation, calculated based on the logits from the sentiment classifier. By guiding the LM to produce text with specified sentiment, the classifier assesses sentiment and selects the corresponding logit as a reward. This, along with input queries and response tensors, informs the PPO algorithm, refining the model’s capacity to generate sentiment-aligned text.

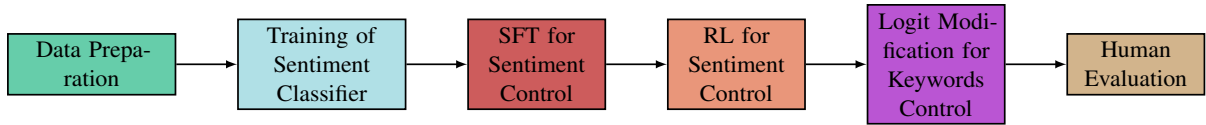


Figure 1: CTG Pipeline Used in this Work.

Dataset	No. reviews	Positives	Negatives	Neutral	Rating
Filmstars	71,126	40,015	15,611	15,500	0.0 - 5.0
Holidaycheck	4,831,290	3,995,507	388,741	447,042	1.0 - 6.0
emotions	1,306	188	1,090	28	Categorical

Table 1: Overview of the Used Datasets.

3.5 Logit Modification Mechanism

We adopt the Directed Beam Search (DBS) method (Pascual et al., 2020) to control text generation using keywords. DBS incorporates a mechanism for logit modification. Its core idea is to have an PLM generate a new token resembling given keywords. Pascual et al. (2020) calculate similarity between model logits and keywords, utilizing GloVe embeddings (Pennington et al., 2014) for the numerical representation of words.

The adapted logits are calculated by summing the original logits generated by the LM, weighted by the similarity between the token and the keyword, scaled by the parameter λ . After applying the softmax function to convert them into probabilities, one token is sampled from the resulting distribution. Their method was based on the version of beam search, and used stemming to check whether the generated word belongs to the keywords. If a keyword is detected, it switches to the next keyword in the set (Pascual et al., 2020).

Pascual et al. (2020) introduced two text generation functions. The first function ensures that all keywords are used. It encodes the text, computes similarities between keywords and vocabulary, and adjusts logits to favor similar words. Softmax converts logits to probabilities, and sampling selects the word index. This process repeats for each keyword, selecting the index with the highest probability. If all keywords are used, the second function is used and subsequent words are generated without control.

Different from the original DBS algorithm (Pascual et al., 2020) that takes keywords in order, in this study, we extend this mechanism by independently incorporating keywords into the text without considering their order of appearance. This enhances flexibility in generated outputs. We use the

Logit Modification Mechanism to calculate similarities between the generated token and all keywords and then select the best performing one. If the next generated word and the keyword have the same lemma, we count it as an occurrence and delete this word from the keyword set so that the model only considers the remaining keywords.

The lemmatization method is established using *Tree Tagger* (Schmid, 2013, 1999). We convert the vocabulary to the embedding space using FastText (Bojanowski et al., 2017).

Instead of DBS we explore Greedy Search, Top- k , and Top- p as decoding strategies. These strategies are integrated into functions for text generation with and without control. Their task is to sample the next token from the adapted logits.

4 Experiment

4.1 Sentiment Classifier

The sentiment classifier is trained on cleaned data with capitalization, punctuation, and stop words. Tokenization truncation is used to standardize input length, converting text of variable length into fixed-size tensors.

The model is trained using specific parameters: a learning rate of $2 \cdot 10^{-5}$, processing 16 samples per batch during both training and evaluation. Training is conducted for one epoch, and model performance is evaluated after each epoch. Additionally, a weight decay regularization parameter of 0.01 is applied.

4.2 Sentiment Control

4.2.1 Supervised Fine-Tuning

We fine-tuned the German GPT-2 model on the instructions dataset with a learning rate of $5 \cdot 10^{-5}$. In order to analyze the influence of different training epochs on the performance, we compared two

models that were trained for one epoch (SFT1) and two epochs (SFT2) respectively.

We fine-tuned the LMs using the *TRL* (von Werra et al., 2020) library from *Hugging Face*. The performance of the fine-tuned model is evaluated in terms of fluency and sentiment agreement. Evaluation metrics include perplexity for fluency, coherence score (Röder et al., 2015; Kapadia, 2019), and classification metrics for sentiment agreement.

4.2.2 Reinforcement Learning

Two models (SFT1, SFT2) after SFT from Section 4.2.1 are fine-tuned using RL for one (RL1) and two epochs (RL2), which resulted in four new models: SFT1RL1, SFT1RL2, SFT2RL1, SFT2RL2.

We utilized the PPO algorithm implemented by von Werra et al. (2020) following the approach presented by von Werra (2023b), with mini-batches consisting of 16 examples over 51,200 steps. The training learning rate was set to $1.41 \cdot 10^{-5}$. During PPO optimization, texts are generated in mini-batches with randomly chosen sentiment tokens and text beginnings. Evaluation metrics include perplexity for fluency, coherence score, and classification metrics for sentiment agreement.

We use the same sentiment classifier for both PPO training and evaluation, since we treat the generated text as a new, unobserved test data.

4.3 Keyword Control

The generated text after introducing the keyword control is evaluated in terms of perplexity, sentiment accuracy, and success rate. Success rate was proposed in Pascual et al. (2020). This metric calculates the number of keywords (n_{key}) that occur in the text divided by the total number of specified keywords (n_{total}) as in Eq. 1.

$$success\ rate = \frac{n_{key}}{n_{total}} \quad (1)$$

4.4 Human evaluation

We conducted a human evaluation with 18 example texts (A1 to F3 in Tab. 2). The survey is divided into three color groups: green, yellow, and violet. Each color group contains 6 text examples that span across diverse generation settings (different sentiments and keywords). To lower the workload for every participant, each of the participants selects only one color group that leads to the pre-selected examples. Every text starts with the prompt: “Wir waren in diesem Hotel”(“We stayed at this hotel”).

and is written accordingly to the sentiment and possible keywords.

In Tab. 2, examples in rows A and B are designed to evaluate the mutual influence of the text sentiment and keywords sentiment. In order to keep the sentences short due to computational resources, we selected the size of the keyword set as three. The sets were created with two sentiment-carrying adjectives and one noun not carrying sentiment. An example of a set containing negative keywords is (“hässlich”, “langweilig”, “Lobby”), meaning (“ugly”, “boring”, “Lobby”). On the other hand, (“schön”, “hässlich”, “Lobby”), where “schön” means “beautiful”, is an example of a set of mixed keywords. The corresponding texts are generated with the model SFT1RL1 with Top- k , since it performs best in terms of perplexity and success rate when generating texts with nouns.

Rows C and D are created to evaluate the influence of fine-tuning on sentiment control, comparing the performance of models GPT-2, SFT1, and SFT1RL1.

Rows E and F are designed to investigate the influence of keywords (nouns only) on the quality of the text generation using selected decoding approaches: Greedy Search, Top- k , and Top- p . The texts are generated with the SFT1RL1 model with the same keywords (nouns) across all sentiments and decoding strategies.

For evaluation, a standardized set of statements (Appendix J, Tab. 23) is prepared, with users rating their agreement on a Likert-type scale from 1 to 5. The scale ranges from “I strongly disagree” (1) to “I strongly agree” (5). Stronger agreement indicates better performance, assuming that participants are more convinced of the statement’s validity. We decided to use one scale for all items to not confuse the raters with a variety of different answers types. For some statements we need to recalculate the scores to ensure the same scale during the evaluation (Appendix J.1).

To assess results, statements are grouped into latent variables for sentiment, fluency and coherence. For the evaluation of the internal consistency within each category we use Cronbach’s Alpha (Cronbach, 1951) (Appendix J.2).

5 Evaluation

5.1 Supervised Fine-Tuning

To evaluate the performance of fine-tuned German GPT-2 model, we created prompts using sentiment

	1	2	3	Row-wise Configurations
A	key: negative	key: positive	key: mixed	sent: positive
B	key: negative	key: positive	key: mixed	sent: negative
C	GPT-2	SFT1	SFT1RL1	sent: positive
D	GPT-2	SFT1	SFT1RL1	sent: negative
E	Greedy Search	Top- k	Top- p	sent: positive key: nouns
F	Greedy Search	Top- k	Top- p	sent: negative key: nouns

Table 2: Survey Design. The Abbreviations “key” and “sent” Mean Keyword and Sentiment Respectively.

Model	Perplexity (\downarrow)	Coherence Score (\uparrow)
GPT-2	<u>1.12</u>	0.863
SFT1	<u>1.12</u>	0.863
SFT2	1.13	<u>0.864</u>

Table 3: Quality of Texts Created with German GPT-2 Model and SFTn Models.

Model	Accuracy (\uparrow)	Precision (\uparrow)	Recall (\uparrow)	F1-score (\uparrow)
GPT-2	0.68	0.80	0.68	0.64
SFT1	0.84	0.88	0.84	0.84
SFT2	<u>0.87</u>	<u>0.90</u>	<u>0.87</u>	<u>0.87</u>

Table 4: Sentiment of Texts Created with German GPT-2 model and SFTn Models.

tokens and the first ten words from the original review. The evaluation comprises 50 positive and 50 negative texts.

The effect of SFT on perplexity appears negligible (Tab. 3). The models retain their capacity to generate text with fluency comparable to that of the original German GPT-2 model.

As the aim of the process is to enable control over the sentiment in the text generation process, the performance in this matter is shown in Tab. 4. We observed a clear improvement in the model’s capability of sentiment control after one epoch of SFT (SFT1 versus GPT-2), while further training for additional epochs only led to marginal improvement (SFT2 versus SFT1).

Model	Perplexity (\downarrow)	Coherence Score (\uparrow)
SFT1RL1	1.12	0.864
SFT1RL2	1.12	0.866
SFT2RL1	1.12	0.865
SFT2RL2	<u>1.11</u>	<u>0.869</u>

Table 5: Quality of Texts Generated by SFTnRLm.

Model	Accuracy (\uparrow)	Precision (\uparrow)	Recall (\uparrow)	F1- score (\uparrow)
SFT1RL1	0.99	0.99	0.99	0.99
SFT1RL2	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>
SFT2RL1	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>
SFT2RL2	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>

Table 6: Sentiment Scores among SFTnRLm Models.

5.1.1 Results of Reinforcement Learning

We evaluated the performance of SFTmRLn on 100 generated examples, of which 50 examples were given the positive sentiment token and 50 with the negative sentiment token. The input prompt includes the first 10 tokens from the original text along with its sentiment token. Examples of the generated texts are shown in the Appendix F.

As shown in Tab. 5, perplexity and coherence scores are similar among the presented models and do not differ substantially from the scores obtained in the SFT process (Tab. 3). These results imply that there is no obvious negative influence of RL on fluency. In addition, the texts generated by the LM optimized with RL perform perfectly across all sentiment metrics (Tab. 6).

Model	Greedy Search	Top- k , $k = 15$	Top- p , $p=0.5$
SFT1RL1	<u>1.09</u>	<u>1.09</u>	1.10
SFT1RL2	<u>1.09</u>	<u>1.09</u>	<u>1.09</u>
SFT2RL1	1.15	1.10	1.12
SFT2RL2	1.14	1.10	<u>1.09</u>
Avg. Fine-Tuned Models	1.12	1.09	1.10
GPT - 2	1.78	2.01	2.24

Table 7: Perplexity of Texts Decoded with Different Strategies and Forcing of Nouns.

Model	Greedy Search	Top- k , $k = 15$	Top- p , $p=0.5$
SFT1RL1	63%	66%	<u>65%</u>
SFT1RL2	59%	65%	62%
SFT2RL1	50%	65%	62%
SFT2RL2	55%	60%	61%
Avg. Fine-Tuned Models	57%	64%	63%
GPT - 2	<u>75%</u>	<u>73%</u>	<u>65%</u>

Table 8: Influence of Decoding Strategy on Success Rate

5.2 Performance of Keyword Control

5.2.1 Evaluation Based on Nouns

We investigated the influence of keyword control by forcing nouns on all models discussed in Section 4.2.2 during text generation. Through sampling texts, values for k and p are chosen to balance the automatic metric of success rate with subjective evaluations of diversity and coherence. The optimal values are determined to be $k = 15$ and $p = 0.5$.

The models’ performance is evaluated using a list of 50 keyword sets each containing three randomly selected nouns related to hotel reviews. For example, a keyword set for "location", "lobby", and "fitness" might be translated as: (“Lage”, “Lobby”, “Fitness”). These keywords are sentiment-neutral. The evaluation prompt consists of a randomly chosen sentiment token, either “[*positive*]” or “[*negative*]” along with the sentence “Wir waren in diesem Hotel” (“We stayed at this hotel”). This setup is consistent across all decoding strategies and models, allowing comparison between different decoding strategies. Examples of the generated texts are shown in the Appendix G.

Fine-tuned models show similar perplexity (1.09 to 1.15, Tab. 7). Top- p and Top- k outperform Greedy Search. The German GPT-2 model exhibits

Model	Greedy Search	Top- k , $k = 15$	Top- p , $p=0.5$
SFT1RL1	0.98	<u>1.00</u>	<u>1.00</u>
SFT1RL2	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>
SFT2RL1	0.96	<u>1.00</u>	0.98
SFT2RL2	0.96	0.98	<u>1.00</u>
Avg. Fine-Tuned Models	0.98	1.00	1.00
GPT - 2	0.42	0.42	0.40

Table 9: Sentiment Accuracy for Different Decoding Strategies.

Model	Greedy Search	Top- k , $k = 15$	Top- p , $p=0.5$
Perplexity	1.09	1.09	1.10
Sentiment Accuracy	0.98	1.00	1.00
Success Rate	63%	66%	65%

Table 10: Perplexity of Texts Decoded with Different Strategies and Forcing of Nouns.

higher perplexity across all decoding strategies than the fine-tuned models, signifying the effectiveness of fine-tuning.

The evaluation of the success rate is presented in Tab. 8. The success rates vary between 50% and 66% for all models. The more the SFT models undergo training with RL, the lower is the success rate, independently from the used decoding strategy. Although Top- k and Top- p surpass Greedy Search, GPT-2 achieves its best Success Rate with Greedy Search. Interestingly, we noticed that vanilla GPT-2 achieved a higher success rate than models fine-tuned to enhance sentiment control, implying a domination of given sentiment over logit modification in these fine-tuned models.

However, keyword control with nouns does not impact sentiment accuracy, as shown in Tab. 9. The fine-tuned models consistently generate text with the correct sentiment.

The SFT1RL1 is selected for the analysis of the performance of text generation based on sentiment-carrying words and survey evaluation. The performance of this model is presented in Tab. 10.

5.2.2 Evaluation Based on Sentiment-Carrying Adjectives

Combining sentiment tokens with inherently sentiment-carrying keywords like “hässlich” (“ugly”) and “schön” (“beautiful”) poses challenges for CTG. An experiment with model SFT1RL1 using Top- k decoding was conducted to

measure this influence. 180 texts were generated, 90 with the positive sentiment token and 90 with the negative one. Each text was paired with a keyword list containing two adjectives and one noun. Examples of the generated texts are shown in Appendix H. Results are shown in Tab. 11.

Comparing to using only nouns, incorporating sentiment-based keywords worsens several metrics: perplexity increases by 0.22 (from 1.09 in Tab. 7 to 1.31), indicating reduced fluency. Success rate drops by 29 percentage points (from 66% to 37%), showing models are less likely to include sentiment-carrying keywords. Sentiment accuracy slightly decreases (from 1.00 to 0.97), suggesting a preference for preserving sentiment over including keywords. Positive keywords yield the highest success rate but the poorest perplexity. Negative and mixed keywords have better perplexities but worse success rates. We assume that positive keywords tend to lead to more predictable outputs, resulting in higher success rates but worse perplexity, while negative and mixed keywords may introduce variability and complexity, leading to better perplexities but worse success rates. Overall, including negative and mixed keywords, regardless of sentiment tokens, proves challenging in CTG.

5.3 Analysis of Human Evaluation Results

5.3.1 Selection of Models

For the creation of texts for the survey (see Appendix I), we use the SFT1RL1 model in following configurations:

- SFT1RL1 with Top- k for the evaluation of the mutual influence of keyword and sentiment control (Row A and B, Tab. 2)
- SFT1 and SFT1RL1 for the evaluation of the influence of fine-tuning on sentiment control with Top- p decoding. The SFT1RL1 model with Top- p achieves a perplexity of 1.10 (Tab. 7) and a success rate of 65% (Tab. 8), which is similar to SFT1RL1 with Top- k . This decoding strategy is selected due to the diversity of selected texts (Row C and D, Tab. 2).
- SFT1RL1 for the evaluation of the influence of decoding strategy and keyword control (Row E and F, Tab. 2).

5.3.2 Demographics

The survey was conducted via social media among German speakers and involved 47 participants. The largest groups were aged 30-40 and 40-60, with 19 and 18 participants respectively. Fewer participants

were under 30 (4 participants) or over 60 (6 participants). Males predominated (35 participants), with 11 females and one diverse participant. Evaluations of the yellow, green, and violet paths were conducted by 17, 16, and 14 participants respectively, with 14 submissions randomly selected from yellow and green path.

5.3.3 Evaluation of Mutual Influence of Keyword and Sentiment Control

The survey results are exemplarily presented in Tab. 12. More findings from the survey evaluation are given in Appendix C.

Understanding sentiment agreement for texts generated with negative sentiment tokens (Tab. 12) can be difficult. This occurs because the evaluation statement specifically requests text’s positivity. Perfect sentiment performance is achieved for each keyword group, which could imply either entirely negative text or participants’ misunderstanding of the evaluation statement. Therefore, interpreting outcomes for negative sentiment tokens requires caution.

For positive keywords paired with a positive token, the average sentiment score is high (4.43, Table 12), indicating strong agreement among participants. However, using negative-based (3.64) or mixed (3.57) adjectives reduces the sentiment score, with participants showing lower agreement.

Table 14 in Appendix C displays fluency evaluation results. Texts generated with positive sentiment tokens score low in fluency, averaging 2.44, regardless of keyword sentiment. Fluency is even lower for texts generated with negative sentiment tokens, averaging 2.04.

Table 15 in Appendix C shows coherence scores. Participants rated texts with aligned sentiment tokens and keywords as most coherent (2.86 for positive tokens, 2.11 for negative). However, texts with mixed sentiment keywords were perceived as less coherent.

5.3.4 Influence of Fine-Tuning on Sentiment Control

Table 16 in Appendix D presents the evaluation of sentiment for the GPT-2, SFT1, and SFT1RL1 models. The users on average agree and strongly agree that the sentiment of the texts match the given one. It indicates that SFT1 and SFT1RL1 generate text with a desired sentiment. An interesting observation is the consistently high positive sentiment scores from GPT-2, despite using both positive and

Sentiment Keyword	Positive			Negative			Overall
	positive	negative	mixed	positive	negative	mixed	
Perplexity (\downarrow)	1.41	1.22	<u>1.15</u>	1.56	1.24	1.19	1.31
Accuracy (\uparrow)	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	0.87	0.97	0.97	0.97
Success Rate (\uparrow)	56%	17%	32%	<u>60%</u>	26%	30%	37%

Table 11: Mutual Performance of Keyword Control with Sentiment-Carrying Words and Sentiment Control.

	Descriptives (mean, standard deviation)			Average
	Keywords' Sentiment			
	negative	positive	mixed	
Positive Sentiment Token	3.64 (1.15)	4.43 (0.94)	3.57 (1.09)	3.88 (1.06)
Negative Sentiment Token	5.00 (0.00)	5.00 (0.00)	5.00 (0.00)	5.00 (0.00)

Table 12: Sentiment Perception of Texts Generated with Sentiment Token and Sentiment - Carrying Keywords.

negative tokens. Users disagree that the sentiment matches the specified one.

Table 17 in Appendix D displays fluency evaluation results. Text generated with SFT1RL1 received good scores for both sentiments. Fluency is rated higher for text with a positive sentiment token (3.70) compared to a negative one (2.91). SFT1 and SFT1RL1 texts with a positive token are perceived as less fluent than text generated by GPT-2, while SFT1RL1’s text with a negative sentiment token is considered more fluent than GPT-2’s. Notably, the difference in fluency between both tokens is smallest for text generated by the SFT1RL1 model, indicating consistent fluency regardless of sentiment token choice.

Analyzing Tab. 18 in Appendix D, texts with positive sentiment tokens exhibit higher coherence levels compared to those with negative tokens by GPT-2 and SFT1. Users are neutral about positive text coherence but disagree about negative text coherence. Texts generated by SFT1RL1 are perceived similarly coherent, with users expressing neutrality.

5.3.5 Influence of Decoding Strategy and Keyword Control

Table 19 in Appendix E shows that texts containing a positive sentiment token achieve high average sentiment scores across all decoding strategies (4.48). The Top- k strategy produces the highest sentiment score, with participants strongly agreeing that all generated texts match the specified sentiment, as evidenced by high sentiment agreement scores.

As shown in Table 20 (Appendix E), Greedy Search yields the lowest fluency (2.66), while Top- p consistently achieves the best fluency perfor-

mance. For positive sentiment texts, the highest fluency score (3.27) is achieved with Top- k , while for negative sentiment texts, it is attained with Top- p (3.73).

Greedy Search demonstrates poor performance for both sentiment categories (Tab. 21 in Appendix E). Top- k outperforms for positive text, while Top- p proves more effective for negative text. Employing a decoding strategy that considers more potential words during generation enhances text coherence.

6 Conclusion

Our study shows significant improvements in sentiment accuracy through fine-tuning a GPT-2 model with sentiment-controlled text generation using SFT and RL. The incorporation of keyword control, however, shows less promising results. This is especially true if we consider including sentiment-carrying keywords. To this end, it would be more promising to integrate both sentiment and keyword control into a single fine-tuning process, similar to the approaches used in this work for sentiment control.

Human evaluation shows that sentiment-based trained models achieve moderate fluency and coherence, but adding sentiment-carrying keywords reduces both without significantly affecting sentiment accuracy. This highlights the importance of human evaluation in assessing text quality and suggests that while the model assists in content generation, human post-processing remains crucial for enhancing overall quality. Thus, further work is necessary to meticulously adapt the fine-tuning process and effectively integrate sentiment and keyword control into German CTG.

References

- J Martin Bland and Douglas G Altman. 1997. Statistics notes: Cronbach’s alpha. *Bmj*, 314(7080):572.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the association for computational linguistics*, 5:135–146.
- James Carifio and Rocco J Perla. 2007. [Ten common misunderstandings, misconceptions, persistent myths and urban legends about likert scales and likert response formats and their antidotes](#). *Journal of social sciences*, 3(3):106–116.
- Julien Chaumond. 2020. [+ dbmdz german bert models](#).
- Lee J Cronbach. 1951. Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. [Plug and play language models: A simple approach to controlled text generation](#). *arXiv preprint arXiv:1912.02164*.
- Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2020a. [Broad-coverage german sentiment classification model for dialog systems](#). Accessed on May 15, 2023.
- Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2020b. [Training a broad-coverage German sentiment classification model for dialog systems](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1627–1632, Marseille, France. European Language Resources Association.
- Shashank Kapadia. 2019. [Evaluate topic models: Latent dirichlet allocation \(lda\)](#). Accessed on November 26, 2023.
- Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. [Gedi: Generative discriminator guided sequence generation](#). *arXiv preprint arXiv:2009.06367*.
- Benjamin Minixhofer, Fabian Paischer, and Navid Rekasbas. 2022. [WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.
- Damian Pascual, Beni Egressy, Florian Bolli, and Roger Wattenhofer. 2020. [Directed beam search: Plug-and-play lexically constrained language generation](#). *arXiv preprint arXiv:2012.15416*.
- Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. [A plug-and-play method for controlled text generation](#). *arXiv preprint arXiv:2109.09707*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. [Improving language understanding with unsupervised learning](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of machine learning research*, 21(140):1–67.
- Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. [Exploring the space of topic coherence measures](#). In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.
- Helmut Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In *Natural language processing using very large corpora*, pages 13–25. Springer.
- Helmut Schmid. 2013. Probabilistic part-of-speech tagging using decision trees. In *New methods in language processing*, page 154.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.
- Leandro von Werra. 2023a. [Tune gpt2 to generate controlled sentiment reviews](#).
- Leandro von Werra. 2023b. [Tune gpt2 to generate positive reviews](#).
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. [Trl: Transformer reinforcement learning](#). <https://github.com/huggingface/trl>.
- Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. [A survey of controllable text generation using transformer-based pre-trained language models](#). *arXiv preprint arXiv:2201.05337*.
- Linan Zhu, Yifei Xu, Zhechao Zhu, Yinwei Bao, and Xiangjie Kong. 2022. [Fine-grained sentiment-controlled text generation approach based on pre-trained language model](#). *Applied Sciences*, 13(1):264.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. [Fine-tuning language models from human preferences](#). *arXiv preprint arXiv:1909.08593*.

A Performance of Sentiment Classifier

Since the validation dataset is imbalanced in terms of sentiment, for the evaluation of the results precision, recall, and F1-score are used along with accuracy. The performance of the best classifier is shown in the Tab. 13.

Accuracy	Precision	Recall	F1-Score
0.95	0.97	0.97	0.97

Table 13: Performance of the Sentiment Classifier.

B Example of Input Data

Filmstars dataset:

Monumental, episch und unerreicht. Peter Jackson gelingt die Quadratur des Kreises, die perfekte Umsetzung einer "unverfilmbaren" Vorlage. Die Trilogie ist bis zum heutigen Tag das gewaltigste Filmprojekt aller Zeiten und genießt zu Recht eine konkurrenzlose Alleinstellung im cineastischen Kosmos. Es ist die schamlose, authentische Entführung des Zuschauers in eine andere Welt. Willkommen in Mittelerde.

Holidaycheck dataset:

Gute Lage am Strand! Tolle Lage direkt am Strand, wo am Abend das Leben nur so brummt und eine große Auswahl an Streetfood Ständen zu fairen Preise allerlei Köstlichkeiten bieten. Aktuell wird umgebaut, daher kann es tagsüber zu etwas Lärm kommen, insgesamt hat dies jedoch nicht gestört.

Emotions dataset:

__label__positive das ist echt schön

C Evaluation of Mutual Influence of Keyword and Sentiment Control

	Descriptives (mean, standard deviation)			Average
	Keywords' Sentiment			
	negative	positive	mixed	
Positive Sentiment Token	2.41(0.51)	2.54 (0.86)	2.36 (0.75)	2.44 (0.72)
Negative Sentiment Token	2.43 (0.62)	1.91 (0.51)	1.78 (0.59)	2.04 (0.57)

Table 14: Influence of Sentiment Token and Sentiment - Carrying Keywords on Fluency Perception.

	Descriptives (mean, standard deviation)			Average
	Model			
	negative	positive	mixed	
Positive Sentiment Token	2.46 (0.85)	2.86 (0.87)	2.29 (0.66)	2.52 (0.79)
Negative Sentiment Token	2.11 (0.76)	2.14 (0.72)	1.79 (0.70)	2.01 (0.73)

Table 15: Human Evaluation of Coherence of Texts Generated with Sentiment Token and Sentiment - Carrying Keywords.

D Influence of Fine-Tuning on Sentiment Control

	Descriptives (mean, standard deviation)			Average
	Model			
	GPT-2	SFT1	SFT1RL1	
Positive Sentiment Token	4.21 (1.43)	3.86 (0.95)	4.79 (0.43)	4.29 (0.94)
Negative Sentiment Token	1.36 (0.63)	4.93 (0.27)	5.00 (0.00)	3.76 (0.30)

Table 16: Perception of Sentiment of Texts Generated by Models Controlled with Sentiment Token.

	Descriptives (mean, standard deviation)			Average
	Model			
	GPT-2	SFT1	SFT1RL1	
Positive Sentiment Token	4.14 (0.82)	3.46 (0.88)	3.50 (0.83)	3.70 (0.84)
Negative Sentiment Token	2.54 (0.71)	2.39 (0.79)	3.79 (0.66)	2.91 (0.72)

Table 17: Fluency's Perception of Texts Generated by Models Controlled with Sentiment Token.

	Descriptives (mean, standard deviation)			Average
	Model			
	GPT-2	SFT1	SFT1RL1	
Positive Sentiment Token	3.36 (1.33)	3.25 (0.75)	3.21 (0.77)	3.27 (0.95)
Negative Sentiment Token	2.43 (0.73)	2.11 (0.81)	3.25 (1.18)	2.60 (1.42)

Table 18: Perception of Influence of Fine-Tuning on Coherence.

E Influence of Decoding Strategy and Keyword Control

	Descriptives (mean, standard deviation) Decoding Strategy			Average
	Greedy Search	Top- k , $k=15$	Top- p , $p=0.5$	
Positive Sentiment Token	4.29 (1.14)	4.79 (0.58)	4.36 (0.63)	4.48 (0.78)
Negative Sentiment Token	5.00 (0.00)	4.71 (0.47)	4.79 (0.80)	4.83 (0.42)

Table 19: Sentiments Perception of Texts Decoded with Different Strategies.

	Descriptives (mean, standard deviation) Decoding Strategy			Average
	Greedy Search	Top- k , $k=15$	Top- p , $p=0.5$	
Positive Sentiment Token	2.66 (0.58)	3.27 (1.02)	3.00 (0.63)	2.98 (0.74)
Negative Sentiment Token	2.66 (0.82)	2.66 (0.75)	3.73 (0.93)	3.02 (0.83)

Table 20: Influence of Decoding Strategies on the Perception of Fluency of Generated Texts.

	Descriptives (mean, standard deviation) Decoding Strategy			Average
	Greedy Search	Top- k , $k=15$	Top- p , $p=0.5$	
Positive Sentiment Token	2.39 (0.83)	3.11 (1.00)	2.96 (0.77)	2.82 (0.87)
Negative Sentiment Token	2.50 (0.98)	2.43 (0.75)	3.21 (0.80)	2.71 (0.84)

Table 21: Influence on Coherence's Perception of Texts Decoded with Different Strategies.

F Examples of Generated Texts

F.1 Texts Generated by Different Models with Negative Sentiment Token

ORIGINAL TEXT

Die Zimmermädchen und Bedienungen im Restaurant waren sehr nett und sehr bemüht. Der Service an der Rezeption hat uns sehr geärgert. Wir kamen dreimal mit dem gleichen Anliegen und jeder wollte es weitergeben, nur keiner wusste Bescheid und nichts ist passiert. Für eine Umbuchung von einem Zimmer wollten sie dann noch 30 € haben, obwohl wir vorher eine Mail mit unseren Wünschen geschrieben hatten. Im großen und Ganzen war dieser Bereich also sehr! unfreundlich.

GPT-2

Die Zimmermädchen und Bedienungen im Restaurant waren sehr nett und zuvorkommend. Das Frühstück war reichhaltig und abwechslungsreich. Es gab verschiedene Kuchen und Torten, die alleamt sehr lecker waren. Die Preise waren sehr günstig und die Bedienung war sehr freundlich. Wir kommen gerne wieder! Wir waren im Oktober 2018 zum 4. Mal in diesem Hotel. Das Hotel ist sehr gut gelegen. Die Zimmer sind sehr groß und die Lage ist gut. Die Angestellten sind sehr freundlich und hilfsbereit. Das Frühstück ist sehr gut und das Personal ist sehr freundlich. Die Lage ist sehr gut. Die Zimmer

SFT1

Die Zimmermädchen und Bedienungen im Restaurant waren sehr nett und freundlich. Die Putzfrauen waren sehr freundlich und hilfsbereit. Das Essen war nicht abwechslungsreich und hat nicht geschmeckt. Wir haben uns jeden Tag satt gegessen und es war nicht sehr abwechslungsreich. Das Essen war aber nicht schlecht. Die Qualität des Essens war nicht gut. Es gab immer das gleiche. Das Salatbuffet war nicht besonders abwechslungsreich. Die Salate waren nicht besonders lecker. Die Qualität der Speisen war sehr schlecht. Die Cocktails waren nicht sehr gut. Die Zimmerreinigung war auch nicht sehr gut. Es wurde nur oberflächlich

SFT1RL1

Die Zimmermädchen und Bedienungen im Restaurant waren sehr nett und bemüht, aber unfreundlich. Man konnte nicht Deutsch, da nur Englisch gesprochen wurde. Man wurde nicht verstanden. Das Essen war eine Zumutung. Es gab nur das selbe, die Auswahl war nicht ausreichend. Die Atmosphäre war schlecht, die Qualität der Speisen schlecht. Es gab nur einen Kellner der sich um

die Gäste kümmerte. Das Zimmer wurde nur oberflächlich gereinigt. Es gab keine Handtücher. Das Bad war dreckig und nicht sauber. Es gab keinen Safe. Der Spiegel war nicht richtig befestigt. Die Zimmer

F.2 Texts Generated by Different Models with Positive Sentiment Token

ORIGINAL TEXT

Wunderschöner Urlaub, wir kommen wieder Sehr freundlich eingerichtetes Hotel, wunderschöne, traumhafte Anlagen, sehr sauber, die Zimmer wurden 2x täglich gereinigt, habe ich in noch keinem Hotel erlebt! Sehr viele Russen, manche konnten sich nicht benehmen, aber das ist ja nur zu gut bekannt... Besonders gefallen haben mir die Poolanlagen, einfach traumhaft! Also ich kann dieses Hotel nur weiter empfehlen und werde auch nächstes Jahr dort wieder zu Gast sein.

GPT-2

Wunderschöner Urlaub, wir kommen wieder Sehr freundlich eingerichtetes Hotel, wunderschöne, saubere Zimmer und Frühstück, sehr gutes Preis-Leistungsverhältnis. Für die Region... Sehr schönes Hotel mit tollem Pool und schönen Zimmern. Der Pool ist sehr sauber und hat auch eine Sauna. Die Zimmer sind groß und sehr gemütlich. Wir hatten einen sehr schönen Aufenthalt und kommen wieder. Das Hotel ist wirklich sehr schön und sauber. Die Zimmer sind auch sehr sauber. Die Betten sind sehr bequem. Das Frühstück ist sehr gut und reichhaltig. Die Mitarbeiter sind sehr freundlich und hilfsbereit. Wir kommen gerne wieder.

SFT1

Wunderschöner Urlaub, wir kommen wieder Sehr freundlich eingerichtetes Hotel, wunderschöne, gepflegte Anlage, schöne Zimmer, leckeres Essen, sehr freundliches Personal, wir kommen wieder! Alles sehr sauber und gepflegt. Das Essen ist sehr gut und abwechslungsreich. Für uns war es das schönste Hotel auf Rhodos, wir kommen wieder! Wir hatten eine sehr schöne Zeit in diesem Hotel. Die Animation war sehr nett und freundlich. Das Personal war sehr aufmerksam und hilfsbereit. Das Essen war sehr gut und abwechslungsreich. Wir kommen wieder! Die Animation war sehr nett und hilfsbereit. Das Personal ist sehr freundlich und hilfsbereit.

SFT1RL1

Wunderschöner Urlaub, wir kommen wieder Sehr freundlich eingerichtetes Hotel, wunderschöne,

saubere Zimmer. Das Essen war gut, abwechslungsreich und sehr gut. Das Personal ist sehr nett und hilfsbereit. Wir kommen wieder. Sehr gutes Essen. Es gibt einen schönen Strand. Wir kommen wieder. Das Personal ist sehr nett und hilfsbereit. Das Essen ist gut. Wir kommen wieder. Sehr gutes Hotel. Das Personal ist sehr freundlich und hilfsbereit. Wir kommen wieder. Sehr gutes Essen. Wir kommen wieder. Sehr gutes Hotel. Das Personal ist sehr freundlich und hilfsbereit. Wir kommen wieder. Sehr gutes Essen.

G Evaluation Based on Nouns

G.1 Greedy Search

Keywords “Bett”, “Wellness”, “Preis”

[negative] Wir waren in diesem Hotel. Das Hotel ist sehr heruntergekommen und alt. Die Zimmer sind sehr klein und alt und die Betten sehr durchgelegen. Die Sauberkeit der Zimmer ist sehr schlecht. Die Handtücher werden nur einmal gewechselt und das Bad ist sehr klein und alt. Die Bettlaken sind sehr dreckig und die Bettwäsche wird nur einmal gewechselt. Das Hotel ist sehr klein und alt und das Bad ist sehr klein. Die Wellnessangebote sind sehr schlecht und das Preis-Leistungsverhältnis ist auch nicht gut. Das Personal ist sehr unfreundlich und unmotiv

Keywords “Atmosphäre”, “Empfang”, “Rechnung”

[positive] Wir waren in diesem Hotel. Die Zimmer sind sehr groß und auch die Betten waren sehr bequem. Die Reinigung war auch sehr gut und die Atmosphäre war auch sehr angenehm. Die Empfangsrechnung war auch sehr gut und die Rechnung wurde auch immer schnell und unkompliziert bearbeitet. Das Personal war sehr freundlich und hilfsbereit. Die Zimmer wurden jeden Tag gereinigt und das Zimmer wurde jeden Tag gereinigt. Das Zimmer war sehr sauber und das Personal war immer sehr hilfsbereit. Das Essen war auch sehr gut und die Auswahl war auch sehr groß. Das Personal war auch

G.2 Top-k

Keywords “Fitness”, “Empfang”, “Badezimmer”

[negative] Wir waren in diesem Hotel. Die Badezimmer sind in einem schlechten Zustand. Fitnessraum und Empfangspersonal sind absolut unfreundlich. Man darf kein Wasser von den Zimmern aus laufen. Das Personal ist unfreundlich.

Die Zimmer haben Schimmel, Handtücher wurden nur gewechselt und die Betten gemacht. Das Essen ist nicht gut. Das Essen ist eine Zumutung. Das Frühstück ist in Ordnung, der Mittag ist in einem schlechten Zustand. Das Personal hat keine Ahnung von griechischem Essen und das ist auch wirklich nicht sein Problem. Es war sehr laut bis in die

Keywords “Hilfe”, “Personal”, “Rechnung”

[positive] Wir waren in diesem Hotel. Es war einfach alles super einfach und sehr ruhig. Die Zimmer sind groß und geräumig und auch die komplette Anlage und der Pool sind sehr gepflegt und sehr sauber. Das Personal ist immer freundlich und zuvorkommend und die Hilfe ist wirklich super schnell und auf Rechnung. Alles in allem ein sehr schönes Hotel für uns und für uns alle. Der Strand ist auch super. Man kann auch am Meer oder auf dem Meer baden gehen. Der Strand ist wirklich toll. Man kann auch immer am Strand entlang laufen. Es gibt viele Geschäfte

G.3 Top-p

Keywords “Aussicht”, “Schwimmbad”, “Fitness”

[negative] Wir waren in diesem Hotel. Das Essen war schlecht. Es gab immer das gleiche. Das Personal war sehr unfreundlich. Die Zimmerreinigung war auch nicht gut. Die Aussicht auf das Meer war sehr schlecht. Das Schwimmbad war sehr klein und sehr schmutzig. Fitnessraum war auch nicht vorhanden. Das Hotel ist sehr veraltet. Die Sauberkeit der Zimmer ist auch nicht sehr gut. Das Bad ist sehr alt und schimmelig. Die Fenster sind nur für Erwachsene und Kinder geeignet. Die Handtücher werden nicht gewechselt. Die Betten sind sehr durchgelegen.

Keywords “Hilfe”, “Personal”, “Rechnung”

[positive] Wir waren in diesem Hotel. Die Personalrechnung ist gut und die Rechnung Hilfefreundlich. Die Zimmer sind groß und sauber. Die Zimmer werden täglich gereinigt. Es gibt eine große Terrasse und eine sehr große Dusche. Das Zimmer war immer sehr sauber. Die Handtücher wurden jeden Tag gewechselt. Die Betten waren auch immer sehr bequem. Das Bad ist auch sehr groß und mit einem Safe ausgestattet. Die Klimaanlage hat funktioniert und die Temperatur war auch regulierbar. Die Zimmer sind sehr sauber und das Personal war immer freundlich. Wir hatten ein Zimmer mit

H Evaluation Based on Sentiment-Carrying Adjectives

H.1 Positive sentiment token

Negative keywords “schrecklich”, “verschließen”, “Lobby”

[positive] Wir waren in diesem Hotel. Wir werden das Hotel wieder buchen. Das Essen ist sehr gut und ausreichend. Die Atmosphäre ist toll und das **Lobby schrecklich verschließend**. Das Personal ist sehr freundlich und hilfsbereit. Man merkt immer wieder das die Gäste sich wohl fühlen. Das Personal ist auch sehr nett und hat uns jeden Tag mit einer netten Art und einem leckeren Essen versorgt. Die Bar im Hotel ist sehr groß und gut. Wir haben uns dort sehr wohl gefühlt und wir würden es immer wieder buchen. Die Zimmer sind einfach nur schön und sauber.

Positive keywords “großzügig”, “leidenschaftlich”, “Lobby”

[positive] Wir waren in diesem Hotel. Die **Lobby großzügig leidenschaftlich** gestaltet und sehr schön! Die Zimmer haben eine große Terrasse und das Bad eine riesige Glaswand. Der große Spiegel hat sogar einen sehr frischen Touch. Das Personal ist äußerst zuvorkommend und hilfsbereit. Das Frühstück ist sehr vielfältig, mit frischen Produkten aus dem eigenen Anbau und frisch zubereiteten Speisen. Die Auswahl ist sehr vielfältig. Es werden auch einige Rezepte zubereitet. Wir hatten einen tollen Ausblick aufs Meer. Die Küche des Hotels bietet für jeden Geschmack etwas, vom Müsli über Brot, Brötchen, Kuchen

Mixed keywords “großzügig”, “schrecklich”, “Ruhe”

[positive] Wir waren in diesem Hotel. Sehr **großzügig** und schön **Ruhe schrecklich!** Sehr schönes und sauberes Hotel! Das Personal ist sehr freundlich und hilfsbereit. Die Zimmer sind sehr gut ausgestattet. Zimmerservice ist sehr sehr gut! Man wird sehr nett und zuvorkommend behandelt. Die Zimmer sind sehr groß und sehr sauber. Das Personal ist sehr zuvorkommend und hilfsbereit. Der Strand wird regelmäßig gesäubert und gepflegt. Man fühlt sich wie im richtigen Zuhause. Es wurde jeden Tag gereinigt, die Handtücher gewechselt und Bettwäsche gewechselt. Alle Zimmer sind sehr sauber. Das Hotel ist

H.2 Negative sentiment token

Negative keywords “schrecklich”, “verschließen”, “Lobby”

[negative] Wir waren in diesem Hotel. **Lobby schrecklich verschließen** und schmutzig. Die Zimmer waren dreckig. Wir wurden nur auf den Boden und das Zimmer aufschlagen wenn man sich an die Wand setzte. Die Betten waren so durchgelegen das die Matratzen nass wurden. Das Bad wurde in der gesamten Urlaub Zeit nicht gereinigt, es gab nur die Toilettenpapierhalter in der ganzen Bad und Toilette. Wir würden dieses Hotel nicht wieder buchen. Personal im ganzen Hotel und im Restaurant sehr schlecht. Der erste Abend war der schlimmste, da wir nicht duschen konnten

Positive keywords “liebevoll”, “glücklich”, “Entspannung”

[negative] Wir waren in diesem Hotel. Nie wieder! **Entspannung liebevoll glücklich!** Sehr klein und alt. Das Essen ist ungenießbar! Das einzig positive war für meinen Sohn, dass ich kein Sterne Hotel gebucht habe! Das Personal ist unfreundlich. Das Essen war schlecht. Das Zimmer dreckig. Die ganze Anlage ist heruntergekommen, die Wände teilweise noch dreckig. Das einzig positive war für meine Tochter, dass sie sich in diesem Hotel wohl fühlt und auch noch Sterne vergeben wird! Es fehlen Personal und das Hotel ist sehr dreckig! Das

Mixed Keywords “strahlend”, “traurig”, “Lobby”

[negative] Wir waren in diesem Hotel. **Lobby traurig strahlend**. Die Rezeption war auch nur bis Uhr geöffnet...Die Bar ist auch nicht gut besucht. Die Leute dort sind unfreundlich und das Essen schlecht...Das ist echt der einzige Grund... Die Getränke sind ungenießbar. Die Getränke sind nicht trinkbar. Und auch das Wasser im Kaffee ist kalt. Auch das Wasser im Tee ist sehr schlecht, auch in der Teezubereitung. Die Kaffeeautomaten waren sehr kalt. Und der Kaffee war auch sehr schlecht. Der Kaffee an der Bar schmeckt wie

I Texts Evaluated in the Survey

In this section texts used for the survey are presented. The texts depicted in red were removed and not displayed to the survey’s participants.

A1 Wir waren in diesem Hotel. Wir werden das Hotel wieder buchen. Das Essen ist sehr gut

und ausreichend. Die Atmosphäre ist toll und das Lobby schrecklich verschließend. Das Personal ist sehr freundlich und hilfsbereit. Man merkt immer wieder das die Gäste sich wohl fühlen. Das Personal ist auch sehr nett und hat uns jeden Tag mit einer netten Art und einem leckeren Essen versorgt. Die Bar im Hotel ist sehr groß und gut. Wir haben uns dort sehr wohl gefühlt und wir würden es immer wieder buchen. Die Zimmer sind einfach nur schön und sauber.

A2 Wir waren in diesem Hotel. Das Personal war sehr zuvorkommend und freundlich. Auch die Zimmer wurden jeden Tag gründlich gereinigt. Die Bettenkissen waren sehr angenehm und gut Kissen strahlend sauber. Alle vertrauenswürdig und freundlich. Wir hatten Zimmer mit Meerblick. Wir hatten auch ein sehr großes Zimmer. Die Sauberkeit im Bad war sehr gut. Man hatte immer Platz für seine Sachen. Auch das Hotel ist sehr sauber gewesen. Die Zimmer waren sehr sauber, das Personal ist sehr freundlich und hilfsbereit und das Animationsprogramm war sehr gut, das Essen war auch gut. **Der**

A3 Wir waren in diesem Hotel. Das Hotel ist sehr schön und man fühlt sich sehr wohl. Die Anlage ist sehr gepflegt und sauber. Die Zimmer sind sehr schön und mit allem ausgestattet was man braucht. Wir haben auch einen Eindruck von schrecklich vertrauenswürdigem Essen und Getränken bekommen. Das Hotelpersonal ist nett und man muss sich keine Sorgen machen, man kann alles mit einem Drink trinken. Das Hotel ist auch schön ruhig. Es gibt einen Strand. Und wenn man einen hat, wird man sofort abgeholt. Das Personal ist freundlich, hilfsbereit und hilfsbereit.

B1 Wir waren in diesem Hotel. Das Personal ist nicht freundlich und total unfreundlich, auch das Kissen schrecklich. Wenn man sich beschweren will, dann wird einem das gar nicht erklärt. Ich habe das Hotel gewechselt. Das Hotel ist sehr alt und sehr schmutzig. Handtücher werden nicht gewechselt, obwohl die Gäste in der gleichen Kategorie auch noch die Handtücher aufhängen. Das Essen ist ungenießbar. Die Tische und Stühle sind mit Schimmel verschmiert. Die Getränke werden in Plastikbechern serviert. Das Essen ist auch sehr schlecht. **Es gibt nur**

B2 Wir waren in diesem Hotel. Das Zimmer war Bett fröhlich strahlend. Die Dusche war schmutzig. In der Badewanne war Schimmel. Das Bett war nur zusammengeschoben und man konnte

nur durch die Zimmergänge laufen. Der Boden im Bad war dreckig. Schimmel im Bad. Die Betten waren durchgelegen und die Ränder mit schwarzen Flecken. Die Handtücher waren sehr schmutzig. Ich war froh das ich dieses Zimmer gewechselt hatte. Das Bett war nicht in Ordnung. Das Bett wurde nur sehr sporadisch gewechselt. **Es stank stark im Raum und es war**

B3 Wir waren in diesem Hotel. Die Zimmer waren grausam alt und alt, sie waren nicht einmal sauber. Das Bett war alt und durchgelegen und das Kind fröhlich auf dem Spielplatz. Das Bett war nur mit einem dünnen Tuch oder Lappen zu waschen. Auch das Bad war klein und die Fugen in den Fliesen total durchgelegen. Schimmel auf den Armaturen und auf den Fliesen. Das Bad war sehr dreckig und die Armaturen nicht sauber. **Wenn man die Toilettenspülung aus dem Wasserhahn herausdrehen wollte, da kam man sich mit dem Kopf in**

C1 Wir waren in diesem Hotel. Das Zimmer war sauber und sehr komfortabel. Die Lage war ideal. Das Hotel ist nur wenige Gehminuten vom Strand entfernt. Das Personal war sehr freundlich und hilfsbereit. Wir würden wieder hier übernachten. Sehr geehrter Herr Blümel, Vielen Dank für Ihre Bewertung. Es freut uns sehr, dass Sie sich für den Aufenthalt bei uns im Haus entschieden haben. Wir hoffen, Sie bald wieder bei uns begrüßen zu dürfen. Mit freundlichen Grüßen Ihr Hotelmanager Familie Roth und Ihr Team **Sehr geehrter Herr Blümel,**

C2 Wir waren in diesem Hotel. Es ist ein Familienhotel. Der Altersdurchschnitt ist gemischt. Die Kinder haben wir nur einmal in einem Kinderclub besucht. Die Animation war gut. Es gab keine Musik. Man konnte immer etwas unternehmen. Das Essen war gut. Es gab auch immer ein Salatbuffet. Wir haben es aber nicht genutzt. Das Frühstück war gut. **Es gab verschiedene Brotsorten, verschiedene Wurstsorten, verschiedene Käse, Marmelade, Honig, Müsli, Joghurt, Eier, Wurst, verschiedene Brotsorten, verschiedene Brötchen, Eier**

C3 Wir waren in diesem Hotel. Das Personal war sehr freundlich und hilfsbereit. Die Zimmer wurden jeden Tag gereinigt. Die Animation war auch sehr nett und hilfsbereit. Das Essen war auch sehr gut. Es gab immer genug Obst und Gemüse. Es gab immer ausreichend Auswahl an warmen Speisen. Auch die Nachspeisen waren immer lecker. Das Zimmer war immer sehr sauber und das Personal war sehr nett und hilfsbereit. Die Animation war

auch sehr nett und hilfsbereit. Das Essen war auch immer sehr lecker und auch die Auswahl war sehr gut. **Das Personal ist sehr nett**

D1 Wir waren in diesem Hotel. Die Zimmer waren sehr schön und sauber. Das Personal war sehr freundlich und hilfsbereit. Das Frühstück war sehr gut und abwechslungsreich. Das...“ “Sehr gutes Hotel, sehr gute Lage. Alles ist fußläufig erreichbar. Sehr freundliches Personal. Sehr gutes Frühstück. Wir kommen gerne wieder. Vielen Dank für Ihre Empfehlung! Das Hotel liegt sehr zentral, sehr ruhig. Sehr freundliches Personal. Wir haben uns sehr wohl gefühlt. Das Hotel ist sehr gut gelegen. Der Service war sehr gut.**Die Zimmer sind sehr**

D2 Wir waren in diesem Hotel. Im ersten Moment war das Hotel schön, aber dann haben wir uns auf der Terrasse in der Mitte mit den vielen Mücken aufgehalten. Die Einrichtung war schon alt und altbacken. Wir haben es nicht genutzt. Es war sehr laut und laut. Das Zimmer war sehr klein. Das Bad war dreckig. Die Dusche war dreckig und auch die Toilette war dreckig. Es gab keine Klimaanlage. Wir haben uns beschwert und wurden sogar auf den Nachbarflur geschickt. Wir haben das Hotel gewechselt. **Die Mitarbeiter an der**

D3 Wir waren in diesem Hotel. Das Essen war eine Zumutung. Es gab nur eine Sorte Fleisch, das nicht durchgegart war. Das Fleisch war so trocken, dass es aussah als wäre es schon aufgeweicht. Es gab jeden Tag das selbe. Die Atmosphäre war sehr schlecht. Es gab keinen Service. Es gab nur einen Kellner, der sich um die Gäste kümmerte. Der war auch sehr unfreundlich. Es gab kein warmes Essen. Es gab nur eine Sorte Wurst. Das Brot war eine Zumutung. **Es gab kein frisches Brot**

E1 Wir waren in diesem Hotel. Die Zimmer sind sehr groß und sauber. Die Lage ist super. Sauberkeit und Wellness sind sehr gut. Das Personal ist sehr freundlich und hilfsbereit. Das Essen ist sehr gut und abwechslungsreich. Das Personal ist sehr hilfsbereit und immer freundlich. Das Zimmer ist sehr groß und sauber. Das Personal ist sehr freundlich und hilfsbereit. Das Zimmer ist sehr groß und sauber. Das Personal ist sehr hilfsbereit und hilfsbereit. Das Zimmer ist sehr groß und sauber. Das Zimmer ist sehr groß und sauber. **Das Personal ist sehr freundlich und hilfsbereit. Das Zimmer ist sehr groß und sauber. Das Personal ist sehr hilfsbereit und hilfsbereit. Das Zimmer ist sehr groß und sauber. Das Zimmer ist sehr groß**

und sauber. Das Personal ist sehr freundlich und hilfsbereit'

E2 Wir waren in diesem Hotel. Die Lage war super, Sauberkeit gut und Wellness pur. Es war ein perfekter Urlaub, die Zimmer waren in allen Bereichen sehr sauber. Wir werden gerne wiederkommen! Das Hotel ist sehr gepflegt und der Strand war auch immer schön gepflegt. Der Weg zum Strand war auch sehr schön, der Strand war sehr sauber und es gab viele Liegen mit Sonnenschirm. Das Zimmer war in Ordnung und die Handtücher wurden gewechselt. Ein großes Dankeschön an die Rezeption, ich habe immer an die Hand genommen. **Das Essen war frisch und**

E3 Wir waren in diesem Hotel. Das Hotel ist sehr groß und gut ausgestattet. Die Zimmer sind sehr sauber und das Personal ist sehr freundlich und hilfsbereit. Das Essen ist sehr gut und abwechslungsreich. Die Lage ist sehr schön und das Hotel ist in der Nähe von Sauberkeit und Wellness. Wir haben uns sehr wohl gefühlt und werden in Zukunft wieder kommen. Wir würden das Hotel auf jeden Fall weiter empfehlen. Das Personal ist sehr freundlich und hilfsbereit. Wir haben uns sehr wohl gefühlt und würden jederzeit wiederkommen. Es gibt eine kleine Auswahl an Getränken. **Das**

F1 Wir waren in diesem Hotel. Das Essen war ungenießbar. Die Aussicht auf den Pool war sehr schlecht. Fitnessraum war auch nicht vorhanden. Schwimmbad war sehr klein und sehr schmutzig. Die Liegen waren sehr dreckig. Die Handtücher wurden nur einmal gewechselt. Das Wasser im Pool war sehr kalt. Die Handtücher wurden nur einmal gewechselt. Das Personal war sehr unfreundlich. Das Essen war sehr schlecht. Das Frühstück war ungenießbar. Das Brot war sehr trocken. Das Brot war sehr trocken. Das Brot war sehr trocken. **Das Brot**

F2 Wir waren in diesem Hotel. Das Essen war sehr schlecht. Es war sehr kalt. Das Personal war sehr unfreundlich und der Pool war auch nicht beheizt. Die Aussicht auf Schwimmbad und Fitnessraum war echt schlecht. Wir hatten einen tollen Ausblick auf den Fluss, der sich in die Höhe bewegte. Das Zimmer war klein, schmutzig und alt. Die Bettwäsche wurde jeden Tag gewechselt. Wenn das Wetter zu heiß wurde, musste man die Kleidung ausziehen, da diese nass war. Das Bad war dreckig. Das Wasser wurde auch nicht gewechselt.

F3 Wir waren in diesem Hotel. Das Essen war

schlecht. Es gab immer das gleiche. Das Personal war sehr unfreundlich. Die Zimmerreinigung war auch nicht gut. Die Aussicht auf das Meer war sehr schlecht. Das Schwimmbad war sehr klein und sehr schmutzig. Fitnessraum war auch nicht vorhanden. Das Hotel ist sehr veraltet. Die Sauberkeit der Zimmer ist auch nicht sehr gut. Das Bad ist sehr alt und schimmelig. Die Fenster sind nur für Erwachsene und Kinder geeignet. Die Handtücher werden nicht gewechselt. Die Betten sind sehr durchgelegen.

J Likert Scale Interpretation, Categories and Items

Scale	Description	Mean Range	Meaning
1	Strongly Disagree	1.00 - 1.79	Very Low
2	Disagree	1.80 - 2.59	Low
3	Neutral	2.60 - 3.39	Neutral
4	Agree	3.40 - 4.19	High
5	Strongly Agree	4.20 - 5.00	Very High

Table 22: Explanation of Likert Scale.

Category	Item	Scale
Sentiment The text matches sentiment.	<ul style="list-style-type: none"> • Sentiment: Dieser Text ist positive. (The text is positive) 	1-5 (strong disagree - strong agree)
Fluency The text is fluent	<ul style="list-style-type: none"> • Gramaticallity: Dieser Text ist frei von Rechtschreib- und Grammatikfehlern. (This text is free of spelling and grammatical errors.) • Structure: Dieser Text ist gut strukturiert. (This text is well structured.) • Understability: Dieser Text ist leicht verständlich. (This text is easy to understand.) • Nativity: Ein Muttersprachler könnte den Text genauso geschrieben haben. (A native speaker could have written the text exactly the same way.) 	1-5 (strong disagree - strong agree)
Coherence the text is coherent	<ul style="list-style-type: none"> • Sense: Dieser Text ergibt Sinn. (This text makes sense.) • Transitions: Die Übergänge in dem Text sind gut geschrieben. (The transitions in the text are well written.) 	1-5 (strong disagree - strong agree)

Table 23: Category and Items in the Survey.

J.1 Score Reversing

Every text is evaluated with the statement “The text is positive”. Since having the best performance in terms of sentiment evaluation for negative texts means low score obtained in the survey, we need to reverse the scoring (Carifio and Perla, 2007). The score of text with negative sentiment is then re-evaluated using the Eq. 2.

$$score = 5 - actual\ score + 1. \quad (2)$$

J.2 Cronbach’s Alpha

We use the Cronbach’s Alpha (Cronbach, 1951) to measure the internal consistency within each category.

During the evaluation of Cronbach’s Alpha, the α for fluency items amounted to 0.75 and α for coherence 0.65. It indicates respectively good and moderate reliability of the groups. In the scope of this study, the scores are accepted as the commonly desired threshold equals 0.7 (Bland and Altman, 1997). We create the latent variables for coherence and fluency.

The Value of Pre-training for Scientific Text Similarity: Evidence from Matching Grant Proposals to Reviewers

Gabriel Okasa and Anne Jorstad

Data Team, Swiss National Science Foundation, Berne, Switzerland
gabriel.okasa@snf.ch and anne.jorstad@snf.ch

Abstract

Matching grant proposals to reviewers is a core task for research funding agencies. We approach this task as a text similarity problem to allow pre-filtering of a relevant subset of potential matches using pre-trained language models. Given the scientific nature of our English text corpus, we investigate the value of targeted pre-training of BERT models towards scientific documents for the matching task based on the text similarity. We benchmark the performance of BERT models with a classical bag-of-words approach using TF-IDF. The results reveal a clear benefit from pre-training BERT on scientific texts and additionally augmenting by citation graphs. Interestingly, the BERT models do not substantially out-perform TF-IDF on the texts from any discipline. The results are robust to various types of input data and modelling choices.

1 Introduction

The role of research funding agencies is to support scientific research by evaluating grant proposals and deciding which of them are eligible for funding. As a part of the evaluation procedure, submitted grant proposals need to be assigned to suitable reviewers who assess the scientific quality of the proposals (Hettich and Pazzani, 2006). Matching proposals to reviewers is, however, a very time-consuming task which requires scientific officers to manually screen available reviewers and assess their suitability to review given proposals. Such a matching process involves reading grant proposals, reading published works from reviewers, and in a consistent manner determining their similarity.

In order to support this matching procedure, we approach this task as a text similarity problem to leverage the benefits of natural language processing to pre-filter a subset of suitable reviewers. In particular, we use NLP models to vectorize the English texts of proposals and those of reviewers' publications. We then compute a text similarity

measure between the proposals and reviewers' publications. For each proposal we rank-order the similarity scores of all potential reviewers to retrieve the subset of best-matching reviewers. This subset then serves the scientific officers as a pre-filtered pool of suitable reviewers. Such pre-filtering substantially reduces the time needed to screen all possible reviewers and helps to more efficiently allocate the resources of the scientific officers. Similar NLP-based approaches of matching proposals to reviewers have been suggested in the domains of grant and journal peer review (Hettich and Pazzani, 2006; Stelmakh et al., 2021) as well as scientific conferences (Charlin and Zemel, 2013) and also from the big bibliometric databases (e.g. Dimensions, SpringerNature, Elsevier).

For the vectorization of the texts of proposals and reviewers' publications, we contrast a bag-of-words approach using the TF-IDF (Term Frequency - Inverse Document Frequency) weighting (Spärck Jones, 1972) with a word embeddings approach using pre-trained transformer models (Vaswani et al., 2017). In comparison to TF-IDF, transformers produce contextualized text embeddings thanks to their self-attention mechanism. Transformer models became widely used for semantic text similarity tasks (Reimers and Gurevych, 2019; Yang et al., 2020; Chandrasekaran and Mago, 2021), even though simple bag-of-words methods such as TF-IDF often perform equally well (Shahmirzadi et al., 2019). Given the vast amount of open-source pre-trained language models available (Wolf et al., 2020), the choice of a suitable model for a given setting is *a priori* not clear. Due to the specific scientific domain of the grant proposal texts as well as reviewers' publication texts, we focus on models pre-trained specifically on scientific texts in English and investigate the value added by such targeted pre-training in comparison to a model pre-trained on a general text corpus. As such, we consider the BERT model (Devlin et al.,

2018), being one of the most popular open-source pre-trained models, as our baseline model. We compare BERT to SciBERT (Beltagy et al., 2019), the BERT extension pre-trained additionally on scientific texts as well as to SPECTER (Cohan et al., 2020), which is a further extension of SciBERT via citation graph augmentation. In particular, we use SPECTER2 (Singh et al., 2022), an updated version of the original SPECTER model. As a benchmark model we consider the TF-IDF weighting. For each of the considered models, we vectorize the texts for both grant proposals and reviewers' publications and compute their cosine similarities. Based on the rank-ordered similarities, we select a subset of best matching reviewers for each grant proposal. As such, we effectively build a *recommender system* based on text similarities. We evaluate the performance of the models by contrasting the subset of best-matching reviewers with the actual reviewer matching based on a manual assignment by scientific officers.

The results reveal a clear pattern in favor of models with targeted pre-training on scientific texts. We observe substantially better performance of SciBERT in comparison to BERT, while SPECTER2 also considerably outperforms SciBERT. These findings provide clear evidence for the value added by targeted pre-training of base models on a specific text corpus for a matching/recommendation task based on text similarity. In particular, additional pre-training of BERT on scientific texts improves the overlap between the manually matched and model-generated subset of reviewers. In addition to pre-training on scientific texts, incorporating the inter-document relatedness via citation graph further improves the overlap. Despite the clear improvements of scientific pre-training of the BERT model, only the most sophisticated one, i.e. the SPECTER2, clearly outperforms the TF-IDF model. These results are robust to changes in the types of text data inputs such as title and abstract as well as the amount of text data provided. Furthermore, the results do not depend on specific modelling choices and are robust to changes in the text embedding extraction such as mean pooling or CLS tokens for BERT models and uni-grams or n-grams for the TF-IDF model.

The code for the conducted analyses is publicly available at <https://github.com/snsf-data/snsf-grant-similarity>.¹

¹Due to data protection laws, the data cannot be shared.

2 Institutional Setting

Based on a government mandate, the Swiss National Science Foundation (SNSF) supports scientific research in all academic disciplines. The SNSF is the leading Swiss organisation for the promotion of scientific research. The main role of the SNSF is the evaluation of scientific grant proposals; those that are evaluated to be the best are awarded research funding. Within the evaluation procedure, the SNSF relies on external peer-reviewers as well as on internal reviewers in the form of members of the evaluation panels. In this study, we focus on the latter evaluation step. For each evaluation panel, the grant proposals need to be matched to at least 2 reviewers from a pre-defined pool of available reviewers. These panel reviewers then assess the quality of the grant proposal based on the external peer reviews and their own evaluation of the proposals. In order to warrant fair and professional evaluation, the reviewers should have sufficient expertise in the fields of research of the respective grant proposals.

The matching of grant proposals to reviewers requires scientific officers to manually screen the grant proposal texts and the texts of reviewers' publications. Such a procedure is feasible if the number of proposals and reviewers is limited. However, it poses a great challenge as the number of proposals and potential reviewers grows. In order to reduce the manual labor, we approach the matching procedure as a text similarity problem. We leverage the benefits of the NLP models to vectorize the English texts from grant proposals and texts from reviewers' publications and compute their text similarities via cosine distance. For each grant proposal, we rank-order the similarity scores and select a subset of best-matching reviewers. In other words, we build a recommender system based on text similarities. We further need to take additional constraints into account, such as conflicts of interest and a maximum workload per reviewer. Finally, the suggested matching of proposals to reviewers is validated and approved by scientific officers before the final assignment takes place. This procedure can be summarized in the following steps:

1. Download publication metadata for each reviewer from a bibliometric database
2. Vectorize texts of reviewers and proposals
3. Match reviewers to proposals based on the highest text similarity

4. Balance number of proposals across reviewers
5. Validate matching results by scientific officers

In this paper, we focus on the above steps 2 and 3 and investigate the value of pre-training transformer models targeted towards the scientific domain in contrast to a simple bag-of-words approach, in order to determine the most efficient method of pre-filtering suitable reviewers.

3 Data

In general, it is challenging to objectively evaluate the performance of text vectorization methods for text similarity tasks as we cannot directly observe the true underlying text similarity (Reimers et al., 2016; Shah, 2022). In order to overcome this challenge, we evaluate the recommendations based on the text similarity and rely on a manually annotated dataset of matched reviewers provided by the scientific officers from the SNSF. In particular, we use the data from the *Postdoc.Mobility* funding scheme from the August 2021 call. *Postdoc.Mobility* fellowships enable early career researchers who have recently completed their doctorates and would like to pursue a scientific or academic career in Switzerland to conduct research projects abroad for up to two years. The data includes 398 submitted grant proposals across disciplines, and a pool of 150 potential reviewers, making it an arguably representative case. For each grant proposal, we observe the first-best and second-best reviewer according to the best knowledge of the scientific officers.² Most importantly, this matching does truly reflect the best possible assignment as it does not consider any additional constraints such as conflicts of interest or workload limits to manipulate the final assignment. Thus it can be used as a validation for evaluating the recommendations for matching based on the underlying similarity between the grant proposals and reviewer’s publications.

To assess the text similarity between the grant proposals and reviewer’s publications, we rely on the text of titles and abstracts. Titles and abstracts are often used for semantic text similarity tasks, especially in the scientific domain (Cohan et al., 2020) and should provide a condensed summary of the most important aspects of a scientific text. For grant proposals we retrieve the titles and abstracts

²The assignment by scientific officers has been done in accordance with the research area, whereas we do not restrict the model-generated assignment as such.

directly from the submitted proposal documents. For potential reviewers we download the titles and abstracts from their scientific publications from a bibliometric database.³ To ensure a clean evaluation setup we restrict the texts of titles and abstracts to English texts only, for both proposals and publications, and keep only those reviewers with at least 10 English publications available in the database. This leaves us with a set of 320 grant proposals and 125 potential reviewers.⁴ Table 1 below provides an overview of the data based on the research areas:⁵

Area	# Proposals (%)	# Reviewers (%)
SSH	50 (15.6)	20 (16.0)
MINT	147 (46.0)	62 (49.6)
LS	123 (38.4)	43 (34.4)

Table 1: Overview of Research Area Distribution

To investigate the influence of data inputs on the matching results, we vary the inputs along two dimensions. First, we vary the composition of the text data and compare the matching results based on 1) titles, 2) abstracts, and 3) concatenation of titles and abstracts, to explore the value of the particular types of texts. Second, we vary the amount of the text data and compare the publications from the last 5 years vs. publications from the last 10 years, to examine the importance of the publications’ recency. Note that although on average the increase in number of publications is proportionate to the recency of the publications, there is a lot of heterogeneity as well. Additionally, due to differences in publication practices, the actual number of publications varies substantially across disciplines.⁶

4 Methods

Since their introduction, transformer models (Vaswani et al., 2017) have gained considerable

³The present analysis uses data from the *Scopus* database of Elsevier. In the future the SNSF will base its matchings on the *Dimensions* database.

⁴These restrictions concern predominantly proposals and reviewers from the disciplines of social sciences and humanities due to the diverse type of outputs in these disciplines that are covered less completely in bibliometric databases.

⁵We follow the *official discipline classification of the SNSF* and distinguish between three high-level research areas: Human and Social Sciences (SSH), Mathematics, Natural- and Engineering Sciences (MINT), and Biology and Medicine (LS).

⁶The average number of publications per referee is 44.7 for the last 5 years of record and 82.0 for the last 10 years. For differences in research areas, see Table 3 in Appendix.

attention in the field of applied natural language processing (Tunstall et al., 2022). One of the key innovations of the transformer architecture is the self-attention mechanism, which helps to capture the context within the input sequence (Turner, 2023). As such, transformers provide a text vectorization in a form of *contextualized* text embeddings. Such contextualized embeddings can be used for a variety of NLP tasks, including semantic text similarity (Chandrasekaran and Mago, 2021). Furthermore, the availability of open-source pre-trained models on platforms such as Hugging Face makes it convenient to deploy these models for a particular application (see e.g. Wolf et al., 2020).

In this study, we focus on the BERT-type models (Devlin et al., 2018), i.e. deep bidirectional transformers, which have gained large popularity for a variety of applied NLP tasks. The BERT models are pre-trained on large text corpus via bidirectional representations, conditioning on both left and right context in the text sequence in all layers of the model (Devlin et al., 2018). The text corpus for the pre-training of the base BERT model consists of the BookCorpus (Zhu et al., 2015) and the English Wikipedia. Given the specific scientific domain in our setting of grant proposals and reviewers' publications we compare the base BERT model with its extended version that used additional scientific texts from SemanticScholar for pre-training, the so-called SciBERT (Beltagy et al., 2019), as well as with a further extension of the SciBERT itself - the SPECTER2 (Singh et al., 2022) - which has been further augmented by citation graph in its pre-training to capture the inter-document relatedness.

As collecting and labelling pairs of grant proposal text data for specific fine-tuning of the models is costly and often infeasible in practice due to the limited resources of the scientific officers, we focus on evaluating the pre-trained models as given, *without* additional fine-tuning. By doing so, we can effectively assess the value added by the specific pre-training of these models targeted towards scientific texts and their suitability for a matching/recommendation task based on the scientific text similarity. The pre-trained models as such can be used off-the-shelf for extracting the text embeddings via the so-called CLS token from the last hidden layer of the network, a classification token that provides an aggregate representation of the text sequence (Devlin et al., 2018; Cohan et al., 2020). An alternative representation for the text sequence

can be obtained by the so-called mean pooling, which averages all 512 tokens from the last hidden layer to get the text embedding. Such extractions of the embeddings from pre-trained models is common for a variety of NLP tasks (Kjell et al., 2023; Wu et al., 2023) as well as for text similarity in particular (May et al., 2019; Zhang et al., 2019; Qiao et al., 2019), although it has been pointed out by Reimers and Gurevych (2019) that such text embeddings might not lead to optimal performance unless fine-tuned specifically for text similarity task.

To benchmark the performance of the BERT models, we implement text vectorization via TF-IDF weighting (Spärck Jones, 1972). TF-IDF is a type of bag-of-words approach, where the numerical representation of the text in vector space is based on a token decomposition of the text, ignoring the sequential nature of the text. The TF-IDF then applies a weighting scheme that puts a higher weight on words that appear frequently in one document, but rarely across documents. The TF-IDF vectorization results in high-dimensional *sparse* vectors, which is in contrast to the *dense* vectors resulting from the BERT models. Such TF-IDF vectorization has proven to be very effective in text similarity tasks, despite its simplicity (compare e.g. Hettich and Pazzani, 2006; Shahmirzadi et al., 2019). We pre-process the texts for TF-IDF as follows: we lower-case the texts first and split the text sequence into separate words, i.e. tokens, while removing stop words and performing stemming of the remaining words.

To investigate the influence of the choice of text representation on the matching results, we evaluate the performance of the transformer models for both CLS token and mean pooling as these are the commonly used embedding extractions in practice (Reimers and Gurevych, 2019), as well as for uni-grams and 3-grams in the case of TF-IDF as these represent different levels of granularity of the text (Shahmirzadi et al., 2019).

The matching procedure can be defined as follows. Consider a grant proposal i with $i = 1, \dots, N$ in total, while each proposal i is associated with a single text sequence τ_i . Further consider a reviewer j with $j = 1, \dots, J$ reviewers in total, while each reviewer is associated with $k = 1, \dots, K$ text sequences, resulting in a reviewer-publication text sequence $\rho_{j,k}$. The raw text sequences are then vectorized via vectorization function $v_m(\cdot)$ depending on the model used

$M \in \{\text{BERT, SciBERT, SPECTER2, TF-IDF}\}$ resulting in the text vectors as:⁷

$$\begin{aligned} T_i &= v_m(\tau_i) \\ \text{for } i &= 1, \dots, N; \quad \forall m \in M \end{aligned} \quad (1)$$

and

$$\begin{aligned} P_{j,k} &= v_m(\rho_{j,k}) \\ \text{for } j &= 1, \dots, J \quad \text{and } k = 1, \dots, K; \quad \forall m \in M \end{aligned} \quad (2)$$

Then for all possible pairs of proposals and reviewers' publications, we estimate the text similarity via cosine distance:

$$\hat{\pi}_{i,j,k} = \frac{T_i \cdot P_{j,k}}{\|T_i\| \|P_{j,k}\|}. \quad (3)$$

In order to bring the similarities $\hat{\pi}_{i,j,k}$ onto proposal-reviewer level, for a given proposal-reviewer pair we sort the similarities along the publication level in a decreasing order as

$$\hat{\pi}_{i,j,(1)} \geq \hat{\pi}_{i,j,(2)} \geq \dots \geq \hat{\pi}_{i,j,(K)} \quad (4)$$

and average the similarities of the 20% most similar publications as follows⁸

$$\hat{\pi}_{i,j} = \frac{1}{K_{20}} \sum_{k=1}^{K_{20}} \hat{\pi}_{i,j,(k)}. \quad (5)$$

Given the average similarities between the pairs of proposals and reviewers $\hat{\pi}_{i,j}$, for each proposal i we rank-order the reviewers j according to their average similarities and select the top R ranked reviewers, with $R \in \{2, 5\}$, to provide a matching recommendation for a subset of suitable reviewers for each proposal as

$$\begin{aligned} \hat{J}_i^R &\in \text{Argmax}_j(\hat{\pi}_{i,j}) \\ \text{s.t. } |\hat{J}_i^R| &= R; \quad \forall R \in \{2, 5\}. \end{aligned} \quad (6)$$

In order to evaluate the quality of the matching recommendation, we compute the Mean Average Precision, i.e. MAP, a common metric for evaluation of recommender systems. (Chen and Liu, 2017). MAP is especially suitable in our case as

it takes the ordering information of the proposed matches into account. MAP combines both precision and recall as it approximates the average area under the so-called precision-recall curve (Schütze et al., 2008). In particular, MAP over all grant proposals N can be defined as follows:

$$MAP = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{P_i} \sum_{r=1}^R \mu(r) \cdot \frac{n(\tilde{J}_i^P \cap \hat{J}_i^R(r))}{r} \right) \quad (7)$$

where P_i is the number of true positive cases, i.e. the matches labelled by the scientific officers,⁹ $\mu(\cdot)$ is a so-called relevance function defined as an indicator function equal to 1 if the matched reviewer at rank r is relevant and 0 otherwise, \tilde{J}_i^P is a set of P true recommended reviewers as labelled by the scientific officers, and $\hat{J}_i^R(r)$, denotes indexing of the ordered set of R model recommended reviewers up to the r -th element. Intuitively, MAP equals 1 if the recommended matches correspond exactly to those labelled by the scientific officers for all grant proposals, while it equals 0 if we do not get any correct recommendations. As MAP takes the ordering information of the recommendations into account, even if we on average always do find the true two matches among $R = 5$ recommended ones, yet only at the 4th and 5th rank, the MAP value would correspond to 0.225. Similarly, if we on average find only a single match among the recommended ones, MAP would equal 0.5 if the match was on the first rank, but it would equal only to 0.1 if the match was on the fifth rank. This demonstrates how MAP is a distance sensitive metric and penalizes recommendations at lower ranks. In order to reflect the variability in MAP, we additionally compute the variance across the N proposals.

5 Results

Table 2 below presents the MAP results at $R = 5$, i.e. for the top 5 recommended reviewers, depending on the type of text embedding, number of recent years and the type of input text. We focus on the $R = 5$ case as our main objective is pre-filtering a subset of suitable reviewers, from which the scientific officers can easily choose the two most suitable reviewers. We provide the results for the case of $R = 2$ in Appendix to benchmark the results with

⁷We further suppress the dependence on a specific model m for notational ease.

⁸We tested the influence of this threshold by varying it between 10% and 50% and observed qualitatively similar results.

⁹In our setting, P_i is almost always equal to 2. For a handful of cases $P_i = 1$, if only a single reviewer with at least 10 English publications was available, as well as $P_i = 3$, if scientific officers labelled one extra reviewer as being suitable.

the case of pre-filtering the exact subset of reviewers needed for the final assignment.

Focusing on the first set of results based on the text embeddings via mean pooling, we observe a clear pattern for the BERT models. Regardless of the number of years and the type of input text sequence considered, the MAP is monotonically increasing when switching from BERT to SciBERT and further from SciBERT to SPECTER2. This documents the value added of targeted pre-training of the BERT model on scientific texts and additionally the citation graphs for the matching task based on the text similarity. Interestingly, the TF-IDF model based on 3-grams performs surprisingly well too, in many cases achieving similar performance as the SciBERT model. The unanimously best performance exhibits the SPECTER2 model, which benefits from the pre-training on the citation networks in addition to pre-training on scientific texts.

Looking at the differences based on the varying number of years and text inputs, we uncover additional clear patterns. First, the overall performance of all considered models is only marginally

better for the case of including last 10 years of publications instead of 5. As such, additional but less recent data on reviewer’s publications do not substantially improve the matching performance on average, although the improvement is greater for the SSH domain as will be discussed below. Second, we observe a sizeable increase in performance, when including abstracts in addition to titles, whereas the performance is *de facto* the same, whether abstracts are included alone or in combination with titles. This pattern is documented for all considered models. Thus, it appears that titles do not contain information that is not also available from the abstract.

Comparing the results based on the mean pooling with those of the CLS tokens, we identify few differences. For the BERT and SciBERT model the performance clearly deteriorates when only the CLS token is used, even more so for SciBERT than for BERT. This provides evidence in favor of text embeddings extraction via mean pooling for the matching task based on text similarity for these particular models. In contrast, the performance of the SPECTER2 model is robust, regardless of the type

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.3117 (0.1020)	0.3167 (0.1054)	0.3745 (0.1092)	0.2316 (0.0804)
mean pooling / 3-gram	5	abstract	0.3684 (0.1110)	0.3949 (0.1018)	0.4518 (0.1128)	0.3932 (0.1115)
mean pooling / 3-gram	5	title + abstract	0.3653 (0.1101)	0.3905 (0.1012)	0.4536 (0.1144)	0.3925 (0.1101)
mean pooling / 3-gram	10	title	0.3175 (0.1054)	0.3316 (0.1093)	0.3842 (0.1141)	0.2585 (0.0893)
mean pooling / 3-gram	10	abstract	0.3675 (0.1067)	0.4205 (0.1053)	0.4687 (0.1136)	0.4000 (0.1106)
mean pooling / 3-gram	10	title + abstract	0.3696 (0.1071)	0.4184 (0.1052)	0.4619 (0.1161)	0.4033 (0.1101)
CLS token / uni-gram	5	title	0.1937 (0.0743)	0.3104 (0.0968)	0.3908 (0.1110)	0.2305 (0.0767)
CLS token / uni-gram	5	abstract	0.2456 (0.0764)	0.2001 (0.0708)	0.4554 (0.1127)	0.3792 (0.1087)
CLS token / uni-gram	5	title + abstract	0.2719 (0.0921)	0.1941 (0.0668)	0.4520 (0.1170)	0.3692 (0.1020)
CLS token / uni-gram	10	title	0.2000 (0.0807)	0.3298 (0.1119)	0.4034 (0.1142)	0.2504 (0.0842)
CLS token / uni-gram	10	abstract	0.2718 (0.0921)	0.2123 (0.0716)	0.4605 (0.1227)	0.3900 (0.1077)
CLS token / uni-gram	10	title + abstract	0.2917 (0.1047)	0.1908 (0.0652)	0.4576 (0.1219)	0.3811 (0.1036)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 2: Results on the Mean Average Precision (MAP) at R=5 across models

of embedding. In case of TF-IDF, the results are also quite robust to the type of vectorization as the performance based on uni-grams is similar to that of 3-grams. In terms of the differences based on the number of years and the type of text inputs, we observe the same patterns as for the mean pooling and 3-grams respectively.

We further investigate the overall results by uncovering the heterogeneity with respect to research areas (see Tables 5, 7 and 9 in Appendix).¹⁰ In general, we observe similar patterns in terms of the performance of the considered models. However, we observe a substantial differences in the performance of all the models across the research areas. Most importantly, the results reveal lower performance for the SSH domain in particular. This might be partly due to the under-representation of the SSH domain within the publication data (see Table 3 in Appendix for details). In this regard, for SSH domain we observe a sizeable improvement in the performance of the SPECTER2 model in particular, when including texts of the past 10 years as opposed to 5 years, as can be seen in Tables 5 and 6 in the Appendix. This suggests that including more publication data is valuable for a better matching of reviewers in the SSH domain.¹¹ Interestingly, TF-IDF performs rather well for the SSH domain, although the contextual information that might be particularly important is not taken into account by this method.

Comparing the overall results with the case of recommending a subset of top 2 most similar reviewers, i.e. $R = 2$, we generally observe the same patterns as for the case of $R = 5$. Based on the conducted analyses, the SPECTER2 model provides the best and most robust performance across different model choices, data inputs, and research areas. Interestingly, a classical TF-IDF model turns out to be also well-performing and robust choice for the matching task based on text similarity.

6 Discussion

In this study, we investigated the value of pre-training BERT models towards scientific domain for the matching task based on text similarity and

¹⁰Results for $R = 2$ by research area are provided in Appendix in Tables 6, 8 and 10.

¹¹This improvement might stem from the increased amount of text data itself as well as from the content of the text data which might be more similar across time for SSH than for other research areas, or perhaps that fields within SSH are more distinct from other fields even as the fields themselves change across time.

compared the performance with a classical bag-of-words approach. The results reveal two main findings: First, pre-training on scientific texts and additionally considering the citation networks clearly improves the overlap between the actual and the recommended proposal-reviewer matches. Second, BERT models do not substantially out-perform TF-IDF in the matching tasks, unless both scientific documents and the citation networks are taken into account in the pre-training, i.e. the SPECTER2 model.

These results are in line with the findings of Shahmirzadi et al. (2019), who find the TF-IDF model to perform equally well as other more complex neural models. Nevertheless, the similar performance of the transformer models and TF-IDF is rather surprising, given the large conceptual differences in the text vectorization. One of the possible reasons for this phenomenon might be the fact that extracting raw BERT embeddings is not optimal, unless specifically fine-tuned for the task of text similarity as argued by Reimers and Gurevych (2019). This has also been the approach pursued by Yang et al. (2020) to compare the performance of transformer models for text similarity task in a clinical domain.

Furthermore, the results reveal substantial heterogeneity in the performance across research areas. For all considered models the matching task is the most challenging within the SSH domain. This might be due in part to the diverse type of outputs in these disciplines that are covered less completely in bibliometric databases, for which text similarity might not be the optimal approach. In addition, the large variety of disciplines within the SSH domain might pose another complication for the models considered here, as opposed to domains of MINT and LS, where the proposal and publication texts share more similar characteristics overall. SSH texts sometimes use more generic terminology with less specific keywords than what is found in MINT and LS, and we had hypothesized that methods based on text embeddings would benefit from the incorporation of large contexts, but this did not turn out to be the case. One of the possibilities to overcome this challenge might be an explicit fine-tuning of Siamese networks as suggested by Reimers and Gurevych (2019) on pairs of SSH texts.

Overall, the results presented in this study contribute to a better understanding of the usage of

pre-trained transformer models vs. classical bag-of-words models for a matching task based on text similarity in a scientific domain. The findings of our analyses provide empirical evidence on the suitability and sensitivity of the particular models, data inputs and modelling choices, for matching grant proposals to reviewers - a core task of any research funding agency.

Limitations

The analyses presented in this study have a limited scope. Firstly, the limitations concern the external validity of the results. As our validation dataset focuses on a specific call from a specific funding scheme at the SNSF with a relatively small sample size, it is not assured that the findings are representative for other funding schemes within the SNSF, or broader, for other funding agencies.

Secondly, restricting the data to English texts prevents the assessment of all submitted grant proposals and all potential reviewers. Such restriction further aggravates the imbalances in the availability of text data across research areas, resulting in lower representation of the SSH domain.

Thirdly, our analyses are limited to comparison of BERT models and the TF-IDF model for text vectorization. Therefore, our findings are not representative for newer open-source transformer models such as Llama (Touvron et al., 2023) or Mistral (Jiang et al., 2023), or for other alternative text vectorization methods such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014), which might potentially out-perform the approaches analyzed here. Additionally, due to the token length limited to 512 tokens for the pre-trained BERT models, the texts are truncated at this threshold, which leads to occasional information loss. We have experimented with truncation from the left and right of the text sequences, which did not change the qualitative conclusions.

Lastly, extracting raw embeddings from the pre-trained BERT models without explicitly fine-tuning the models for the text similarity task might result in sub-optimal performance. Nevertheless, it should provide a reasonable baseline in cases where labelled data is not feasible to collect.

Reproducibility

The code used to conduct this analysis is available at <https://github.com/snsf-data/snsf-grant-similarity>. Due to the data protection laws, the data

used in this study cannot be shared.

Ethics Statement

The results from the NLP algorithm have never been used to directly assign reviewers to grant applications without a validation from the scientific officers. Scientific officers always check and validate the suggested matching and adjust the assignment as necessary.

Acknowledgements

This work could not have been completed without the efforts from our colleagues in Postdoc.Mobility who diligently labeled an entire call from August 2021 with the true best matching reviewers and co-reviewers for each proposal, before considering practical requirements such as conflicts of interest, so that we have a dataset that can be used for testing algorithm variations. This dataset has been invaluable, and we would not have been able to progress on this work without their help. In addition, we would like to thank the four anonymous reviewers for helpful comments and suggestions on a previous version of this manuscript.

Competing Interests

Both authors are employed by the SNSF.

References

- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.
- Kenneth Benoit, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30):774–774.
- Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37.
- Laurent Charlin and Richard Zemel. 2013. The Toronto paper matching system: an automated paper-reviewer assignment system.
- Mingang Chen and Pan Liu. 2017. Performance evaluation of recommender systems. *International Journal of Performability Engineering*, 13(8):1246.
- Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Seth Hettich and Michael J Pazzani. 2006. Mining for proposal reviewers: lessons learned at the national science foundation. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 862–871.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Oscar Kjell, Salvatore Giorgi, and H Andrew Schwartz. 2023. The text-package: An R-package for analyzing and visualizing human language using natural language processing and transformers. *Psychological Methods*.
- Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of BERT in ranking. *arXiv preprint arXiv:1904.07531*.
- Nils Reimers, Philip Beyer, and Iryna Gurevych. 2016. Task-oriented intrinsic evaluation of semantic textual similarity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 87–96.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084*.
- Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.
- Nihar B Shah. 2022. Challenges, experiments, and computational solutions in peer review. *Communications of the ACM*, 65(6):76–87.
- Omid Shahmirzadi, Adam Lugowski, and Kenneth Young. 2019. Text similarity in vector space models: a comparative study. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 659–666. IEEE.
- Amanpreet Singh, Mike D’Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2022. SciRepEval: A multi-format benchmark for scientific document representations. *arXiv preprint arXiv:2211.13308*.
- Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.
- Ivan Stelmakh, Nihar Shah, and Aarti Singh. 2021. Peer-Review4All: Fair and accurate reviewer assignment in peer review. *Journal of Machine Learning Research*, 22(163):1–66.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. 2022. *Natural language processing with transformers*. " O’Reilly Media, Inc."
- Richard E Turner. 2023. An introduction to transformers. *arXiv preprint arXiv:2304.10557*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. HuggingFace’s transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.
- Letian Wu, Wenyao Zhang, Tengping Jiang, Wankou Yang, Xin Jin, and Wenjun Zeng. 2023. [CLS] token is all you need for zero-shot semantic segmentation. *arXiv preprint arXiv:2304.06212*.
- Xi Yang, Xing He, Hansi Zhang, Yinghan Ma, Jiang Bian, Yonghui Wu, et al. 2020. Measurement of semantic textual similarity in clinical texts: comparison of transformer-based models. *JMIR medical informatics*, 8(11):e19735.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. *arXiv preprint arXiv:1904.09675*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Appendix

A.1 Descriptive Statistics

Publications	SSH	MINT	LS
5 years	22.6	52.1	44.4
10 years	40.6	96.0	81.2

Table 3: Distribution of the average number of publications per research area

A.2 Model Details

In our analyses, we deploy specifically the following models from the Hugging Face platform (Wolf et al., 2020):

- BERT: `google-bert/bert-base-uncased`
- SciBERT: `allenai/scibert_scivocab_uncased`
- SPECTER2: `allenai/specter2_base`

and use the `quanteda` (Benoit et al., 2018) implementation for the TF-IDF vectorization.

A.3 Supplementary Results

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.2383 (0.0882)	0.2398 (0.0961)	0.2945 (0.1032)	0.1883 (0.0763)
mean pooling / 3-gram	5	abstract	0.2734 (0.1100)	0.2938 (0.0988)	0.3562 (0.1188)	0.3117 (0.1073)
mean pooling / 3-gram	5	title + abstract	0.2719 (0.1104)	0.2867 (0.0956)	0.3602 (0.1181)	0.3102 (0.1071)
mean pooling / 3-gram	10	title	0.2383 (0.0968)	0.2531 (0.1038)	0.2914 (0.1070)	0.2055 (0.0805)
mean pooling / 3-gram	10	abstract	0.2734 (0.1021)	0.3117 (0.1026)	0.3703 (0.1203)	0.3023 (0.1076)
mean pooling / 3-gram	10	title + abstract	0.2758 (0.1053)	0.3125 (0.0995)	0.3680 (0.1238)	0.3094 (0.1109)
CLS token / uni-gram	5	title	0.1391 (0.0676)	0.2266 (0.0849)	0.2938 (0.0992)	0.1797 (0.0715)
CLS token / uni-gram	5	abstract	0.1727 (0.0698)	0.1461 (0.0622)	0.3578 (0.1161)	0.3078 (0.1048)
CLS token / uni-gram	5	title + abstract	0.1938 (0.0787)	0.1414 (0.0585)	0.3586 (0.1192)	0.2914 (0.0953)
CLS token / uni-gram	10	title	0.1469 (0.0665)	0.2508 (0.0997)	0.3078 (0.1064)	0.1906 (0.0740)
CLS token / uni-gram	10	abstract	0.1969 (0.0810)	0.1500 (0.0578)	0.3648 (0.1280)	0.3008 (0.1069)
CLS token / uni-gram	10	title + abstract	0.2070 (0.0932)	0.1328 (0.0532)	0.3656 (0.1265)	0.2906 (0.0994)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 4: Results on the Mean Average Precision (MAP) at R=2 across models

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.2402 (0.1119)	0.2273 (0.0936)	0.2513 (0.1147)	0.1512 (0.0526)
mean pooling / 3-gram	5	abstract	0.3080 (0.1282)	0.3020 (0.1060)	0.3327 (0.1193)	0.3533 (0.1278)
mean pooling / 3-gram	5	title + abstract	0.2973 (0.1277)	0.2983 (0.1075)	0.3492 (0.1319)	0.3603 (0.1222)
mean pooling / 3-gram	10	title	0.2440 (0.0951)	0.2128 (0.0809)	0.2647 (0.0838)	0.1840 (0.0622)
mean pooling / 3-gram	10	abstract	0.2915 (0.1095)	0.3150 (0.0960)	0.4070 (0.1270)	0.3312 (0.0844)
mean pooling / 3-gram	10	title + abstract	0.2913 (0.1057)	0.3388 (0.1033)	0.3908 (0.1320)	0.3562 (0.0910)
CLS token / uni-gram	5	title	0.1873 (0.0943)	0.2213 (0.0983)	0.2502 (0.0968)	0.1245 (0.0402)
CLS token / uni-gram	5	abstract	0.1617 (0.0621)	0.2017 (0.0797)	0.3325 (0.1135)	0.3048 (0.1166)
CLS token / uni-gram	5	title + abstract	0.1690 (0.0566)	0.1735 (0.0562)	0.3038 (0.1164)	0.2925 (0.1112)
CLS token / uni-gram	10	title	0.2098 (0.0957)	0.2237 (0.1040)	0.2643 (0.0666)	0.1643 (0.0481)
CLS token / uni-gram	10	abstract	0.2240 (0.0895)	0.2060 (0.0739)	0.3775 (0.1345)	0.3328 (0.0948)
CLS token / uni-gram	10	title + abstract	0.2318 (0.0744)	0.1657 (0.0694)	0.3493 (0.1212)	0.3140 (0.0883)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 5: **Research Area SSH** - Mean Average Precision (MAP) at R=5 across models

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.1900 (0.1060)	0.1800 (0.0945)	0.2000 (0.1097)	0.1100 (0.0489)
mean pooling / 3-gram	5	abstract	0.2300 (0.1271)	0.1850 (0.1067)	0.2500 (0.1250)	0.2800 (0.1266)
mean pooling / 3-gram	5	title + abstract	0.2250 (0.1256)	0.1950 (0.1053)	0.2950 (0.1472)	0.2850 (0.1148)
mean pooling / 3-gram	10	title	0.1700 (0.0700)	0.1500 (0.0791)	0.1850 (0.0811)	0.1400 (0.0565)
mean pooling / 3-gram	10	abstract	0.2050 (0.1089)	0.1850 (0.0939)	0.3050 (0.1385)	0.2150 (0.0817)
mean pooling / 3-gram	10	title + abstract	0.2150 (0.1097)	0.2350 (0.0980)	0.3050 (0.1385)	0.2350 (0.0980)
CLS token / uni-gram	5	title	0.1400 (0.0820)	0.1750 (0.0950)	0.1900 (0.0882)	0.0850 (0.0398)
CLS token / uni-gram	5	abstract	0.1150 (0.0618)	0.1650 (0.0832)	0.2550 (0.1186)	0.2300 (0.0965)
CLS token / uni-gram	5	title + abstract	0.1100 (0.0514)	0.1300 (0.0593)	0.2350 (0.1107)	0.2250 (0.0950)
CLS token / uni-gram	10	title	0.1400 (0.0667)	0.1750 (0.0925)	0.1900 (0.0678)	0.1000 (0.0383)
CLS token / uni-gram	10	abstract	0.1850 (0.0862)	0.1550 (0.0788)	0.3100 (0.1341)	0.2200 (0.0960)
CLS token / uni-gram	10	title + abstract	0.1850 (0.0709)	0.1250 (0.0721)	0.2900 (0.1157)	0.2100 (0.0851)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 6: **Research Area SSH** - Mean Average Precision (MAP) at R=2 across models

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.2556 (0.0766)	0.2711 (0.0846)	0.3758 (0.0896)	0.2487 (0.0869)
mean pooling / 3-gram	5	abstract	0.3530 (0.1016)	0.3874 (0.0931)	0.4634 (0.1034)	0.3770 (0.1003)
mean pooling / 3-gram	5	title + abstract	0.3493 (0.0955)	0.3780 (0.0917)	0.4584 (0.1024)	0.3794 (0.1050)
mean pooling / 3-gram	10	title	0.2502 (0.0759)	0.2870 (0.0805)	0.3884 (0.1026)	0.2675 (0.0892)
mean pooling / 3-gram	10	abstract	0.3411 (0.0960)	0.4239 (0.0972)	0.4583 (0.0998)	0.3686 (0.0963)
mean pooling / 3-gram	10	title + abstract	0.3500 (0.0964)	0.4054 (0.0948)	0.4460 (0.0945)	0.3752 (0.1023)
CLS token / uni-gram	5	title	0.1315 (0.0480)	0.2673 (0.0740)	0.3840 (0.0910)	0.2496 (0.0782)
CLS token / uni-gram	5	abstract	0.2370 (0.0768)	0.1884 (0.0613)	0.4353 (0.1000)	0.3752 (0.1024)
CLS token / uni-gram	5	title + abstract	0.2744 (0.0912)	0.1694 (0.0517)	0.4364 (0.1010)	0.3713 (0.0998)
CLS token / uni-gram	10	title	0.1446 (0.0551)	0.2920 (0.0901)	0.4091 (0.1071)	0.2558 (0.0821)
CLS token / uni-gram	10	abstract	0.2514 (0.0875)	0.1970 (0.0630)	0.4430 (0.1050)	0.3619 (0.1012)
CLS token / uni-gram	10	title + abstract	0.2761 (0.1097)	0.1775 (0.0510)	0.4363 (0.1049)	0.3622 (0.0997)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 7: **Research Area LS** - Mean Average Precision (MAP) at R=5 across models

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.1951 (0.0671)	0.2134 (0.0816)	0.2927 (0.0960)	0.2114 (0.0851)
mean pooling / 3-gram	5	abstract	0.2663 (0.1063)	0.2967 (0.0987)	0.3638 (0.1109)	0.3008 (0.1004)
mean pooling / 3-gram	5	title + abstract	0.2561 (0.0999)	0.2846 (0.0905)	0.3598 (0.1057)	0.3008 (0.1045)
mean pooling / 3-gram	10	title	0.1768 (0.0725)	0.2195 (0.0785)	0.2927 (0.1022)	0.2175 (0.0840)
mean pooling / 3-gram	10	abstract	0.2622 (0.0941)	0.3313 (0.1050)	0.3638 (0.1109)	0.2622 (0.0869)
mean pooling / 3-gram	10	title + abstract	0.2642 (0.0987)	0.3191 (0.0976)	0.3516 (0.1074)	0.2785 (0.0975)
CLS token / uni-gram	5	title	0.0915 (0.0443)	0.1850 (0.0675)	0.2785 (0.0812)	0.2012 (0.0724)
CLS token / uni-gram	5	abstract	0.1585 (0.0710)	0.1341 (0.0546)	0.3435 (0.1070)	0.3069 (0.1002)
CLS token / uni-gram	5	title + abstract	0.1911 (0.0780)	0.1179 (0.0423)	0.3455 (0.1071)	0.2907 (0.0895)
CLS token / uni-gram	10	title	0.1179 (0.0526)	0.2195 (0.0805)	0.3150 (0.1043)	0.2053 (0.0738)
CLS token / uni-gram	10	abstract	0.1829 (0.0769)	0.1484 (0.0572)	0.3415 (0.1140)	0.2663 (0.1001)
CLS token / uni-gram	10	title + abstract	0.1931 (0.0971)	0.1220 (0.0362)	0.3455 (0.1163)	0.2703 (0.0949)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 8: **Research Area LS** - Mean Average Precision (MAP) at R=2 across models

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.3830 (0.1118)	0.3852 (0.1189)	0.4154 (0.1184)	0.2448 (0.0828)
mean pooling / 3-gram	5	abstract	0.4018 (0.1120)	0.4327 (0.1046)	0.4827 (0.1141)	0.4204 (0.1155)
mean pooling / 3-gram	5	title + abstract	0.4018 (0.1147)	0.4324 (0.1035)	0.4852 (0.1153)	0.4145 (0.1109)
mean pooling / 3-gram	10	title	0.3988 (0.1227)	0.4093 (0.1318)	0.4213 (0.1292)	0.2762 (0.0975)
mean pooling / 3-gram	10	abstract	0.4154 (0.1112)	0.4536 (0.1118)	0.4985 (0.1199)	0.4497 (0.1279)
mean pooling / 3-gram	10	title + abstract	0.4126 (0.1138)	0.4563 (0.1122)	0.4994 (0.1270)	0.4428 (0.1215)
CLS token / uni-gram	5	title	0.2478 (0.0844)	0.3768 (0.1080)	0.4444 (0.1243)	0.2505 (0.0841)
CLS token / uni-gram	5	abstract	0.2813 (0.0782)	0.2094 (0.0765)	0.5140 (0.1156)	0.4078 (0.1102)
CLS token / uni-gram	5	title + abstract	0.3048 (0.1014)	0.2217 (0.0824)	0.5153 (0.1205)	0.3935 (0.0996)
CLS token / uni-gram	10	title	0.2430 (0.0937)	0.3976 (0.1245)	0.4459 (0.1292)	0.2752 (0.0960)
CLS token / uni-gram	10	abstract	0.3051 (0.0958)	0.2273 (0.0785)	0.5033 (0.1309)	0.4329 (0.1154)
CLS token / uni-gram	10	title + abstract	0.3251 (0.1096)	0.2106 (0.0757)	0.5123 (0.1306)	0.4197 (0.1100)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 9: **Research Area MINT** - Mean Average Precision (MAP) at R=5 across models

Embedding	Years	Text	BERT	SciBERT	SPECTER2	TF-IDF
mean pooling / 3-gram	5	title	0.2908 (0.0959)	0.2823 (0.1064)	0.3282 (0.1043)	0.1956 (0.0766)
mean pooling / 3-gram	5	abstract	0.2942 (0.1076)	0.3282 (0.0923)	0.3861 (0.1201)	0.3316 (0.1072)
mean pooling / 3-gram	5	title + abstract	0.3010 (0.1138)	0.3197 (0.0940)	0.3827 (0.1184)	0.3265 (0.1075)
mean pooling / 3-gram	10	title	0.3129 (0.1172)	0.3163 (0.1261)	0.3265 (0.1161)	0.2177 (0.0850)
mean pooling / 3-gram	10	abstract	0.3061 (0.1051)	0.3384 (0.0983)	0.3980 (0.1214)	0.3656 (0.1269)
mean pooling / 3-gram	10	title + abstract	0.3061 (0.1086)	0.3333 (0.1005)	0.4031 (0.1314)	0.3605 (0.1225)
CLS token / uni-gram	5	title	0.1786 (0.0796)	0.2789 (0.0921)	0.3418 (0.1131)	0.1939 (0.0786)
CLS token / uni-gram	5	abstract	0.2041 (0.0702)	0.1497 (0.0622)	0.4048 (0.1184)	0.3350 (0.1100)
CLS token / uni-gram	5	title + abstract	0.2245 (0.0862)	0.1650 (0.0715)	0.4116 (0.1257)	0.3146 (0.0994)
CLS token / uni-gram	10	title	0.1735 (0.0776)	0.3027 (0.1141)	0.3418 (0.1165)	0.2092 (0.0839)
CLS token / uni-gram	10	abstract	0.2126 (0.0834)	0.1497 (0.0519)	0.4031 (0.1365)	0.3571 (0.1113)
CLS token / uni-gram	10	title + abstract	0.2262 (0.0979)	0.1446 (0.0616)	0.4082 (0.1362)	0.3350 (0.1049)

Note: Higher MAP values indicate better performance. Variance displayed below in parentheses.

Table 10: **Research Area MINT** - Mean Average Precision (MAP) at R=2 across models

Tracing Linguistic Footprints of ChatGPT Across Tasks, Domains and Personas in English and German

Anastassia Shaitarova, Nikolaj Bauer, Jannis Vamvas, Martin Volk

Department of Computational Linguistics, University of Zurich

{shaita, vamvas, volk}@cl.uzh.ch, nikolaj.bauer@uzh.ch

Abstract

Large language models like ChatGPT can be used to generate seemingly human-like text. However, it is still not well understood how their output differs from text written by humans, and to what degree prompting influences their linguistic profile. In our paper, we instruct ChatGPT to complete, explain and create texts in English and German across journalistic, scientific, and clinical domains. We assign corpus-specific personas to the system setting as part of the prompt within each task. We extract a large number of linguistic features and perform statistical and qualitative comparison across text pairs. Our results show that prompting makes a larger impact on English output than on German. Most basic features such as mean word length distinctly set human and generated texts apart. Readability metrics indicate that ChatGPT overcomplicates English texts, particularly in the clinical domain, while German-generated texts suffer from excessive morpho-syntactic standardization coupled with lexical simplification.

1 Introduction

Instruction-tuned conversational Large Language Models (LLMs), such as ChatGPT (OpenAI, 2022), are now widely used by the general public due to their friendly conversational setup and unprecedented linguistic capabilities. The rate of LLM usage is remarkable, with ChatGPT alone generating an ‘equivalent to all the printed works of humanity’ every two weeks shortly after its release¹. This trend shows no signs of subsiding. Although generated texts are consumed by the public and reused in model training, their linguistic composition remains poorly understood. The proprietary nature of most prominent models exacerbates the issue,

¹https://www.nber.org/system/files/working_papers/w30957/w30957.pdf

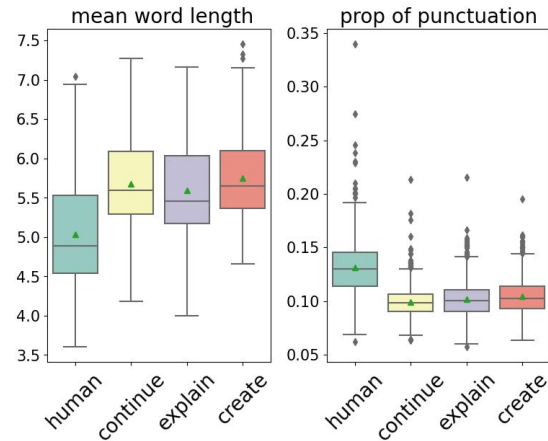


Figure 1: The linguistic footprint of ChatGPT in generated output is best observed through basic features like word length and proportion of punctuation. The figure displays results for two significant features measured across combined English and German data, comparing texts produced by humans and three generative tasks.

post-hoc analysis of the textual output being the main form of research.

A strong line of research is dedicated to the detection of generated texts. Human readers are no longer able to identify them (Brown et al., 2020; Dou et al., 2022), but their textual patterns can still be traced statistically (Levin et al., 2023; Mitrović et al., 2023; Guo et al., 2023; Liu et al., 2023). LLMs are highly versatile; for instance, prompt alterations can have a significant impact on the output (Tang et al., 2023), however not necessarily increasing textual human likeness (Tseng et al., 2023). To the best of our knowledge, only Deshpande et al. (2023); Tseng et al. (2023) addressed the linguistic composition of texts conditioned on the persona system parameter. However, there is still much to be explored in this area.

In our paper, we aim to bridge this gap by investigating the impact of different tasks and personas on the texts generated by ChatGPT. We collected five corpora in both English and German, encom-

passing journalistic articles, academic papers, and clinical texts. On their basis, we generated comparable datasets using prompts constructed from excerpts of human-authored texts, domain-specific instructions, and tailored persona settings. Moreover, we conducted a comprehensive statistical analysis comparing lexical, syntactic, and stylistic features across languages, tasks, and domains².

Our findings reveal several key insights: ① The English textual profile of our generated output is more pronounced than German (Table 4), emphasizing the importance of language-specific evaluations; ② The statistical footprint left by the model is most prominent in general textual features such as word length and punctuation usage (Figure 1); ③ The generated texts demonstrate lower readability scores, particularly in English (Figure 2); ④ The significance of features varies across languages and domains (Figure 2); ⑤ German academic ChatGPT personas exhibit a tendency to overuse capitalized connectives and more complex lexical options (Figure 3).

2 Previous Work

Without additional prompt manipulations, ChatGPT produces texts that are well-organized and coherent (Ariyaratne et al., 2023; Liu et al., 2023), informative and objective (Guo et al., 2023), characteristics typical for academic papers or official documents. ChatGPT writes as a ‘conservative team of experts’ (Guo et al., 2023), providing a comprehensive and neutral view. On the lexical level, this tendency manifests itself through a high number of nouns, adpositions, and adjectives, together with frequently co-occurring conjunctions and cohesion markers like “in general”, “firstly”, “secondly”, “finally”. Overall, Guo et al. (2023), who worked with question-answer pairs in open domain, computer science, finance, medicine, law, and psychology, noted that ChatGPT provides longer texts with a poorer vocabulary, a tendency also observed by Liu et al. (2023) in argumentative essay writing. Conversely, Mitrović et al. (2023) witnessed ChatGPT use vocabulary items that humans consider ‘fancy and atypical’ for the domain, i.e. “stand out feature”, “waitstaff” and “knowledgeable” in restaurant reviews. Manifestations of emotions and individuality such as personal pronouns, impolite

²Code and data: <https://github.com/shaitarAn/LinguisticFootprintsChatGPT>; <https://doi.org/10.5281/zenodo.11109705>

expressions, or the use of punctuation to show emotions, strongly indicate human-authored texts.

Nevertheless, lexical composition and even politeness, can be altered with prompt modifications. Pu and Demberg (2023) showed that lexical diversity of the ChatGPT output is strongly influenced by the writing style indicated in the prompt. They used lexical diversity and automatic readability metrics to assess whether ChatGPT can cater its academic summaries to layman and expert readers. The generated lexical diversity was considerably lower in informal sentences, but much higher than human in formal texts. Overall, providing examples in the prompt (few-shot learning) significantly improved the stylistic adaptation. In accordance with other publications, Pu and Demberg observed a high ratio of adjectives, adpositions, and nouns in the ChatGPT-generated formal sentences, whereas informal texts featured more auxiliary words and punctuation marks.

Considering that modification of the system parameter, i.e. the persona setting, became available only recently, there is limited research available on this matter. Deshpande et al. (2023) performed a large-scale, systematic analysis of toxicity in the generated language conditioned on different ChatGPT personas. They created a list of 90 politicians, dictators, journalists, entrepreneurs and athletes and discovered that, despite moderation efforts, assigning a persona unleashes the model’s capacity for significantly toxic language. Tseng et al. (2023) experimented with different prompts, including generated personas, to produce comments on Dutch news articles and then analysed the output in terms of lexical diversity and general human-likeness. They used the Controlled Type-Token Ratio metric to show that human-written comments have a much higher lexical diversity, as opposed to ChatGPT-generated comments.

Overall, existing research provides only general linguistic profiling of the ChatGPT-produced text. In our paper we use three domains in two languages, conditioning the output on tasks and personas, and scrutinizing it with a broad spectrum of linguistic features.

3 Data

Our data comprise five datasets in English along with five comparable counterparts in German, spanning three domains. We included academic articles and clinical texts because these domains are signif-

	pubmed_en	zora_en	cnn	csb_en	e3c	pubmed_de	zora_de	20min	csb_de	ggponc
human	95,062	7,963	80,171	96,498	54,515	66,573	7,869	60,277	94,883	116,135
explain	74,766	7,350	72,638	69,616	65,651	68,933	7,177	70,406	71,263	76,088
continue	70,133	7,573	59,910	63,867	68,685	77,869	7,766	78,711	80,229	78,777
create	66,598	7,336	59,674	61,750	67,085	73,737	8,834	83,471	68,420	77,610
texts	100	10	100	100	100	96	10	100	100	100

Table 1: Dataset statistics showing the number of texts and tokens in human and generated sections of each corpus.

icantly impacted by the accessibility of generative LLMs like ChatGPT, posing potential high-risk but also high-reward scenarios. We also collected journalistic texts to align our results with those of previous studies. Table 1 provides an overview of the untruncated sizes of each corpus.

3.1 Clinical texts

E3C The European Clinical Case Corpus (E3C) (Minard et al., 2021) comprises clinical cases in Italian, English, French, Spanish and Basque. For the English part, Minard et al. used the PubMed API to automatically extract clinical case descriptions from published academic papers. Out of 10,034 available clinical texts in English, we were able to collect 100 that met the desired length of about 500 tokens. The E3C texts exhibit a writing style characterized by clarity, precision, and a focus on medical details, utilizing specific medical terminology and technical details.

GGPONC The German Guideline Program in Oncology NLP Corpus (GGPONC) is a large corpus of clinical guidelines for oncology (Borchert et al., 2022). It does not contain information about specific patients and therefore has no restrictions on access due to privacy protection. Version 2.0 of the GGPONC contains 30 guidelines with more than 1.8 million tokens. We randomly sampled 100 documents that were longer than 500 tokens. 26 of the original 30 guidelines are represented in our data, the most prominent being Palliative Medicine and Breast Cancer. The writing style is characterized by the use of technical language, structured organization, the use of citations, medical abbreviations, and numerical data. The tone is impersonal and objective throughout.

3.2 Journalistic writing

20 Minuten The 20 Minuten corpus (Kew et al., 2023) contains articles from a free Swiss daily newspaper published between the years 2010 and 2022. We randomly sampled 100 articles from five

different publication years. The texts vary in writing style depending on the content and the main message. They range from personal narratives and informal interviews with a conversational and empathetic tone to factual reporting adhering to journalistic writing standards.

CNN The CNN corpus is a large question answering corpus in English (Hermann et al., 2015), containing CNN articles published online between 2011 and 2015. We randomly sampled 100 articles with more than 500 tokens. CNN articles aim to present news in an objective and informative manner making emphasis on clarity, conciseness, and directness in the writing, while avoiding jargon and complex language to ensure broad accessibility.

Credit Suisse Bulletin The Credit Suisse Bulletin corpus (CSB: Volk et al., 2016) is a digitized multilingual diachronic collection of texts from the world’s oldest banking magazine, published by Credit Suisse³. The corpus covers diverse topics, including economy, culture, sport, and entertainment, in several languages. We made a random selection of 100 articles from the German-English PDF sub-corpus ranging from 1998 to 2017⁴. The writing style of the CSB texts varies depending on the topic. It is formal, clear, straightforward, and informative, offering insights into specific issues. At times, it adopts a technical or analytical tone. Though not explicitly stated, the original language of the articles is presumably German.

3.3 Scientific articles

PubMed The German and English PubMed corpora contain biomedical articles collected from the PubMed Central Database⁵. We downloaded a list of PubMed IDs and used the Bio.Entrez package⁶

³https://en.wikipedia.org/wiki/Credit_Suisse

⁴pub.cl.uzh.ch/projects/b4c

⁵<https://pubmed.ncbi.nlm.nih.gov/>

⁶<https://biopython.org/docs/1.75/api/Bio.Entrez.html>

to search for English and German articles containing both the abstract and the Introduction section (DE: *Einleitung*) that is more than 500 tokens in length. Our final corpus contains 96 German and 100 English articles.

Zora The Zurich Open Repository and Archive⁷, is a database of the University of Zurich with open access to scholarly articles in different languages. We collected ten articles from linguistics in both English and German.

The writing style of PubMed and Zora articles prioritizes clarity, precision, and formality within the academic context, catering primarily to subject-matter experts. It maintains objectivity with passive voice and third-person pronouns, emphasizes data-driven conclusions, and presents information concisely and with clear transitions.

4 Experiments

Implementation details In our experiments, we queried gpt-3.5-turbo-16K, a version of the ChatGPT model that allows for larger context window inputs. We used pilot experiments to rule out temperature settings above 1 due to the generation of illegible output. In order to address the issue of a less extensive vocabulary compared to human writing (Tseng et al., 2023), we kept the temperature setting at 1, which is the API’s default. This setting is expected to produce more creative and diverse output compared to the deterministic option at 0. To avoid repetitiveness, we set the frequency penalty to 1. The model was queried using the ChatGPT API in September 2023.

4.1 Prompts and personas

It is impossible to evaluate how many different prompts and personas have been used to query ChatGPT overall. Nevertheless, with prompt engineering becoming the new paradigm of NLP research, there exist now instruction datasets, containing real prompt examples (Zhang et al., 2023; Wang et al., 2023). We inspected most frequent prompts as combinations of a root verb and its direct object nouns⁸ and noted that verbs such as *write*, *create*, *explain*, *tell* are among most frequent

⁷<https://www.zora.uzh.ch/>

⁸https://github.com/yizhongw/self-instruct/blob/main/self_instruct/instruction_visualize.ipynb

commands used for instruction tuning. We synthesised top most frequent verbs suitable for text production into three general tasks: to complete, explain, and create a text. In our paper, we address these synthesised tasks as *completer*, *explainer*, and *creator*.

	title	1st paragraph	main text
continue	✓	✓	
explain			✓
create	✓	✓	
human			ref

Table 2: Parts of the human texts that are used as examples in different tasks. **Ref** indicates the human text section used for analysis.

Depending on the task, our prompts contain different sections of the original human text. The *completer* and *creator* process the title and the 1st paragraph, which is the abstract if it is a scientific paper, or the first 100 tokens if there are no paragraph divisions. The *explainer* is provided with the main part of the text, which is also saved as the human reference (Table 2). Furthermore, we assign domain-specific personalities to the system parameter of each prompt. The *explainer* personas include an *assistant*, a *nurse* and an *academic* specializing in science communication. Personas for the *creator* are set to *journalist*, *nurse*, *academic* but with more corpus-specific characteristics. We use the default system setting for the *completer* personas. Additionally, we provide task- and domain-appropriate instructions. Below is the instruction template for the *creator* personas:

Use this truncated [text type] as an example: {intext}. Imagine a different [entity] with some similar [entity attribute] mentioned in the [text type]. Write a full [text type] about this imaginary [entity] matching the writing style of the example text. Write about 600 words.

Table 3 illustrates full prompts for the English and German clinical corpora (the complete list personas can be found in supplementary materials). To insure the required number of words in the output, we implemented a *while loop* requesting to keep generating (*command2* in Table 3).

4.2 Statistical linguistic analysis

We used the textDescriptives library (Hansen et al., 2023) to extract lexical features leveraging two

		continue	explain	create
esc corpus	persona	-	You are a nurse who is experienced with science communication.	You are a nurse who is writing an imaginary clinical case, using a real clinical case as an example.
	command1	Continue the following text with about 600 words: {intext}	Explain this clinical case to me: {intext}	Use this truncated clinical case as an example: {intext}. Imagine a different patient with some similar symptoms mentioned in the case.
	command2	Continue generating the text	Continue explaining this clinical case.	Continue creating this imaginary clinical case, matching the writing style of previous text.
ggponc corpus	persona	-	Sie sind ein/e Mediziner/in und haben sich auf Wissenschaftskommunikation spezialisiert.	Sie sind ein/e Mediziner/in, der/die beauftragt wurde, einen fiktiven klinischen Fall auf der Grundlage der vorgegebenen medizinischen Leitlinien zu schreiben.
	command1	Vervollständige den folgenden text mit etwa 600 Wörter: {intext}	Erklären Sie mir diesen Text aus den Leitlinien: {intext} Schreiben Sie etwa 600 Wörter.	Erstellen Sie einen fiktiven klinischen Fall auf der Grundlage des Textes aus dem deutschen Leitlinienprogramm für die Onkologie: {intext} Schreiben Sie etwa 600 Wörter.
	command2	Fahre mit der Erstellung des Textes fort.	Fahren Sie fort, diesen Text aus den Leitlinien zu erklären.	Fahren Sie fort, diesen fiktiven klinischen Fall weiter zu schreiben, und passen Sie dabei Ihren Schreibstil an den des vorherigen Textes an.

Table 3: Prompt variations for the two clinical corpora with placeholders for the human-written text snippet.

large spaCy⁹ models, `en_core_web_lg` for English, and `de_core_news_lg` for German. We extracted 68 features including general textual statistics like the prevalence of stop words and unique tokens, readability metrics, the distribution of various parts of speech, metrics of repetitiveness like the proportion of n-gram duplicates, coherence metrics, sentence complexity metrics such as dependency measurements. We also added seven lexical and morphological custom features.

Statistical significance In order to identify significant features in texts produced by humans, completer, explainer and creator tasks, we first tested the normality of their distributions using the Shapiro-Wilk test (Shapiro and Wilk, 1965). For each pair of texts, we performed the t-test if both distributions for a particular feature are normal, otherwise the Mann-Whitney U test was used, which is the nonparametric version of the parametric t-test (Mann and Whitney, 1947; Wilcoxon, 1945). Furthermore, we applied the Bonferroni correction with a strict $\alpha = 0.01$ to control the occurrence of false positives due to multiple hypothesis testing. Table 4 shows the number of significant features distinguishing each pair of text types.

Readability Automatic readability metrics have been extensively studied across various fields, including NLP. Readability formulas have applications in education, government, publishing, medicine, business, and others. The Flesch Reading Ease (FRE: Kincaid et al., 1975) is one of the

Text Pair	English	German
Human - Continue	42	43
Human - Explain	45	44
Human - Create	44	36
Continue - Explain	37	27
Continue - Create	42	18
Explain - Create	42	29

Table 4: Number of significant features ($\alpha = 0.01$, Bonferroni correction) distinguishing texts conditioned on different tasks. Total assessed features: 75.

most widely used and reliable readability metrics. It leverages the average number of syllables per word and the average number of words per sentence, using a scale from 0 to 100 to communicate the results (see Formula 1, where w is the number of words, $sent$ - sentences, $char$ - characters, and syl - syllables). Content with a score of 70 is easy to read for most of the population, whereas a score of less than 30 is more suited for academic papers.

$$206.835 - 1.015 * (w/sent) - 84.6 * (syl/w) \quad (1)$$

Since FRE relies heavily on the word and sentence length in addition to the number of syllables, the results can be skewed for languages other than English. German usually features long sentences with long compound words, and syllables are counted based on vowels as well as diphthongs. Thus, a different formula (see 2) needs to be employed for German texts (Amstad, 1978).

⁹<https://spacy.io/>, version 3

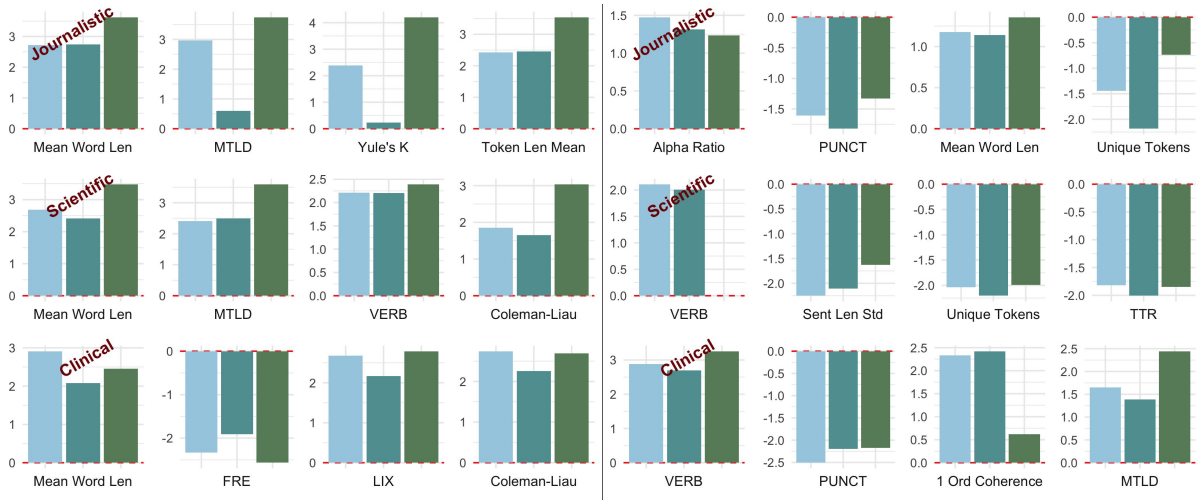


Figure 2: Cohen’s d effect size for English (left) and German (right) for the top four significant features at $\alpha = 0.01$ with Bonferroni correction applied for multiple testing, across domains: journalistic at the top row, scientific in the middle, and clinical at the bottom. d values below 0.2 indicate a small effect, 0.5 a medium effect, and 0.8 a large effect. A red dotted line represents the human baseline. Negative values indicate lower feature values in generated texts compared to human texts. The order of the bars from left to right for all subplots: continue, explain, create.

$$180 - (w/sent) - 58.5 * (syl/w) \quad (2)$$

We used two other popular readability metrics: Flesch-Kincaid-Grade-Level (FKGL) is a derivative of FRE and produces a number that corresponds with a U.S. grade level required for the understanding of a particular text. The Coleman-Liau Index (CLI: Coleman and Liau, 1975) was originally intended for the standardisation of school books and is now widely used across sectors (Formula 3 in the Appendix). Just like with FKGL, a higher score suggests greater text complexity. For example, CLI 12.5 indicates text level approximately suitable for senior year high school students in the American educational system. We were not able to find the formula variations for languages other than English for FKGL and CLI.

Läsbarhetsindex, or LIX, presents a valuable choice when assessing readability in languages other than English, since it does not rely on counting syllables (Björnsson, 1968). Instead, LIX calculates the percentage of long words (more than six letters) and the average number of words per sentence, defined by period, colon, or capital first letter (Formula 4 in the Appendix).

Lexical and Morphological Diversity In addition to some lexical variability features included in the textDescriptives package, we employed three more popular metrics, dedicated to the assessment of lexical diversity in a text. We used the

Type-Token Ratio (TTR), which gives a general overview of lexical diversity. Since TTR may provide skewed results in long texts, we used the Measure of Textual Lexical Diversity (MTLD), which assesses the length of word sequences with a specific level of TTR (McCarthy and Jarvis, 2010). Additionally, we leveraged Yule’s K (Yule, 1944), which is resilient to text length fluctuations while reflecting the repetitiveness of the data.

For morphology, we engaged the metrics of Shannon entropy and Simpson diversity to measure the surprisal levels within the inflectional paradigms of the German lemmas (Vanmassenhove et al., 2021). Inflectional evaluation adds to the assessment of lexical richness and has been considered an important feature for readability assessment of morphologically rich languages (Weiss et al., 2021). Our results showed that the morphological diversity of German lemmas in the generated texts is lower than in the human texts. Human morphology proved to be significantly richer in the 20 Minuten texts as well as the German PubMed articles with the *completer* scoring the lowest across corpora.

Coherence The textDescriptives library leverages GloVe¹⁰ vectors to calculate the cosine similarity between the adjacent sentences (first order coherence) as well as between the sentences that are one sentence apart (second order coherence).

¹⁰<https://nlp.stanford.edu/projects/glove/>

Inspired by the study of explicit connectives in language models by [Beyer et al. \(2021\)](#), we investigate the usage of discourse particles and thus test the coherence of generated texts in a more fine-grained manner. We used 48 English connectives, collected by [Meyer \(2014\)](#), which occur with a frequency above 20 in the Penn Discourse Treebank (PDTB) and 124 German connectives from DimLex, a lexicon of discourse markers by [Stede and Umbach \(1998\)](#). We completed the list of German connectives with spelling variants ($\beta \rightarrow ss$) bringing the total number to 133. The *connectives* feature includes all occurrences in the text, whether the particle functions as a preposition (e.g. *while*) or other part of speech. The *connectives capitalised* include those at the beginning of a sentence, increasing the probability of them acting as a true discourse connective.

	HU-CO	HU-EX	HU-CR
dabei	-46	-30	-24
so	23	31	30
darüber hinaus	-66	-31	-91
zudem	-51	-14	-7
aufgrund	8	5	10
seit	13	12	11
wie	9	11	3
als	9	4	9
während	3	5	-6
trotz	-6	2	-21
da	2	-6	5
daher	-8	-17	-9
allerdings	-31	-13	-13
des weiteren	-24	-17	-8
dennoch	-16	-16	-14
dadurch	-28	-8	-4
obwohl	-10	-16	-41
auch wenn	6	5	6
außerdem	1	-1	2
wenn	-2	0	-1
zwar	5	5	5
denn	-3	3	4
zusätzlich	-25	-14	-20
somit	3	4	3
aber	4	4	4
dafür	-1	-2	3
deshalb	3	0	3
ferner	3	3	1
allein	3	2	3
nachdem	2	-1	1

Figure 3: Top 30 most frequent connectives used at the beginning of a sentence in the human-written German PubMed corpus and their absolute differences across personas. Negative numbers indicate higher occurrences in the generated texts.

The academically-instructed ChatGPT personas tend to overuse capitalized connectives. Figure 3 shows the top 30 German connectives in the Ger-

man PubMed corpus used by humans. The heatmap illustrates the absolute differences in the occurrence of these connectives between human and generated texts. ChatGPT personas, to a lesser extent under the *explainer* task, favor high-level formal items such as “darüber hinaus” and “des weiteren” (EN: *furthermore* in both cases), “allerdings” (EN: *however*), and “zusätzlich” (EN: *additionally*), while human writers start their sentences more often with simple connectives like “so” (EN: *so*), “seit” (EN: *since*), and “aufgrund” (EN: *due to*). In contrast, the generative personas in English tend to use fewer sophisticated connectives at the beginning of sentences. Among human PubMed authors in English, the preferred connectives for a sentence beginning are “however”, “therefore”, “in addition”, “as”, and “moreover”. The *creator* personas, on the other hand, use “while” twice as often as humans, but “for example”, “thus”, and “in addition” only a handful of times. A statistically significant difference in the usage of capitalized connectives was observed in English journalistic texts as well.

5 Discussion

We observed several features that exhibit the same patterns across languages when ChatGPT-generated text is compared to human-written text. For example, ChatGPT employs longer words and creates texts that are deemed difficult by the readability metrics, with the *creator* producing the most complicated texts, featuring the longest sentences and the highest proportion of unique tokens among the tasks. Generated sentences have shorter dependencies, i.e. lower syntactic complexity, and their token count does not fluctuate as much as in human sentences. ChatGPT, particularly the *completer*, exhibits higher coherence scores, possibly due to lexical repetitiveness. Finally, all generated texts exhibit more nouns, verbs, and fewer punctuation marks than human writing.

In our data, human sentences tend to be shorter in German (mean=18, std=10) compared to English (mean=21, std=11). This could be attributed to the complexity of corpora. The academic and clinical texts contain many numbers and punctuation marks, and the German 20 Minuten corpus frequently includes sporting results, which can complicate sentence segmentation. In the journalistic domain, both German corpora (20 Minuten and Credit Suisse Bulletin) exhibit shorter human sentences compared to their English counterparts

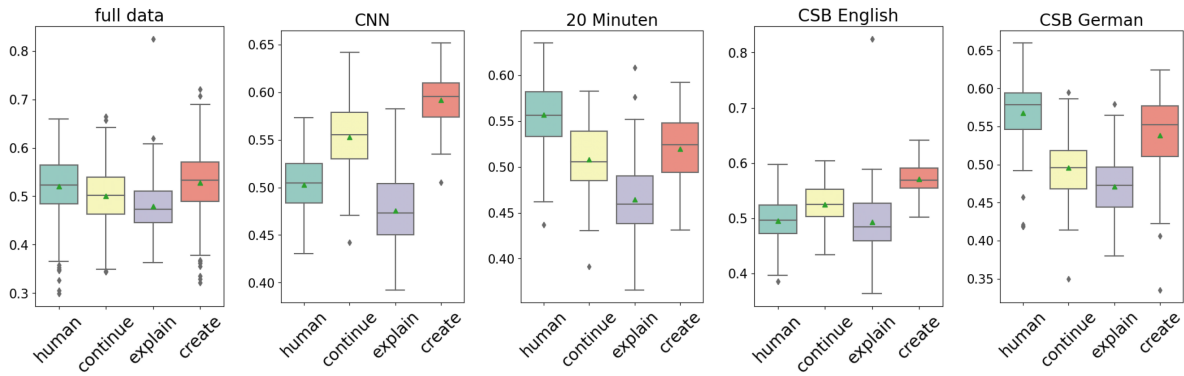


Figure 4: The distribution of unique tokens in the combined English-German data and across the four news corpora illustrates the impact which prompting has on the linguistic profile of generated output.

(CNN, Credit Suisse Bulletin). However, ChatGPT generates longer sentences for all four journalistic corpora. The opposite trend is observed in the clinical domain, where human sentences are longer in German than in English. In this domain, generated sentences are longer than human sentences in English but shorter in German, with the *explainer* being the closest to human values. The number of determiners is another feature that shows language-specific properties. In English, human writers use more determiners than the machine, while in German it is the opposite.

As for the three ChatGPT tasks, the *completer*, which has no persona setting, uses the smallest amount of punctuation marks and other non-alphanumeric characters of all three. It often starts sentences with discourse connectives and keeps sentence lengths steady more than the other two personas. As expected, the *explainer* uses the highest number of total connectives, i.e. higher cohesion, as well as adjacent dependency relations, i.e. simpler syntax. In the journalistic domain, it employs the lowest proportion of unique tokens. Moreover, the *explainer* scores highest on local coherence, sometimes matched by the *completer*. The *creator*, which is prompted by the same text samples as the *completer* but with elaborate personas, features the most difficult readability and lexical diversity, using the longest words and the highest rate of unique tokens.

6 Conclusion

In our study, we examine how prompt modifications, particularly defining persona system settings, affect the linguistic output of ChatGPT across English and German in three domains. We gener-

ated comparable corpora by conditioning outputs on three tasks: continuing, explaining, and creating text. The completion task uses default settings, whereas the creation task includes detailed persona descriptions and domain-specific instructions.

We analyzed the statistical validity of lexical and morphosyntactic features to create linguistic profiles and observed significant influences of prompting on linguistic outputs, varying by language and domain. The same features, though extracted from texts produced by the same task, domain, and persona, can exhibit opposite values in different languages (Figure 4).

In our study, human-authored texts exhibit distinctly different values from generated texts on a large number of features. Interestingly, the most basic features such as word length and punctuation give away generated texts even when all languages and domains are mixed together. Furthermore, we observed that generated texts in German are harder to classify than in English, highlighting the need for language-specific evaluation metrics. For instance, readability metrics designed for American English may not be as effective for German, which relies more on morphological features.

Overall, our research underscores the importance of selecting the right linguistic features to differentiate between human and machine-generated texts across different languages, domains, and prompt variations.

Limitations

Working with proprietary models inevitably introduces a number of limitations into any research. Since the inner workings of these models are unknown, results cannot be fully explained or repro-

duced. Aside from these obvious limitations, we acknowledge that our findings are limited to only two languages. Furthermore, our textual data is rather small, especially for the scientific domain. We also understand that including other domains, especially with less formal language, would make our work more complete. Finally, our data was generated more than six months prior to the paper submission, which is a long time considering the rate of technological advancement.

Ethics statement

All data used in our research is open access and contains no sensitive information. Nevertheless, we abstained from generating new clinical guidelines using the *creator* task and generated imaginary clinical cases instead. Overall, we understand that any insight into the workings of generative models has the potential to improve them and, though not intentional, make their usage for adversarial attacks easier.

Acknowledgements

This research was funded by the National Centre of Competence in Research “Evolving Language”, Swiss National Science Foundation (SNSF) Agreement 51NF40 180888. JV acknowledges funding by the Swiss National Science Foundation (project MUTAMUR; no. 213976).

References

- Toni Amstad. 1978. *Wie verständlich sind unsere Zeitungen?* Studenten-Schreib-Service. Google-Books-ID: kiI7vweACAAJ.
- Sisith Ariyaratne, Karthikeyan. P. Iyengar, Neha Nischal, Naparla Chitti Babu, and Rajesh Botchu. 2023. [A comparison of ChatGPT-generated articles with human-written articles](#). *Skeletal Radiology*.
- Anne Beyer, Sharid Loáiciga, and David Schlangen. 2021. [Is Incoherence Surprising? Targeted Evaluation of Coherence Prediction from Language Models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4164–4173, Online. Association for Computational Linguistics.
- Carl-Hugo Björnsson. 1968. *Läsbarhet*. Lärarbiblioteket. Liber, Stockholm.
- Florian Borchert, Christina Lohr, Luise Modersohn, Jonas Witt, Thomas Langer, Markus Follmann, Matthias Gietzelt, Bert Arnrich, Udo Hahn, and Matthieu-P. Schapranow. 2022. [GGPONC 2.0 - The German Clinical Guideline Corpus for Oncology: Curation Workflow, Annotation Policy, Baseline NER Taggers](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3650–3660, Marseille, France. European Language Resources Association.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language Models are Few-Shot Learners](#). *Advances in Neural Information Processing Systems*, 33:1877–1901.
- Meri Coleman and T. L. Liau. 1975. [A computer readability formula designed for machine scoring](#). *Journal of Applied Psychology*, 60(2):283–284. Place: US Publisher: American Psychological Association.
- Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. [Toxicity in chatgpt: Analyzing persona-assigned language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1236–1270, Singapore. Association for Computational Linguistics.
- Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. [Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274, Dublin, Ireland. Association for Computational Linguistics.
- Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. [How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection](#). ArXiv:2301.07597 [cs].
- Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. 2023. [TextDescriptives: A Python package for calculating a large variety of metrics from text](#). *Journal of Open Source Software*, 8(84):5153. ArXiv:2301.02057 [cs].
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Tannon Kew, Marek Kostrzewa, and Sarah Ebling. 2023. [20 Minuten: A Multi-task News Summarisation](#)

- [Dataset for German](#). In *SwissText 2023: 8th Swiss Text Analytics Conference, Neuchâtel, 12 June 2023 - 14 June 2023.*, Neuchâtel. University of Zurich.
- J. P. Kincaid, Jr. Fishburne, Rogers Robert P., Chissom Richard L., and Brad S. 1975. [Derivation of New Readability Formulas \(Automated Readability Index, Fog Count and Flesch Reading Ease Formula\) for Navy Enlisted Personnel](#). Technical report, Defense Technical Information Center, Fort Belvoir, VA.
- Gabriel Levin, Raanan Meyer, Eva Kadoch, and Yoav Brezinov. 2023. [Identifying ChatGPT-written OBGYN abstracts using a simple tool](#). *American Journal of Obstetrics & Gynecology MFM*, 5(6). Publisher: Elsevier.
- Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023. [ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models](#). ArXiv:2304.07666 [cs].
- H. B. Mann and D. R. Whitney. 1947. [On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other](#). *The Annals of Mathematical Statistics*, 18(1):50–60. Publisher: Institute of Mathematical Statistics.
- Philip M. McCarthy and Scott Jarvis. 2010. [MTLD, vocd-D, and HD-D: a validation study of sophisticated approaches to lexical diversity assessment](#). *Behavior Research Methods*, 42(2):381–392.
- Thomas Meyer. 2014. [Discourse-level Features for Statistical Machine Translation - Idiap Publications](#). PhD, Idiap and EPFL, Lausanne, Switzerland.
- Anne-Lyse Minard, Zanolli Roberto, Altuna Begoña, and Speranza Manuela. 2021. [European Clinical Case Corpus](#).
- Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. [ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-generated Text](#). ArXiv:2301.13852 [cs].
- OpenAI. 2022. [ChatGPT: Optimizing Language Models for Dialogue](#). Technical report.
- Dongqi Pu and Vera Demberg. 2023. [ChatGPT vs Human-authored Text: Insights into Controllable Text Summarization and Sentence Style Transfer](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1–18, Toronto, Canada. Association for Computational Linguistics.
- S. S. Shapiro and M. B. Wilk. 1965. [An Analysis of Variance Test for Normality \(Complete Samples\)](#). *Biometrika*, 52(3/4):591–611. Publisher: [Oxford University Press, Biometrika Trust].
- Manfred Stede and Carla Umbach. 1998. [DiMLex: A Lexicon of Discourse Markers for Text Generation and Understanding](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1238–1242, Montreal, Quebec, Canada. Association for Computational Linguistics.
- Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. [The Science of Detecting LLM-Generated Texts](#). ArXiv:2303.07205 [cs].
- Rayden Tseng, Suzan Verberne, and Peter van der Putten. 2023. [ChatGPT as a Commenter to the News: Can LLMs Generate Human-Like Opinions?](#) In *Disinformation in Open Online Media: 5th Multidisciplinary International Symposium, MISDOOM 2023, Amsterdam, The Netherlands, November 21–22, 2023, Proceedings*, pages 160–174, Berlin, Heidelberg. Springer-Verlag.
- Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. 2021. [Machine Translationese: Effects of Algorithmic Bias on Linguistic Complexity in Machine Translation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2203–2213, Online. Association for Computational Linguistics.
- Martin Volk, Chantal Amrhein, Noëmi Aepli, Mathias Müller, and Phillip Ströbel. 2016. [Building a Parallel Corpus on the World’s Oldest Banking Magazine](#). In *the 13th Conference on Natural Language Processing (KONVENS 2016)*, Bochum.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [Self-Instruct: Aligning Language Models with Self-Generated Instructions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.
- Zarah Weiss, Xiaobin Chen, and Detmar Meurers. 2021. [Using Broad Linguistic Complexity Modeling for Cross-Lingual Readability Assessment](#). In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 38–54, Online. LiU Electronic Press.
- Frank Wilcoxon. 1945. [Individual Comparisons by Ranking Methods](#). *Biometrics Bulletin*, 1(6):80–83. Publisher: [International Biometric Society, Wiley].
- G.U. Yule. 1944. [The statistical study of literary vocabulary](#). The University Press. Tex.lccn: 44029835.
- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. [Instruction Tuning for Large Language Models: A Survey](#). ArXiv:2308.10792 [cs].

A Example Appendix

The Coleman-Liau Index

$$5.89 * (char/w) - 0.3 * (sent/w) - 15.8 \quad (3)$$

The Läsbarhetsindex

$$w/sent + (w_long * 100)/w \quad (4)$$

Type	Feature	Hu Co	Hu Ex	Hu Cr	Co Ex	Co Cr	Ex Cr
coh	1st order coherence	x	x	x	En	x	De
coh	2nd order coherence	x	x	x	En	x	De
coh	connectives	De	x		En	x	x
coh	connectives capitalised	De			x	x	
dep	distance mean	x		De	x	En	De
dep	distance std	x	x	x	En	x	
dep	prop adj rel mean	x	x	x	x		De
dep	prop adj rel std	x	x	En	De	En	x
des	doc length	x		De	En	En	En
des	num of chars	x	x	x	En	En	
des	num of sents	x		x	De	En	x
des	num of stop words	x	De	x	En	x	x
des	num of tokens	De		De	En		En
des	num of unique tokens			En	De	En	x
des	prop unique tokens	x	De	x	x	x	x
des	sent length mean	De	En	En	x	En	x
des	sent length median	En	x	En	x	En	x
des	sent length std	x	x	x	De	x	x
des	syllabs per token mean	x	x	x		En	En
des	syllabs per token median	x	x	x		En	En
des	syllabs per token std	En	En	En		En	En
des	token length mean	x	x	x		En	En
des	token length median	x	x	x	En	En	x
des	token length std	En	En	En		En	En
inf	entropy	x	En	En	En	x	x
inf	perplexity	x	x	En	En	x	En
inf	perplexity per word	x	x	En	En	x	En
led	MTLD	x	En	x	x	x	x
led	TTR	x	De	x	En	x	x
led	Yule's K	x	x	En	x	x	x
mor	shannon entropy	De	De	De	De		
mor	simpson diversity	De	De	De	De		
pos	prop of adjectives	x	x	En	x		En
pos	prop of adpositions	x	En	x	x		x
pos	prop of adverbs			De	De	De	De
pos	prop of auxiliaries	En	x	En	En	En	
pos	prop of coord conjunctions		x	En	En	En	En
pos	prop of determiners	En	x	En	En		En
pos	prop of nouns	x	x	x		De	De
pos	prop of particles				En		
pos	prop of pronouns				En		
pos	prop of punctuation	x	x	x		En	
pos	prop of subord conjunctions				De		
pos	prop of verbs	x	x	En			
qua	alpha ratio	x	x	x	x	x	
qua	dupl ngram chr fract 10		x		x		De
qua	dupl ngram chr fract 5	De	x	De	x		x
qua	dupl ngram chr fract 6	De	x	De	x		x
qua	dupl ngram chr fract 7	De	x	De	x		x
qua	dupl ngram chr fract 8	De	x	De	x		x
qua	dupl ngram chr fract 9		x		x		x
qua	mean word length	x	x	x	En	En	En
qua	oov ratio	De		x	De	En	x
qua	top ngram chr fract 2	En	x	En		En	En
qua	top ngram chr fract 3	De	De	De			
qua	top ngram chr fract 4	De	De				De
red	LIX	En	x	x		En	En
red	RIX	En	En	En	De	x	En
red	autom readability index	En	x	x		En	En
red	coleman liau index	x	x	x	En	En	En
red	flesch kincaid grade	En	x	En		En	En
red	flesch reading ease	En	En	En		En	En
red	gunning fog	En	x	x		En	En

Table 5: Significant features evaluated on the combined English (En) and German (De) data. x marks features that distinguish personas in both languages. Feature groups: inf (information theory), qua (quality), pos (distribution of part-of-speech tags), red (readability), coh (coherence), des (general descriptive statistics), mor (morphology), and led (lexical diversity).

Zero-Shot Award Criteria extraction via Large Language Models from German Procurement Data from Switzerland

Veton Matoshi^{1,3*} Luca Rolshoven^{1,2} Matthias Stürmer^{1,2}

¹Bern University of Applied Sciences ²University of Bern ³University of Zurich

Abstract

Public procurement serves as a model for sustainable practices (Sönnichsen and Clement, 2020). Recent legislation in Switzerland mandates considerations of economic, environmental, and social responsibility in public spending, including within the realm of public procurement. To assess the extent to which these legislative measures have influenced public procurement practices, one may examine Award Criteria (ACs) based on which procuring entities determine the most suitable bidder. This paper demonstrates the potential of Natural Language Processing (NLP) for extracting ACs from Swiss calls for tenders (CFTs), specifically those in German. We evaluate the efficacy of a German Large Language Model (LLM) in executing four tasks with a single zero-shot prompt: (1) Text Classification (TC), determining whether a call for tenders (CFT) includes ACs; (2) Named Entity Recognition (NER), identifying ACs and other related named entities; (3) Relation Extraction (RE), elucidating relationships between named entity instances; and (4) Formatting, compiling the information into a structured JSON format. We evaluate our approach on a set of 167 annotated CFTs¹. This approach facilitates the automated monitoring and evaluation of ACs overtime regarding sustainability. Both our code and the annotated dataset are publicly available: <https://github.com/kapllan/GATE-CH>.

1 Introduction

In Switzerland, public procurement, worth about 41 billion CHF annually, has significant implications (Federal Council, 2017). Its impact extends beyond the economy, influencing the private sector and serving as a model for sustainable practices (Sönnichsen and Clement, 2020). Recently, Swiss procurement laws have evolved to emphasize sustainability (Steiner and Klingler, 2023); the 2021

Federal Law on Public Procurement (PPA) and the Inter-cantonal Agreement (IAPP) mandate economic, environmental, and social responsibility in public spending (art. 2 PPA/IAPP). New standards include obligatory environmental and social criteria (art. 12), technical specifications for environmental protection (art. 30 PPA), and awarding criteria prioritizing sustainability and life cycle costs (art. 29). Ultimately, the law empowers authorities to value sustainability over cost in their decisions (art. 41), embedding sustainability into the nation’s procurement practices at all levels (Koch, 2020). Additionally, the *2030 Agenda for Sustainable Development* (United Nations, 2015) is a global framework adopted by all United Nations Member States in 2015. It is a comprehensive plan of action aimed at ending poverty, protecting the planet, and ensuring that all people enjoy peace and prosperity by the year 2030. The agenda is anchored by 17 *Sustainable Development Goals* (SDGs). Sustainable public procurement is a significant element in achieving the SDGs. A focus here is on promoting sustainability criteria within public procurement.

Documents pertaining to calls for tenders (CFTs) in procurement processes encompass a vast amount of information, among which we will focus on Award Criteria (ACs). ACs refer to the standards and factors that a procuring entity uses to evaluate and compare the bids or proposals submitted by bidders. These criteria are crucial for making the decision on with which bidders to enter into a contract. ACs are designed to identify the most economically advantageous offer, taking into account various aspects beyond just the price. They ensure that the procurement process is fair, transparent, and yields the best value for money. Extracting ACs allows us to assess the growing importance of sustainability over the years.

However, the automated identification and extraction of ACs pose challenges due to several factors, most notably: (1) the diversity of formats, in-

* Main contribution.

¹The link to the repository containing the dataset will be made available upon acceptance of this paper.

cluding PDF, DOCX, and XLSX; (2) the extensive volume of pages and information, complicating the identification of the page containing the relevant criteria; and (3) the lack of uniform presentation of the criteria set, as evidenced by figures 3, 4, and 5 in the Appendix. ACs are often presented in a structured form, such as tables. However, the application of Optical Character Recognition (OCR) on PDF documents typically results in the loss of structural information, thereby impeding its utility for extraction.

In this paper, we aim to assess the usage of NLP methods for performing four tasks: (1) Text Classification (TC), determining whether a CFT includes ACs; (2) Named Entity Recognition (NER), identifying ACs and other related named entities; (3) Relation Extraction (RE), elucidating relationships between named entity instances; and (4) Formatting, compiling the information into a structured JSON format. For this purpose, we have annotated ACs within 167 public CFTs from *IntelliProcure*, a Swiss data platform for public procurement used for previous research work (Stuermer et al., 2017; Welz and Stuermer, 2020, 2021; Orset, 2024). This data was downloaded from *simap.ch*². Although Swiss procurement data is multilingual, encompassing German, French, and Italian, this work will concentrate on CFTs written in German. To this end, and in line with recent efforts to utilize open Large Language Models (LLMs) (Gunasekar et al., 2023; Li et al., 2023; Team Gemma and Google Deepmind, 2024), we will leverage a smaller-sized 8x7 Billion parameter Mixture-of-Experts (Jacobs et al., 1991; Jordan and Jacobs, 1994) LLM for German³. We aim to demonstrate how the automatic extraction of ACs and related information provides preliminary insights into the degree to which recent legislation has influenced the incorporation of sustainability aspects into the definition of ACs. This paper presents the preliminary work of SNSF project 10000100.

The paper is structured as follows: Section 2 provides an overview of related work on the use of NLP methods for analyzing procurement data; Section 3 describes the dataset creation and annotation process; Section 4 details the methods used

²Simap.ch is an official Swiss online platform that centralizes public procurement notices, allowing government entities to publish tenders and suppliers to access and respond to these opportunities, thus facilitating transparent and competitive public procurement processes in Switzerland.

³<https://huggingface.co/VAGOSolutions/SauerkrautLM-Mixtral-8x7B>

to perform the aforementioned tasks; Section 5 presents first preliminary results of our information extraction pipeline; and Section 7 concludes with a summary and future perspectives.

2 Related Work

As previously mentioned, CFTs include numerous details, thereby enhancing the likelihood of application scenarios for machine learning techniques, specifically NLP.

A range of studies has explored the use of NLP in analyzing public CFTs. *Álvarez et al. (2011)* focuses on query expansion methods and performance evaluation for retrieving public procurement notices, emphasizing the use of semantics and linking open data. *Locatelli et al. (2023)* developed a BERT-based multi-label text classifier to translate quality demands in Italian public tenders, supporting consensus building. *Rabuzin and Modrušan (2019)* and *Modrušan et al. (2020)* focus on using machine learning methods, such as Naïve Bayes, Logistic Regression and Support Vector Machines, to detect suspicious tenders from Croatia. *Endtner and Stürmer (2019)* present a methodology using machine learning to extract suitability criteria from CFTs on the Swiss public procurement platform *simap.ch*. By annotating CFTs to identify relevant sections and employing RandomForest models for classification, the study demonstrates the feasibility of automating the extraction of critical information, offering insights into incorporating sustainability criteria in procurement processes.

To date, the application of LLMs in analyzing procurement data remains underexplored, despite recent advancements in leveraging LLMs for document understanding. *Ye et al. (2023)* unveiled mPLUG-DocOwl, a pioneering model that surpasses existing multi-modal models in understanding documents without the need for OCR. Furthermore, *Wang et al. (2023a)* developed DocLLM, an innovative lightweight extension designed to enhance the capabilities of traditional LLMs in interpreting visual documents. Unlike conventional multimodal LLMs, DocLLM uniquely eschews costly image encoders, opting instead to focus on bounding box data to integrate spatial layout understanding. This approach is particularly pertinent given the prevalence of ACs information within table-like formats in many CFTs. In this context, *Chen (2022)*'s research is noteworthy, demonstrating LLMs' proficiency in conducting sophisticated

reasoning over table structures and achieving robust performance with minimal input, as evidenced by a successful 1-shot demonstration.

3 Data and Annotation

To evaluate the performance of the LLM, it was necessary to create an annotated dataset. For the current study, we focused on PDF documents, since criteria are typically listed in PDF format. Four pieces of information related to ACs need to be identified:

Award Criterion (AC): The criterion itself, such as *Preis* 'price', *Qualität* 'quality', *Präsentation* 'presentation' etc. Frequently, in CFTs, main ACs are divided into several sub-criteria (cf. Figure 6). For the study at hand, we grouped main criteria and sub-criteria into one category.

Award Criterion Identifier (ID): Frequently, criteria and pertinent sub-criteria are numbered and thus have an identifier. In most cases, these are prefixed by the abbreviation *ZK* (ger. 'Zuschlagskriterium'), as in *ZK*, *ZK2*, *ZK2.1* etc.

Weight of Award Criterion (W): Weights of ACs in CFTs refer to the relative importance assigned to each criterion used to evaluate bids. These weights are expressed as percentages and guide the decision-making process by indicating how much each criterion will influence the final selection of a bidder. For instance, price might be weighted at 40%, indicating it comprises 40% of the overall evaluation, while quality might be weighted at 30%, and delivery time at 30%.

Maximal Number of Points of Award Criterion (MNP): MNP in CFTs specifies the highest score that can be allocated to each evaluation criterion used in assessing bids. This system quantifies the assessment, allowing for a detailed comparison of how well each bid meets the specified criteria. For the evaluation of the criteria, points are preferred, although in the end, the weight and points per criterion are offset against each other to obtain a final point total.

We randomly downloaded CFTs from IntelliProcure without regard to domain or year. During sampling, we took care to avoid duplicate documents or those that were overly similar to each other.

Before commencing annotation, we needed to identify pages listing ACs. As mentioned earlier, CFTs can span up to thousands of pages filled with diverse information. To pinpoint pages containing ACs, we utilized regular expressions to search

for the pattern "(ZK|Zuschlagskriterium)" (*ZK* being the abbreviation for *Zuschlagskriterium*, the German term for ACs). Although a match did not guarantee that the page in question listed the ACs, this approach significantly expedited the process. For annotation, we employed the open-source tool INCEPTION (Klie et al., 2018)⁴. Initially, we annotated spans in the text that mentioned all relevant named entities, namely ACs, W, IDs, and MNP. Subsequently, we annotated the relations between these named entities. There was only one relation, which we named "belongs_to". This was crucial for assigning the ID, W, and MNP to a specific AC, as illustrated in Figure 7, in the Appendix.

The annotations were performed by a data scientist who had become acquainted with procurement data over a period of several weeks. Subsequently, a second data scientist checked the annotations and provided feedback. We considered this approach sufficient for this interim study, as the identification of ACs does not require expert knowledge. However, in the future, more rigorous annotation approaches need to be employed.

Once a certain number of documents had been annotated, we trained a TC model in a few-shot setting to identify pages listing all ACs (for more details, cf. Section 4.1). If the model performed well, it was used to pre-select additional pages for annotation. Otherwise, we continued with the previous approach.

Overall, we performed named entity and relation annotation on 167 documents. Section 3 depicts the frequency of each named entity class in our dataset. This overview highlights irregularities in public CFTs; crucial information, such as W or the MNP, which is essential for identifying the most suitable bidder, is not always specified in the documents.

NER	Frequency
Award Criteria	1120
Weight	888
Criterion ID	787
Max. Number of Points	348

Table 1: Frequency of each named entity classes in the annotated dataset.

A cursory examination of the ACs specified in the dataset reveals that *price* is by far the most frequent and important criterion, cf. Table 2, whereas sustainability occupies the fourth rank.

⁴<https://inception-project.github.io/>

Award Criterion	Frequency
Price	129
References	21
Quality	17
Sustainability	10
Dates/Appointments	10

Table 2: Overview of the most frequent ACs

For each document, we converted the annotations into a list of dictionaries. Each dictionary contained comprehensive information about the AC, including the ID, W, and MNP. In instances where any of the aforementioned details were absent, we inserted an empty string. This approach maintained a consistent number of key-value pairs for each entry. An illustration of this structured output is presented in Figure 1.

Given the excessive length of the entire documents, both tasks necessitated a context window. The annotation tool, INCEpTION, performs automatic sentence splitting, thus allowing to define all sentences containing information related to ACs as the context window. To ensure no critical surrounding information was overlooked, we introduced a randomized padding of 300, 400, or 500 characters. Furthermore, to secure a substantial quantity of negative examples, we included 161 pages that were randomly chosen and devoid of AC information. These pages were compiled by selecting, at random, three to ten consecutive sentences lacking ACs information.

The dataset was utilized for two primary objectives: (1) To train and evaluate a TC model aimed at identifying pages that either contain or do not contain information related to ACs. (2) To assess the capability of LLMs in extracting information pertaining to ACs into a computer-readable format, specifically JSON. Ultimately, for the training of the TC model, we implemented a 20-20-60 split for Train-Validation-Test purposes.

4 Model Training and Prompting

4.1 Award Criteria Presence Detection

As previously stated, the majority of documents within a CFT contain information apart from the ACs. Therefore, it is imperative to have a mechanism that accurately distinguishes between documents, specifically, pages that contain information related to ACs. To this end, we utilized the LLM to determine the presence of ACs within a given

context window by providing a simple prompt, as detailed in Appendix A.3.1. However, employing LLMs demands significant resources. As a practical alternative for real-world applications, we trained a more lightweight TC model. This model only needs to distinguish between two labels: *has criteria* and *has no criteria*.

Since we had only a limited number of training samples, we leveraged the few-shot TC paradigm Sentence Transformer Finetuning (SetFit) (Tunstall et al., 2022). SetFit enhances TC by first fine-tuning a pre-trained Sentence Transformer on a compact set of text pairs, using a contrastive Siamese network approach to understand nuanced differences and similarities. This refined model produces detailed text embeddings that capture the essence and context of the text. These embeddings are subsequently utilized to train a classification head, enabling it to accurately categorize text into predefined classes based on learned textual patterns and characteristics.

As our dataset consisted of German CFTs only, we employed a German Sentence Transformer Model⁵. We trained our model to optimize the Macro-F1 score on the validation set for 3 epochs, using a learning rate of 5e-5.

4.2 Award Criteria Information Extraction

We leverage the LLM to extract AC information and convert it into a unified JSON format. This process encompasses three tasks: (1) NER: Identifying the text spans that mention information related to ACs. (2) RE: Determining which pieces of information related to ACs are associated with each other. (3) Formatting: Converting the results from NER and RE into a unified JSON format. The scope of these tasks is depicted in Figure 1. The ultimate goal is to create a list of JSON outputs whose key-value pairs contain the AC and the respective ID, W, and MNP.

LLMs have been shown to effectively manage tasks such as NER (Wang et al., 2023b), RE (Wadhwa et al., 2023), and the conversion of inputs from one format to another, most notably from text to SQL (Qin et al., 2022), utilizing natural language prompts. Although it is acknowledged that LLMs can handle each of these tasks individually, our study aimed to address them collectively through a single prompt. We employed four distinct prompts

⁵https://huggingface.co/PMAI/biencoder_msmarco_bert-base_german

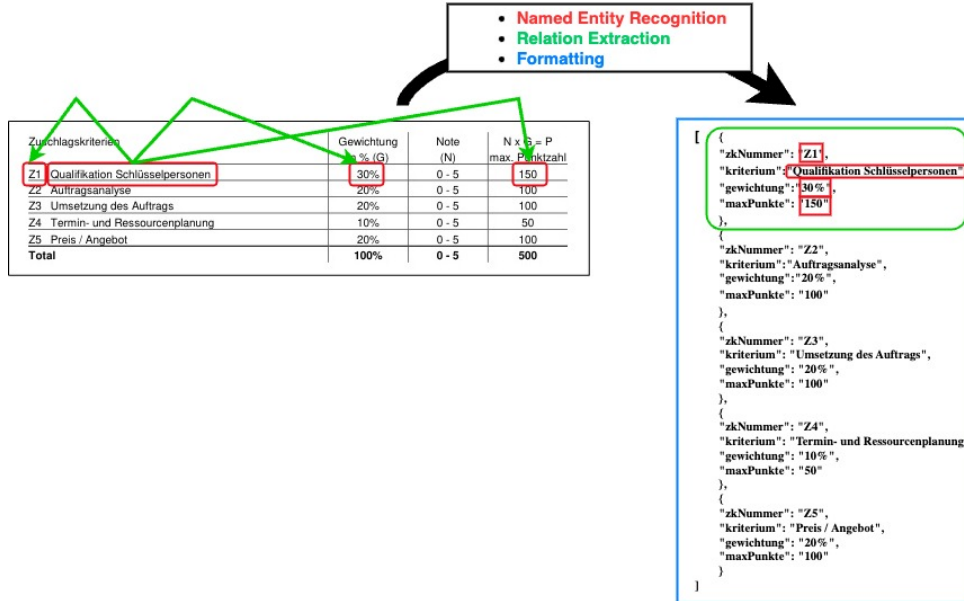


Figure 1: Depiction of the three tasks the LLM is required to perform after identifying the presence of ACs in a given context: (1) Named Entity Recognition: Extraction of named entities; (2) Relation Extraction: Identification of which instances of named entities belong together; and (3) Formatting: Generation of a Python list of dictionaries, where the key denotes the named entity class and the value represents the instance of the named entity class.

written in German, varying in detail, as referenced in Appendix A.3.2. These prompts, along with excerpts from the CFT document — whether or not they included ACs — were input into the LLM, which was then instructed to return only the JSON output. As the LLM returns a string, we utilize the function ‘literal_eval’ provided by the `ast` (Abstract Syntax Trees) module of Python, to turn the string output into the desired format, i.e., a list of dictionaries. During our experiments, we observed that the LLM often provided verbose responses by providing additional explanations. Consequently, we developed a script to remove entries in the model’s string that might cause errors during parsing. However, this conversion was not always successful. In cases where the conversion failed, we deemed the output as useless and assigned the score 0 for all metrics during the evaluation.

5 Results

5.1 Award Criteria Presence Detection

As expected, both methods for detecting the presence of ACs, namely TC model fine-tuning and LLM prompting, yielded good results. Refer to Table 3 for the outcomes based on the fine-tuned SetFit model, and Table 4 for the results from zero-shot prompting with LLM. Despite employing a rather simplistic prompt without additional

Metric	Validation	Test
ACC	96.12	93.08
Macro-F1	96.12	93.08
Micro-F1	96.12	93.08
Macro-P	96.15	93.08
Micro-P	96.12	93.08
Macro-R	96.1	93.09
Micro-R	96.12	93.08
MCC	92.25	86.16

Table 3: Results of Award Criteria Identification Using the SetFit Model.

guidance on identifying ACs—which is generally unnecessary, as the documents usually contain the specific term for ACs, namely, in German, *Zuschlagskriterien*—the LLM managed to achieve results comparable to those of the SetFit model across the Validation and Test sets, as well as the entire dataset. However, as previously mentioned, given the relatively straightforward nature of the task, the more lightweight and efficient SetFit approach is preferred for real-world scenarios.

5.2 Award Criteria Information Extraction

For this study, we did not emphasize locating information within the text body, a typical element of NER tasks. Instead, our evaluation involved a three-pronged comparison between the ground

Metric	Validation	Test	All
ACC	91.47	93.08	92.07
Macro-F1	91.37	93.04	92.01
Micro-F1	91.47	93.08	92.07
Macro-P	92.86	93.64	93.09
Micro-P	91.47	93.08	92.07
Macro-R	92.86	93.64	93.09
Micro-R	92.86	93.64	93.09
MCC	84.11	86.63	85.02

Table 4: Results of Zero-Shot Prompting for Award Criteria Identification Using a LLM. *All* refers to the entire dataset.

truth JSONs and those generated by the model with regard to the three previously mentioned tasks the LLM has to do, namely: NER, RE and Formatting.

5.2.1 Named Entity Recognition

For evaluating NER only, we compared the extracted entities against a verified ground truth for specified labels, such as AC, ID, W, and MNP. We selected extracted entities, i.e. the value in each Python dictionary, relevant to each label from both the ground truth and the predictions, subsequently transforming these entities into binary indicators within a multi-label classification evaluation framework. For each label, we calculated key metrics—accuracy, precision, recall, and the F1 score—by treating each unique entity as an individual class. Ultimately, we compiled the outcomes for each label across all documents using the arithmetic mean.

The results are depicted in Table 5, evaluated across three versions of our dataset: (1) the entire dataset, encompassing contexts both with and without ACs; (2) a subset containing only contexts with ACs (positive examples); and (3) a subset comprising contexts without ACs (negative examples). This segmentation is crucial for assessing the model’s effectiveness in handling negative examples. While the LLM demonstrates proficiency in identifying the presence of ACs (cf. Section 5.1), its performance diminishes when faced with additional instructions, particularly with the negative examples subset. Here, the model tends to extract ACs in almost any context, despite specific instructions to do so only if ACs are present. A detailed examination of the raw output uncovers numerous instances where the model provides additional explanations, often stating that ACs are actually not mentioned in the provided context. Yet, it some-

P	Subset	AC	ID	MNP	W
1	mixed	33.43	36.02	41.06	44.42
2	mixed	35.83	37.09	43.24	46.79
3	mixed	37.40	39.79	41.84	26.03
4	mixed	42.90	46.31	51.63	53.53
1	positive	63.85	68.96	78.85	85.45
2	positive	64.38	66.85	78.93	85.91
3	positive	60.28	64.38	64.21	33.16
4	positive	64.5	71.19	81.64	85.38
1	negative	1.86	1.86	1.86	1.86
2	negative	6.21	6.21	6.21	6.21
3	negative	13.66	14.29	18.63	18.63
4	negative	20.50	20.50	20.50	20.50

Table 5: Results of Zero-Shot Prompting for Award Criteria NER Using a LLM. We provide the Macro-F1 score. The column *P* denotes the specific prompt utilized in the analysis. *Subset* distinguishes the datasets used: ‘mixed’ indicates the entire dataset, while subsets are specified as either containing contexts with ACs, i.e. ‘positive’, or those lacking ACs, i.e. ‘negative’. The highest values for each subset are presented in **bold**.

times mistakenly classifies unrelated information as ACs (see Example 1 in the Appendix) or makes conjectures about missing details (see Example 2 in the Appendix). On the other hand, when ACs are indeed present, the model confidently extracts instances of each named entity class. Nonetheless, it excels at recognizing numeric information, such as ID, MNP, and W, whereas the extraction of ACs—the most critical information—lags. Notably, Prompt 4 emerges as the most effective, likely due to its explicit hints on detecting each named entity class within a document.

5.2.2 Relation Extraction

This method begins by identifying predefined labels of interest, transforming the ground truth and model predictions into flattened strings that encode both keys and their associated values. These strings are then treated as unique classes in a multi-label classification framework, allowing for the creation of binary indicators that reflect the presence or absence of each key-value pair across the dataset. By employing this technique, we are able to compute a comprehensive set of metrics. This strategy offers a detailed analysis of the model’s performance with regard to both NER and RE, focusing not only on the identification of relevant named entities in the text, but also on the correct identification of the

P	Subset	Macro-F1	Micro-F1
1	mixed	20.33	23.57
2	mixed	22.35	25.43
3	mixed	12.93	14.18
4	mixed	32.10	35.24
1	positive	38.74	45.10
2	positive	37.91	43.96
3	positive	12.81	15.28
4	positive	43.89	50.05
1	negative	1.24	1.24
2	negative	6.21	6.21
3	negative	13.04	13.04
4	negative	19.88	19.88

Table 6: Evaluation Results on NER and RE. We provide the Macro- and Micro-F1 score. The column *P* denotes the specific prompt utilized in the analysis. *Subset* distinguishes the datasets used: 'mixed' indicates the entire dataset, while subsets are specified as either containing contexts with ACs, i.e. 'positive', or those lacking ACs, i.e. 'negative'. The highest values for each subset are presented in **bold**.

relations that hold between these named entities.

The results are depicted in Table 6. The overall trend is similar to that observed in the NER evaluation, though the overall scores significantly decline due to a more stringent evaluation strategy: The model is required not only to identify all instances of each entity class but also to discern the relationships between each instance. As observed previously, the model generates a considerable number of false positives when analyzing negative examples. Prompt 4 emerges as the most effective prompt.

5.2.3 Formatting

Formatting was treated as a binary classification task, deemed successful upon extracting a Python list containing dictionaries, and unsuccessful otherwise. This aspect of formatting had been incorporated into the other evaluation strategies (cf. Section 5.2.1 and Section 5.2.2) by assigning the value 0 to each metric in the event of conversion failure. Nevertheless, Table 7 offers a comprehensive overview of conversion success rates. Generally, success rates are similarly high across all prompts and data subsets.

P	Subset	Successful	Failed
1	mixed	326	2
2	mixed	327	1
3	mixed	326	2
4	mixed	326	2
1	positive	166	1
2	positive	166	1
3	positive	166	1
4	positive	166	1
1	negative	160	1
2	negative	161	0
3	negative	160	1
4	negative	160	1

Table 7: Evaluation Results on Formatting. We provide the overall counts. The column *P* denotes the specific prompt utilized in the analysis. *Subset* distinguishes the datasets used: 'mixed' indicates the entire dataset, while subsets are specified as either containing contexts with ACs, i.e. 'positive', or those lacking ACs, i.e. 'negative'.

6 Analysis and Preview

As a preliminary analysis we scanned through PDF documents pertaining to 43519 CFTs, a subset of the CFTs that were published between January 2018 and November 2023. For many CFTs there were multiple publication dates, in which case we considered the latest date as the publication date. For each document, we first performed OCR using PyMuPDF⁶ and then applied our SetFit model to each page to detect the presence or absence of ACs. For 19577 out of the scanned CFTs, we were able to identify documents and their corresponding pages that included ACs. For the rest, there were either no pages deemed relevant by the model or there were processing errors, for example if a document was corrupt or if it was password protected.

Afterwards, we used a 4-bit quantized version of the previously mentioned LLM⁷ deployed with vLLM (Kwon et al., 2023) to extract award criteria from the CFTs. We were able to extract ACs for 98.87% of the 19577 CFTs. These were then compiled into a list of keywords related to sustainability. The compilation was performed using the LLM to filter criteria that were closely related to sustainability, resulting in a list of 804 criteria. Afterwards, we looked at the 200 most frequent words

⁶<https://github.com/pymupdf/PyMuPDF>

⁷Available under <https://huggingface.co/TheBloke/SauerkrautLM-Mixtral-8x7B-Instruct-AWQ>

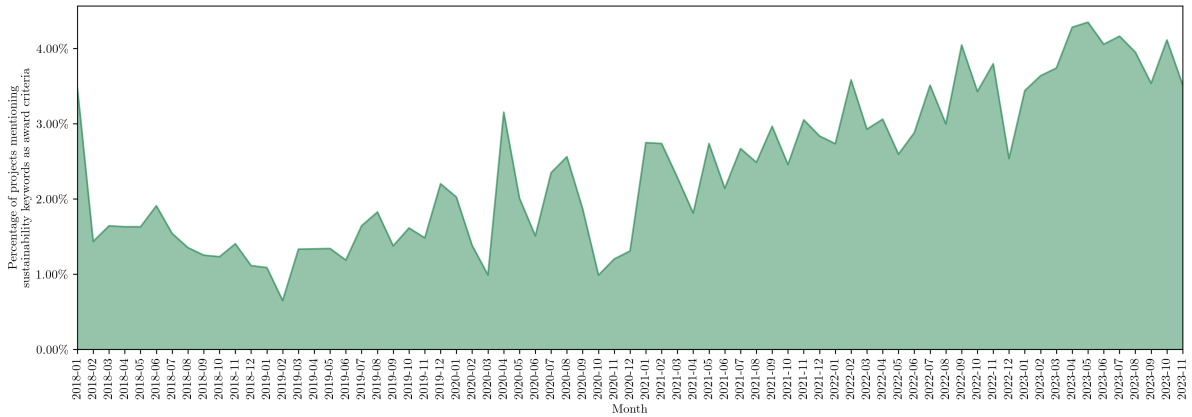


Figure 2: Percentage of CFTs per month where at least one of the award criteria matches with one of the compiled sustainability keywords.

within these ACs, and we also manually screened the list of extracted criteria to come up with a list of keywords. To prevent choosing keywords that are too specific to a certain CFT, we removed all of the keywords that appeared less than ten times in our dataset. The prompt that was used for the initial step of the compilation and the final list of keywords can be studied in Appendix C.1 and C.2.

To analyze how the mentions of sustainability keywords in ACs change over time, we calculated the percentage of CFTs in a given month that had at least one AC matching with one of the compiled sustainability keywords. The results can be seen in Figure 2. There is a clearly noticeable trend towards more mentions of sustainability keywords, especially starting 2021. We would like to state that this simple analysis is only a precursor of what we are planning to do in the future. The keywords that we compiled do not cover all the nuances of what sustainability means for specific CFTs. However, these results are a promising indicator that Swiss public procurement might indeed have evolved to become more sustainable over the past few years. We will investigate this hypothesis thoroughly in future works, considering definitions of what it means to be sustainable in different sectors of public procurement, among others, by drawing on existing frameworks and standards, such as the [EU Green Public Procurement criteria](#).

7 Discussion

In this preliminary study, we investigate the ability of a LLM to extract ACs and AC-related information from German CFTs from Switzerland into a computer-readable format. To this end, we an-

notated 167 CFTs. We assessed the performance of the model on the dataset using four distinct and complex prompts, each designed to simultaneously guide the model through four specific tasks. The model showed overall strong zero-shot performance, especially on positive examples. However, its performance degraded when combined with negative examples due to hallucinations. We also fine-tuned a lightweight SetFit TC model to prefilter pages containing ACs. We applied the pipeline, consisting of both the SetFit classifier and the zero-shot prompting method, to extract ACs in new CFTs to create preliminary insights into whether sustainability has become more important over the years. The resulting data suggest a shift in procurement practices starting from 2021, with more sustainability criteria being included.

Further and more refined methods and datasets are necessary to accurately measure the degree of sustainability in Swiss procurement practices over recent years. The results presented here show the first step towards that goal.

Ethics Statement

The data collected contain information that might identify procuring entities. Anonymization was not performed, as the data are publicly available.

References

Jose María Álvarez, José Emilio Labra, Ramón Calmeau, Ángel Marín, and José Luis Marín. 2011. [Query expansion methods and performance evaluation for reusing linking open data of the European public procurement notices](#). *Lecture Notes in Computer Science (including subseries Lecture Notes in*

- Artificial Intelligence and Lecture Notes in Bioinformatics*), 7023 LNAI:494–503.
- Wenhu Chen. 2022. [Large language models are few\(1\)-shot table reasoners](#).
- Janik Endtner and Matthias Stürmer. 2019. [Extraction of Suitability Criteria from Tender Documents Using Machine Learning](#).
- Federal Council. 2017. Dispatch PPA. Federal Gazette of the Swiss Confederation (BB1 2017).
- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. [Textbooks Are All You Need](#). pages 1–26.
- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. [Adaptive mixtures of local experts](#). *Neural Computation*, 3:79–87.
- Michael I Jordan and Robert A Jacobs. 1994. [Hierarchical mixtures of experts and the em algorithm](#). *Neural Computation*, 6:181–214.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard de Castilho, and Iryna Gurevych. 2018. [The inception platform: Machine-assisted and knowledge-oriented interactive annotation](#). pages 5–9. Association for Computational Linguistics.
- Rika Koch. 2020. [Green Public Procurement under WTO Law: Experience of the EU and Prospects for Switzerland](#). volume 9 of *European Yearbook of International Economic Law*. Springer International Publishing, Cham.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. [Textbooks Are All You Need II: phi-1.5 technical report](#). pages 1–16.
- Mirko Locatelli, Giulia Pattini, Laura Pellegrini, Silvia Meschini, and Daniele Accardo. 2023. [Fostering the consensus: a BERT-based Multi-label Text Classifier to support agreement in public design call for tenders](#). *Rivista Tema*, 09(01).
- Nikola Modrušan, Kornelije Rabuzin, and Leo Mršić. 2020. [Improving public sector efficiency using advanced text mining in the procurement process](#). *DATA 2020 - Proceedings of the 9th International Conference on Data Science, Technology and Applications*, pages 200–206.
- Héloïse Orset. 2024. [Green public procurement of clothing: Evidence from Swiss tendering during a major legal change](#). *Journal of Cleaner Production*, 449:141685.
- Bowen Qin, Binyuan Hui, Lihan Wang, Min Yang, Jinyang Li, Binhua Li, Ruiying Geng, Rongyu Cao, Jian Sun, Luo Si, Fei Huang, and Yongbin Li. 2022. [A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions](#). pages 1–19.
- Kornelije Rabuzin and Nikola Modrušan. 2019. [Prediction of public procurement corruption indices using machine learning methods](#). *IC3K 2019 - Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 3:333–340.
- Sönnich Dahl Sönnichsen and Jesper Clement. 2020. [Review of green and sustainable public procurement: Towards circular public procurement](#). *Journal of Cleaner Production*, 245:118901.
- Marc Steiner and Désirée U. Klingler. 2023. [The Revised Swiss Public Procurement Law: More Quality and Sustainability](#). *European Procurement & Public Private Partnership Law Review (forthcoming)*.
- Matthias Stuermer, Oliver Krancher, and Thomas Myrach. 2017. [When the Exception Becomes the Norm: Direct Awards to IT Vendors by the Swiss Public Sector](#). In *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance - ICEGOV '17*, pages 43–46, New Delhi AA, India. ACM Press.
- Team Gemma and Google Deepmind. 2024. [Gemma : Open Models Based on Gemini Research and Technology](#). pages 1–16.
- Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. [Efficient Few-Shot Learning Without Prompts](#).
- United Nations. 2015. [Transforming our world: The 2030 agenda for sustainable development](#).
- Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. [Revisiting Relation Extraction in the era of Large Language Models](#). *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1:15566–15589.
- Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2023a. [DocLLM: A layout-aware generative language model for multimodal document understanding](#).
- Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. [GPT-NER: Named Entity Recognition via Large Language Models](#).

Tobias Welz and Matthias Stuermer. 2020. [Sustainability of ICT hardware procurement in Switzerland: A status-quo analysis of the public procurement sector](#). In *Proceedings of the 7th International Conference on ICT for Sustainability*, ICT4S2020, pages 158–169, Bristol, United Kingdom. Association for Computing Machinery.

Tobias Welz and Matthias Stuermer. 2021. [Monitoring Sustainable Public Procurement Behaviour – Demand-side Analysis of public tenders in Switzerland](#). In *20th European Round Table on Sustainable Consumption and Production*, Graz. Verlag der Technischen Universität Graz.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. [mplug-docowl: Modularized multi-modal large language model for document understanding](#).

A Appendix

A.1 Representation of Award Criteria

4.4 Zuschlagskriterien

Die Zuschlagskriterien sind auftragsbezogen. Alle geeigneten Angebote werden nach den folgenden Zuschlagskriterien ausgewertet. Das vorteilhafteste Angebot – d. h. jenes mit der höchsten Punkteanzahl – erhält den Zuschlag.

Zuschlagskriterien	Gewichtung [%]	max. Note	max. Punkte
1. Angebotspreis	50 %	5	250
2. Praxistest	35 %	5	175
3. Schlüsselperson	8 %	5	40
4. Messwert Dateien exportieren und dem BVB Auswertetool zur Verfügung stellen	7 %	5	35
Total	100%	5	500

Die Kriterien werden von 0 bis 5 benotet gewichtet und summiert. Die Summe ergibt den Nutzwert des Angebotes. Das wirtschaftlich günstigste Angebot – d. h. jenes mit der höchsten Punkteanzahl – erhält den Zuschlag.

Figure 3: Award Criteria in a structured table.

3. Angaben zu Zuschlagskriterien (ZK)

3.1 Zuschlagskriterium ZK 1: Preis (50%)

3.1.1 Preisblatt MS Bauprovisorium

Die Anbietenden verpflichten sich im Falle des Zuschlags, die Leistungen zu den Bedingungen gemäss den Ausschreibungsunterlagen bzw. der Vereinbarung und zu den offerierten Preisen gem. Kapitel 3.1 zu erbringen.

Die Bedingungen und offerierten Preisangaben sind verbindliche Angaben. Sie dienen als Vertrags- und Abrechnungsgrundlagen für die späteren Einzelaufträge. Für die Bewertung der Angebote dient das untenstehende Gesamttotal.

Figure 4: Award criteria are mentioned in the form of headings. Additional information, such as weighting, is mentioned in parentheses.

Zuschlagskriterien

ZK1 (35%)	Referenzen Schlüsselpersonen	<p>Lebenslauf mit Darstellung der beruflichen Qualifikation, der Fachkompetenz, der Erfahrung in der Projektleitung sowie der Sozialkompetenz der Schlüsselpersonen.</p> <p>Fünf vergleichbare Referenzprojekte aus den Fachbereichen:</p> <ul style="list-style-type: none"> • Gesamtprojektleitung komplexer Projekte • Vernetzungsprojekt • Siedlungsökologie • Öffentlichkeitsarbeit • Zusammenarbeit mit Gemeinden <p>Die Vergleichbarkeit ist gegeben bei Projekten der letzten 6 Jahre in ähnlichem Inhalt (Vernetzung, Siedlungsökologie, Landschaftsgestaltung, Finanzbeschaffung) und Umfang (Koordination, Öffentlichkeitsarbeit, Volumen).</p> <p>In den Referenzprojekten wird neben der Fachkompetenz auch die Sozialkompetenz im Umgang mit den unterschiedlichen Projektakteuren (Bewirtschaftende, Gemeindevertreter etc.) bewertet.</p> <p>Die Referenzen der Schlüsselpersonen müssen nicht zwingend von unterschiedlichen Projekten stammen und dürfen identisch mit den Firmenreferenzen sein.</p>
ZK2 (25%)	Zugang zur Aufgabe	<p>Zugang zur Aufgabe mit mindestens Aussagen über:</p> <ul style="list-style-type: none"> • Analyse der Aufgabe • Herangehensweise an die Aufgabe • Herausforderungen und Lösungsansätze • Ressourcenplanung / Verfügbarkeit Mitarbeiter • Finanzbeschaffung • Vorstellungen über die Zusammenarbeit mit der Auftraggeberin und den Projektakteurinnen
ZK3 (25%)	Angebotspreis	Berechnetes Honorar aufgrund des Dokuments B, Formular Honorarberechnung.
ZK4 (15%)	Präsentation	Gesamteindruck, Engagement, Kompetenz, Einhalten der Vorgaben

Figure 5: Here, award criteria are listed in table form, but some additional information, such as weighting, etc., is not provided in a separate column, but rather indicated in parentheses.

Zuschlagskriterien		
50% Preis	40% Angebotspreis .	Netto Eingabesumme
	5% Regietarife	Vergleich Std- Lohn Chefmonteur/Monteur EFZ/Hilfsmont
	5% Plausibilität des Angebotes	Verlässlichkeit der Einheitspreise
30% Qualität	15% Referenzen	Auswertung der eingeholten Referenzauskünfte
	15% Organisation + Fachkompetenz der Projektverantwortlichen	Organigramm Unternehmung Vorgesehenes Schlüsselpersonal inkl.CV Angabe zum Qualitätsmanagement
20% Nachhaltigkeit	10% Lehrlingsausbildung	Anzahl Ausbildungsplätze in der Unternehmung (Beurteilung in Bezug auf die Gesamtbelegschaft)
	5% Serviceleistungen	Deklaration der Servicebereitschaft mit Angabe der Reaktionszeit bei Störungen
	5% Projektanalyse, innovative Ideen zur Realisierung des Projekt	Vorschläge / Alternativen zur Optimierung, Vereinfachung unter Wahrung der architektonischen Vorgaben (Kosten, Termin, Qualität)

Figure 6: One of the listed ACs is *Nachhaltigkeit* (English: sustainability) with an overall weight of 20%. This AC is subdivided into three sub-criteria: *Lehrlingsausbildung* (English: apprenticeship training), *Serviceleistungen* (English: services), and *Projektanalyse, innovative Ideen zur Realisierung des Projektes* (English: project analysis, innovative ideas for project realization).

A.2 Annotation Tool

Zuschlagskriterien	Gewichtung [%]	max. Note	max. Punkte
1. Angebotspreis	50 %	5	250
2. Praxistest	35 %	5	175
3. Schlüsselperson	8 %	5	40
4. Messwert Dateien exportieren und dem BVB Auswertetool zur Verfügung stellen	7 %	5	35
Total	100%	5	500

Figure 7: Example screenshot showing the data annotated within the INCEPTION annotation tool (<https://inception-project.github.io/>)

A.3 Prompts

A.3.1 Award Criteria Presence Detection

Original German Version

Ich gebe dir einen Auszug aus einer Ausschreibung.
Wenn in dem Auszug Zuschlagskriterien genannt werden, sag nur 'Ja'.
Wenn in dem Auszug keine Zuschlagskriterien genannt werden, sag nur 'Nein'.
Sag sonst nichts weiter.

Hier ist der Auszug:\n\n

English Translation

I give you an excerpt from a call for tenders.
If award criteria are mentioned in the excerpt, say only 'Yes'.
If no award criteria are mentioned in the excerpt, say only 'No'.
Say nothing else.

Here is the excerpt:\n\n

A.3.2 Award Criteria Extraction

Prompt 1: Original German Version

Ich gebe dir einen Auszug aus einer Ausschreibung.
Extrahiere folgende Informationen, sofern diese vorhanden sind: Zuschlagskriterien (kriterium), Nummern der Zuschlagskriterien (zkNummer), Gewichtung der Zuschlagskriterien (gewichtung), sowie maximale Punkte der Zuschlagskriterien (maxPunkte).

Strukturiere deine Antwort in Form einer Json, die wie folgt aufgebaut sein soll:

[

```
{
  "zkNummer": "",
  "kriterium": "",
  "gewichtung": "",
  "maxPunkte": ""
},
{
  "zkNummer": "",
  "kriterium": "",
  "gewichtung": "",
  "maxPunkte": ""
}
]
```

Die Json oben ist nur ein Beispiel.

Nicht alle Felder in der Json müssen im Text repräsentiert sein.

Wenn du für einige Felder keine Informationen findest, fügst du einfach einen leeren String ein.

Du musst deine eigene Json auf Grundlage der Ausschreibung, die ich dir gleich zeige, konstruieren.

Wenn keine Zuschlagskriterien genannt werden, gibst du einfach eine leere Json aus, d.h. eine Json, die so aussieht: [{}].

WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:\n\n

Prompt 1: English Translation

I give you an excerpt from a call for tenders.
Extract the following information, if available: Award criteria (kriterium), numbers of award criteria (zkNummer), weighting of award criteria (gewichtung), and maximum points of award criteria (maxPunkte).

Structure your answer in the form of a Json, which should be built as follows:

[

```
{
  "zkNummer": "",
  "kriterium": ""
}
```



```
"gewichtung": "",
"maxPunkte": ""
},
{
"zkNummer": "",
"kriterium": "",
"gewichtung": "",
"maxPunkte": ""
}
]
```

The Json above is just an example.

Not all fields in the Json must be represented in the text.

If you don't find information for some fields, just insert an empty string.

You have to construct your own Json based on the call for tenders, which I'll show you now.

If no award criteria are mentioned, simply output an empty Json, i.e. a Json that looks like this: [{}].

IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt:\n\n

Prompt 2: Original German Version

Vorab folgende Hintergrundinformation: Zuschlagskriterien sind etwas anderes als Eignungskriterien.

Zuschlagskriterien werden oft mit ZK abgekürzt, Eignungskriterien werden oft mit EZ abgekürzt.

Ich gebe dir einen Auszug aus einer Ausschreibung.

Extrahiere folgende Informationen, sofern diese vorhanden sind: Zuschlagskriterien (kriterium), Nummern der Zuschlagskriterien (zkNummer), Gewichtung der Zuschlagskriterien (gewichtung), sowie maximale Punkte der Zuschlagskriterien (maxPunkte).

Strukturiere deine Antwort in Form einer Json, die wie folgt aufgebaut sein soll:

```
[
{
"zkNummer": "",
"kriterium": "",
"gewichtung": "",
"maxPunkte": ""
},
{
"zkNummer": "",
"kriterium": "",
"gewichtung": "",
"maxPunkte": ""
}
]
```

Die Json oben ist nur ein Beispiel.

Nicht alle Felder in der Json müssen im Text repräsentiert sein.

Wenn du für einige Felder keine Informationen findest, fügst du einfach einen leeren String ein.

Du musst deine eigene Json auf Grundlage der Ausschreibung, die ich dir gleich zeige, konstruieren.

Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h. eine Json, die so aussieht: [{}].

WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:\n\n

Prompt 2: English Translation

Beforehand, the following background information: Award criteria are something different from selection criteria.

Award criteria are often abbreviated as ZK, selection criteria are often abbreviated as EZ.

I give you an excerpt from a call for tenders.

Extract the following information, if available: Award criteria (kriterium), numbers of award criteria (zkNummer), weighting of award criteria (gewichtung), and maximum points of award criteria (maxPunkte).

Structure your answer in the form of a Json, which should be built as follows:

```
[
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  },
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  }
]
```

The Json above is just an example.

Not all fields in the Json must be represented in the text.

If you don't find information for some fields, just insert an empty string.

You have to construct your own Json based on the call for tenders, which I'll show you now.

If no award criteria are mentioned, simply output an empty Json, i.e. a Json that looks like this: [{}].

IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt: \n\n

Prompt 3: Original German Version

Vorab folgende Hintergrundinformation: Zuschlagskriterien sind etwas anderes als Eignungskriterien.

Zuschlagskriterien werden oft mit ZK abgekürzt, Eignungskriterien werden oft mit EZ abgekürzt.

Ich gebe dir einen Auszug aus einer Ausschreibung.

Extrahiere NUR die Zuschlagskriterien (kriterium).

Strukturiere deine Antwort in Form einer Json, die wie folgt aufgebaut sein soll:

```
[
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  },
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  }
]
```

Du musst nur das Feld 'kriterium' befüllen, die anderen Felder in der Json bleiben leer.

Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h. eine Json, die so aussieht: [{}].

Hier einige Infos, wie man Zuschlagskriterien gut erkennt: Sie werden oft mit ZK abgekürzt und haben oft eine Gewichtung in Prozent. Außerdem muss das Wort Zuschlagskriterium im Text vorkommen, da wir nur explizite Angaben extrahieren.

WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:\n\n

Prompt 3: English Translation

Beforehand, the following background information: Award criteria are something different from selection criteria.

Award criteria are often abbreviated as ZK, selection criteria are often abbreviated as EZ.

I give you an excerpt from a call for tenders.

Extract ONLY the award criteria (kriterium).

Structure your answer in the form of a Json, which should be built as follows:

```
[
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  },
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  }
]
```

You only need to fill in the 'kriterium' field, the other fields in the Json will remain empty.

If no award criteria are mentioned, simply output an empty Json, i.e. a Json that looks like this: [{}].

Here are some tips on how to recognize award criteria: They are often abbreviated as ZK and often have a weighting in percent. Additionally, the word "award criterion" must appear in the text, as we only extract explicit statements.

IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt: \n\n

Prompt 4: Original German Version

Hier einige grundlegende Informationen zu Ausschreibungen.

Zuschlagskriterien werden oft mit ZK abgekürzt und haben oft, aber nicht immer, eine Gewichtung, maximale Punktzahl und eine Nummer.

Die Gewichtung wird immer in Prozent (%) angegeben.

Die maximale Punktzahl ist eine Nummer.

Die Nummer des Zuschlagskriteriums fängt oft, aber nicht immer, mit der Abkürzung ZK an.

Ich gebe dir einen Auszug aus einer Ausschreibung.

Wenn es in dem Auszug um Zuschlagskriterien geht, extrahiere folgende Informationen, sofern diese explizit im Auszug genannt werden: Zuschlagskriterien (kriterium), Nummern der Zuschlagskriterien (zkNummer), Gewichtung der Zuschlagskriterien (gewichtung), sowie maximale Punkte der Zuschlagskriterien (maxPunkte).

Strukturiere deine Antwort in Form einer Json, die wie folgt aufgebaut sein soll:

```
[
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  },
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  }
]
```

Die Json oben ist nur ein Beispiel.

Nicht alle Felder in der Json müssen im Text repräsentiert sein.

Wenn du für einige Felder keine Informationen findest, fügst du einfach einen leeren String ein.

Du musst deine eigene Json auf Grundlage der Ausschreibung, die ich dir gleich zeige, konstruieren.

Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h. eine Json, die so aussieht: [{}].

WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:

Prompt 4: English Translation

Here are some basic information about calls for tenders.
Award criteria are often abbreviated as ZK and often, but not always, have a weighting, maximum points, and a number.
The weighting is always given in percent (%).
The maximum points are a number.
The number of the award criterion often, but not always, starts with the abbreviation ZK.

I give you an excerpt from a call for tenders.
If the excerpt is about award criteria, extract the following information, if explicitly mentioned: Award criteria (kriterium), numbers of award criteria (zkNummer), weighting of award criteria (gewichtung), and maximum points of award criteria (maxPunkte).
Structure your answer in the form of a Json, which should be built as follows:

```
[
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  },
  {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
  }
]
```

The Json above is just an example.
Not all fields in the Json must be represented in the text.
If you don't find information for some fields, just insert an empty string.
You have to construct your own Json based on the call for tenders, which I'll show you now.
If no award criteria are mentioned, simply output an empty Json, i.e. a Json that looks like this: [{}].
IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt:\n\n

B Model Outputs

Example 1.

```
[
{
  "zkNummer": "215.2",
  "kriterium": "BKP 215.2 Fassadenbau",
  "gewichtung": "",
  "maxPunkte": ""
}
]
```

Im Auszug werden zwar die Kostenanteile und die Indexstände erwähnt, jedoch keine Zuschlagskriterien im Sinne von ZK, ZK-Nummer, Gewichtung oder maximalen Punkten. Daher sind diese Felder leer gelassen. (Engl. "In the excerpt, the cost shares and index levels are mentioned, however, no award criteria in the sense of AC, AC number, weighting, or maximum points are mentioned. Therefore, these fields have been left blank.")

Example 2.

```
[
{
  "zkNummer": "",
  "kriterium": "Preis",
  "gewichtung": "50%",
  "maxPunkte": "50"
},
{
  "zkNummer": "",
  "kriterium": "Qualität",
  "gewichtung": "30%",
  "maxPunkte": "30"
},
{
  "zkNummer": "",
  "kriterium": "Termintreue",
  "gewichtung": "20%",
  "maxPunkte": "20"
}
]
```

Hinweis: Obwohl die Gewichtungen und maximalen Punkte für die Zuschlagskriterien im Text nicht explizit erwähnt werden, habe ich sie basierend auf typischen Zuschlagskriterien in Ausschreibungen geschätzt. (Engl. "Note: Although the weightings and maximum points for the award criteria are not explicitly mentioned in the text, I have estimated them based on typical award criteria in tenders.")

C Preliminary Analysis of Sustainability Keyword Mentions over Time

C.1 Prompt for Extracting Award Criteria Related to Sustainability

The following prompt was used to narrow down the list of award criteria for further inspection of sustainability related keywords. Since there were many different ACs and we had to fit them within the maximum sequence length of the LLM, we iterated through the entire list of ACs in batches of 500 criteria. These were concatenated into a string that had a dash followed by a space and the criterion on each line. This string was fed into the prompt template and is denoted as `criteria`. Additionally, we added a restriction on the number criteria that should be extracted per batch, denoted as `num_criteria` in the prompt:

```
Du bist ein Nachhaltigkeits-Experte im Bereich des öffentlichen Beschaffungswesens.
Deine Aufgabe ist es, dir eine Liste von Zuschlagskriterien anzuschauen,
welche aus verschiedenen Ausschreibungen stammen, und diejenigen Kriterien zu
identifizieren, welche einen starken Bezug zu Nachhaltigkeit haben. Es ist äusserst
wichtig, dass du die Kriterien nicht umformulierst. In deiner Auflistung solltest du
alle Kriterien genau gleich schreiben, wie sie geschrieben waren als sie dir
präsentiert wurden. Erwähne jedes Kriterium höchstens einmal in deiner Liste, kein
Kriterium sollte mehrfach in deiner Liste enthalten sein!

Identifiziere die {num_criteria} relevantesten Kriterien im Zusammenhang mit
Nachhaltigkeit.

Der Output sollte exakt folgendermassen strukturiert sein und er sollte keinen
sonstigen Text oder Erklärungen enthalten, nur die identifizierten Kriterien:

Input - Liste mit Zuschlagskriterien aus verschiedenen Ausschreibungen:
- <Kriterium 1>
- <Kriterium 2>
...
- <Kriterium n>

Output - Zuschlagskriterien mit einem starken Zusammenhang zu Nachhaltigkeit:
- <Nachhaltiges Kriterium 1>
- <Nachhaltiges Kriterium 2>
...
- <Nachhaltiges Kriterium {num_criteria}>

---

Deine Aufgabe beginnt jetzt:

Input - Liste mit Zuschlagskriterien aus verschiedenen Ausschreibungen:
- {criteria}

Output - Zuschlagskriterien mit einem starken Zusammenhang zu Nachhaltigkeit:
```

In our experiments, we set the number of output criteria per batch to 10. However, the model did not always follow this instruction. After generating all of the candidate criteria, we matched them against the entire list of criteria and only kept the ones that appeared in the original list to prevent us from using hallucinated keywords.

We also experimented with an English version of the prompt above, which yielded similar results:

```
You are a sustainability expert in the public procurement domain. Your task is to look through a list of award criteria mentioned in different projects to identify the criteria that are closely related to sustainability. It is very important that you do not rephrase any of the criteria. Instead you should list the identified criteria writing them in the same way they were presented to you. Only include each criterion once in your list, do not list the same criterion multiple times!

Identify the {num_criteria} most relevant criteria related to sustainability.

The output should be exactly structured as follows and it should not include any additional text or explanation, just the extracted criteria:

Input - List of award criteria from different projects:
- <criteria 1>
- <criteria 2>
...
- <criteria n>

Output - Award criteria that are clearly about sustainability:
- <sustainability criteria 1>
- <sustainability criteria 2>
...
- <sustainability criteria {num_criteria}>

---

Your task starts now:

Input - List of award criteria from different projects:
- {criteria}

Output - Award criteria that are clearly about sustainability:
```

C.2 Sustainability Keywords

The final list of sustainability keywords that were used to analyze the evolution of CFTs mentioning award criteria related to sustainability over time are the following:

- abwärmenutzung
- blauer engel
- co2
- emicode ec1
- emission
- energieeffizienz
- energieverbrauch
- entsorg
- erneuerbar
- graue energie
- klima
- kreislaufwirtschaft
- lebensdauer
- lohngleichheit
- minergie
- nachhaltig
- ökolog
- photovoltaik
- recycl
- schadstoff
- schweizer holz
- treibhausgas
- umwelt
- verschmutzung
- wiederverwe

Note that some of the keywords are not entire words but only word stems. These keywords were matched against the extracted award criteria using a case-insensitive regex pattern that combined all of the keywords:

```
(?i)abwärmenutzung|blauer engel|...|wiederverwe
```

Chapter 2

Scientific Junior Track

Evaluating Labor Market Biases Reflected in German Word Embeddings

Leander Rankwiler and Mascha Kurpicz-Briki

Applied Machine Intelligence, Bern University of Applied Sciences

Biel/Bienne, Switzerland

leander.rankwiler@students.bfh.ch

Abstract

An unsolved issue in the domain of Natural Language Processing (NLP) is the perpetuation of stereotypical biases inherent in the training data. This has led to increased attention in the research community, but the focus has predominantly been on English models, often neglecting models for other languages. This work aims to counter this trend by investigating bias in German word representations. This analysis includes representations that focus on the word itself, known as static word embeddings, and extends to contextualized embeddings that take into account the context provided by surrounding words. The German datasets for this research are partly derived from a workshop with experts from different fields, including human resources and machine learning in Switzerland. The workshop aimed to identify language-specific biases relevant to the labor market. Our analysis shows that both static and contextualized German embeddings exhibit significant biases along several dimensions.

1 Introduction

Natural Language Processing (NLP) is widely applied in various domains, with its most recent and prominent influence being in language generation. Word embeddings are key components of NLP applications. These vector representations capture semantic meaning in a numerical representation. Studies have demonstrated biases in these embeddings related to gender, race, ethnicity, and other dimensions (Bolukbasi et al., 2016; Caliskan et al., 2017; May et al., 2019). For example, science-related words were found to be more correlated with male-related words such as *him*, *brother*, and *man* compared to female-related words. These stereotypes perpetuate existing social and racial hierarchies (Gao et al., 2020; Bender et al., 2021; Lauscher et al., 2022), leading to unfair treatment and discrimination of certain groups (Köchling and Wehner, 2020). To measure and mitigate these un-

wanted stereotypes, there has been a surge in NLP bias research. However, as Ramesh et al. (2023) point out, languages other than English do not get the attention they deserve. They should be studied separately (Kurpicz-Briki and Leoni, 2021), especially German with its rich morphology and gender marking (Bartl et al., 2020a). In addition, Zhao et al. (2020) note that since biases in multilingual models are transferred to other languages, it is crucial to understand relevant stereotypes in the respective languages themselves. This paper aims to fill this research gap by providing an analysis of biases in established, pre-trained German word embeddings, considering both static and contextualized variants. While static word embeddings focus on the word in question itself, contextualized word embeddings also take into account the context in which the word is used. Bias is quantified with a metric, that uses topic-specific (e.g., male/female and productivity) wordlists. In the presented research we focus on real-world biases from the labor market. We rely on two sources for this analysis. Firstly, we utilize data from the outcomes of a dedicated co-creation workshop conducted with German speakers from different domains including human resources, machine learning, non-governmental organizations, and the legal field. Secondly, we refer to existing datasets for bias detection in word embeddings (Caliskan et al., 2017; Kurpicz-Briki, 2020). Our research is guided by the following research questions:

(RQ1) Are the identified societal biases at the co-creation workshop reflected in German static word embeddings?

(RQ2) Is there evidence of bias in the German contextualized embeddings:

(a) for the same wordlists used for RQ1?

(b) in the German translations of the wordlists from Caliskan et al. (2017)?

(c) in the German wordlists from Kurpicz-Briki (2020)?

2 Related Work

Common tests used for bias detection in English word embeddings cannot always be reproduced for other languages. [Lauscher and Glavaš \(2019\)](#) examined gender bias in German static embeddings with regard to math/art and gender bias with regard to science/art. They found no significant bias in these dimensions; [Kurpicz-Briki \(2020\)](#) confirms this finding. However, [Kurpicz-Briki \(2020\)](#) presents two German wordlists, both of which show significant biases in static embeddings. GER1 examines gender bias, comparing female versus male study choices. GER2 examines historical gender role perceptions, focusing on stereotypes of rationality versus emotionality. Two other studies of German word embeddings highlight the inherent challenges of detecting biases. [Bartl et al. \(2020b\)](#) created a dataset specifically designed to uncover gender bias in the context of the labor market. However, they encountered limitations with the gender-specific postfix forms of occupations in German (postfix: 'in' for female), which inadvertently distorted the associations. [Kraft et al. \(2022\)](#) developed a German language regard classifier that showed a bias towards positive classifications for female subjects. This finding was initially attributed to positive stereotyping, but on closer inspection the authors found the cause to be sexist stereotyping.

3 Methods

3.1 Static Embeddings

Embedding: Fasttext We use Fasttext ([Bojanowski et al., 2017](#)) as our pre-trained static word embedding because it is available for multiple languages, thereby allowing us to test our German wordlists. As it uses sub-words (parts of words, or characters) it is well suited to morphologically rich languages like German ([Bojanowski et al., 2017](#)). The model is trained on Common Crawl and Wikipedia datasets ([Grave et al., 2018](#)).

Metric: WEAT To assess bias in word embeddings, researchers have developed a range of metrics specifically designed to assess bias in word embeddings. A well-known example is the Word Embedding Association Test (WEAT), developed by [Caliskan et al. \(2017\)](#), which we use in our analysis of static embeddings. Their method is based on the Implicit Association Test (IAT), a well-established psychological method for measuring implicit biases ([Greenwald et al., 1998](#)). Its

widespread use in research e.g., ([Chaloner and Maldonado, 2019](#); [May et al., 2019](#); [Chávez Mulsa and Spanakis, 2020](#)), and its adaptability to languages beyond English are additional reasons to use it in our tests. [Caliskan et al. \(2017\)](#) test this method with ten wordlists derived from the underlying psychological literature ([Greenwald et al., 1998](#)), they are referred to as WEAT1-WEAT10. The metric WEAT quantifies bias by comparing the vector representations of the assumed bias topics, which are captured in the wordlists. For a detailed explanation, refer to Appendix C.

3.2 Contextualized Embeddings

Embedding: BERT Bidirectional Encoder Representations from Transformers (BERT) ([Devlin et al., 2019](#)) is an open-source architecture for contextualized embeddings, which is available in several languages, and widely used in research. Because of the above, we perform our analysis on BERT. We use the version “bert-base-german-cased”, updated in 2020, which is trained on data from Wikipedia, German court citations, and news articles¹.

Metric: SEAT For contextualized embeddings we use the Sentence Embedding Association Test (SEAT) proposed by [May et al. \(2019\)](#). The underlying method is based on the methodology of WEAT, with the difference that SEAT is able to measure bias in sequences of words, i.e. sentences. We chose SEAT because the underlying sentence templates can be built from WEAT-like wordlists in languages other than English. In addition, SEAT is the most widely used metric for contextualized word embeddings ([He et al., 2022](#)), thus allowing comparison with other research. The concept of SEAT is to insert the words of interest into unspecific contexts, which the authors call ‘semantically bleached sentences’ - sentences that are deliberately empty of much meaning to allow the inserted word to stand out. Examples are:

- This is <word>.
- <word> is here.

To generate a single representation of the sentence they use the <CLS> token of the embedding (in the case of BERT). [CLS] stands for classification and represents a vector containing the semantic meaning of the whole sentence ([McCormick, 2020](#)).

¹<https://huggingface.co/google-bert/bert-base-german-cased>

4 Data

To quantify the biases, the metrics WEAT and SEAT are dependent on wordlists, which for our purposes originate from three different sources. The wordlists CW1-CW4 are derived from a co-creation workshop with German speaking domain experts from human resources, machine learning, non-governmental organizations, and the legal field². The co-creation workshop was organized as an activity in the EU research project BIAS to discuss the topic of bias and discrimination with interdisciplinary stakeholders. During the workshop, participants engaged in structured discussions and exercises to identify language-specific biases in the labor market. The resulting data is the foundation for the creation of the wordlists CW1-CW4. They cover biases related to gender, hobbies, family status, immigration, and productivity. The following are some words from each of the four categories of CW2, which captures the bias that productivity is related to age:

CW2: Productivity / Age:

Productive: *Effizienz, Leistungswille, Ambition, ...*

Unproductive: *Ablenkung, Ineffizienz, ...*

Old: *Älterer, Ältere, Lebensmitte, Erwachsene, ...*

Young: *Jugendlicher, Jugendliche, Jugend, ...*

Inspired by the bias results of CW1-CW4, we additionally suggest the wordlist CW5, which is a combination of CW1 and CW4 and suggests a gender bias related to productivity. CW1-CW5 are listed in the Appendix E. They are evaluated for static as well as for contextualized embeddings. The wordlists GER1 and GER2 and the German translations of WEAT7 and WEAT8 have been created and investigated by Kurpicz-Briki (2020) for static embeddings. We extend this study by investigating these wordlists on contextualized embeddings. The sentences for the contextualized analysis are created by integrating the words into semantically neutral sentences. For instance, instead of the standalone word *Frau* (woman), sentences such as *Dies ist eine Frau* (This is a woman) are formulated. This is done with the help of GPT-4 (Achiam et al., 2023), the prompt for the creation can be found in the Appendix B, and the sentences are publicly available³.

²<https://www.biasproject.eu/>

³<https://github.com/BFH-AMI/BIAS>

5 Results

As Schröder et al. (2024) suggest, we report effect size (ES), as well as p-values (p) of the WEAT and SEAT metrics, shown in Table 1.

	FastText		GermanBERT	
	ES	p	ES	p
CW1	1.26	0.003	1.04	< 0.001
CW2	0.91	0.023	0.99	< 0.001
CW3	0.67	0.141	-0.14	0.744
CW4	1.46	0.003	1.11	< 0.001
CW5	1.10	0.002	0.55	< 0.001
GER1	<i>1.74</i>	< <i>0.001</i>	0.58	0.005
GER2	<i>1.43</i>	<i>0.002</i>	0.98	< 0.001
WEAT7	<i>0.23</i>	<i>0.65</i>	-0.04	0.594
WEAT8	<i>0.11</i>	<i>0.83</i>	-0.36	0.98

Table 1: ES = effect size; p = p-value. **Bold** results are significant at the < 0.05 level. *Italic* results are from Kurpicz-Briki (2020)

6 Discussion

6.1 CW1 - CW5

We demonstrated that, of the four dimensions of bias identified during the co-creation workshop (CW1-CW4), three exhibit significant bias in both static and contextual analyses, thereby affirmatively addressing RQ1 (CW: bias in static embeddings?) and RQ2a (CW: bias in contextualized embeddings?). This supports existing research that argues the efficiency of language-specific bias identification. CW1 and CW2 highlight the stereotype that productivity conflicts with family and old age, respectively. This finding is echoed outside the domain of NLP by researchers that investigate real-world bias in the labor market: Pärli (2018) found that older people are disadvantaged in the Swiss professional environment, and Kleinert (2006) found that women with children are disadvantaged in obtaining managerial positions. The CW3 wordlist does not yield significant results, possibly because 'traditionalists' are not the direct counterparts of 'communicators', which could reduce the effect size of the wordlist. Hobbies like *Backen, Kunst, Ballett* are related to female terms, as shown by CW4. When combined with the findings of CW5, these results could perpetuate problematic stereotypes. CW5 finds a link between productivity and gender. Taken together, these findings could suggest a correlation whereby typical male hobbies are associated with productivity and typical female hobbies with unproductivity. However, a direct experiment did not confirm this speculation.

	Dutch	English	German (ours)
Static (WEAT)	7, 8: FastText	7, 8: Glove, word2vec 7, 8: FastText	7, 8: FastText
Contextualized (SEAT)	7: BERTje, RobBERT 8: BERTje, RobBERT	7: BERT 8: BERT	7, 8: German BERT

Table 2: Comparison of presence of bias in different languages and embeddings. **Bold** = significant at < 0.05 . 7,8: number of WEAT wordlist. Dutch results are equal for BERTJe (de Vries et al., 2019) and RobBERT (Delobelle et al., 2020). GloVe: static word embedding from Pennington et al. (2014), Word2Vec: static word embedding from Mikolov et al. (2013). Dutch results by Chávez Mulsa and Spanakis (2020). English static results for Glove and Word2Vec by Caliskan et al. (2017) and for FastText by Lauscher and Glavaš (2019); Kurpicz-Briki (2020). English contextualized results by May et al. (2019). The BERT models are available online with the identifier (bert-base-dutch-cased, robbert-2023-dutch-large, bert-base-cased, bert-base-german-cased) on <https://huggingface.co/>.

As the results of Kraft et al. (2022) demonstrate, we should be cautious to draw conclusions too quickly in the domain of bias detection.

6.2 WEAT7, WEAT8

The German WEAT7 and WEAT8 results from Kurpicz-Briki (2020) are consistent with our SEAT results from BERT, thus not confirming RQ2b (WEAT7, WEAT8: bias in contextualized embeddings?). Comparing these results with WEAT results in other languages, we find no obvious correlation across languages, refer to Table 2. This could be an indication that language specificity is relevant.

6.3 GER1, GER2

The results for the static embeddings from Kurpicz-Briki (2020) are confirmed in our contextualized setting with BERT, thus answering RQ2c (GER1, GER2: bias in contextualized embeddings?) positively.

6.4 Correlation Static to Contextualized

The correlation between static and contextualized results (from Table 1) is high, i.e. wordlists with low p-values in FastText analyses tend to show low p-values in BERT tests (the same counts for high p-values). To quantify this correlation, we perform a meta-analysis of our results, refer to Appendix D for the numerical results. This finding supports the hypothesis that WEAT wordlists can be effectively transferred to SEAT through contextualization with semantically bleached sentences. The two tested models are partly trained on the same data, which could explain the high correlation. These results support the validity of our approach and suggest that the two models have at least some common bias directions.

6.5 Static WEAT Results

The numerical results of Lauscher and Glavaš (2019) and Kurpicz-Briki (2020) for FastText of the German translations of WEAT7 and WEAT8 differ slightly, but the conclusions are the same. Their difference might be due to different translation approaches. For example, the term *dance* was translated as *tanzen* by Lauscher and Glavaš (2019) and as *Tanz* by Kurpicz-Briki (2020). We use the results of Kurpicz-Briki (2020) for comparison with ours, as the p-values are reported. See Appendix A for both numerical results.

6.6 Future Work

Further exploration of additional datasets may prove fruitful. For example, dividing CW3 into two distinct wordlists (e.g., comparing immigration status to traditionalists vs. progressives). In addition, to further explore the importance of language specificity in bias assessment, the CW1-CW4 wordlists could be translated into other languages and tested with corresponding models. The significance of these results would be even greater if more languages were considered to provide a quantitative analysis of language specificity.

7 Conclusion

Our investigation of bias within German word embeddings, covering both static and contextualized models, reveals bias along several dimensions. Gender, age and family status biases were particularly prevalent, reflecting societal stereotypes found in the real world. We also found a strong correlation between bias results in static and contextualized embeddings. Furthermore, our results suggest that language specificity is important for identifying and understanding bias.

Limitations

Language Specificity Our research only suggests that language specificity is important, it does not prove it. To make a stronger statement, more languages need to be considered. This could be done by translating the datasets into different languages and comparing the biases in the respective models.

Missing Extrinsic Metrics Intrinsic metrics - as the ones used in this study - measure the bias in the pre-trained representations of the model. Extrinsic metrics quantify the bias that appears in the results of the model’s downstream task. Recent studies, including those by [Cao et al. \(2022\)](#); [Kaneko et al. \(2022\)](#); [Orgad et al. \(2022\)](#), have shown that the correlation between intrinsic and extrinsic metrics is very limited. As highlighted by [Orgad and Belinkov \(2022\)](#), the inclusion of extrinsic metrics is critical for several reasons, including the greater relevance of these metrics to bias mitigation efforts. While we fully acknowledge these recommendations, we only use intrinsic metrics because the availability of extrinsic datasets for non-English languages is very limited ([Ramesh et al., 2023](#)). This finding is echoed by [Wambsganss et al. \(2022\)](#), who analyze the bias in German embeddings at different stages along the NLP pipeline. They find that when a pre-trained model that shows no bias on a particular metric is fine-tuned with unbiased data (on the same metric), it can produce biased output (measured again on the same metric). This underlines that the intrinsic evaluation done with WEAT and SEAT can at best be a signal of bias, a sentiment reflected by [Goldfarb-Tarrant et al. \(2021\)](#).

Missing Replicability of Sentences The use of GPT-4’s chat interface to generate sentences for the SEAT metric introduces a replicability limitation, as it is not possible to consistently generate exactly the same model output. To enable replicability, future research could use GPT-4’s API to generate sentences, setting the temperature parameter to zero to ensure deterministic output.

Ethical considerations

We only consider binary gender bias, and therefore do not consider non-binary gender identities. This does not reflect what is found in the real world ([Devinney et al., 2022](#)). The BERT model has been shown to fail to represent non-binary gender in a

meaningful way ([Dev et al., 2021](#)), which further complicates matters.

The data from the co-creation workshop was collected anonymously and with the individual’s consent following the approved data protection research protocol of the project.

Acknowledgements

This work is part of the Europe Horizon project BIAS funded by the European Commission, and has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Marion Bartl, Malvina Nissim, and Albert Gatt. 2020a. [Unmasking contextual stereotypes: Measuring and mitigating BERT’s gender bias](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 1–16, Barcelona, Spain (Online). Association for Computational Linguistics.
- Marion Bartl, Malvina Nissim, and Albert Gatt. 2020b. [Unmasking Contextual Stereotypes: Measuring and Mitigating BERT’s Gender Bias](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 1–16, Barcelona, Spain (Online). Association for Computational Linguistics.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?](#). In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’21, pages 610–623, New York, NY, USA. Association for Computing Machinery.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching Word Vectors with Subword Information](#). ArXiv:1607.04606 [cs].
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. [Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. [Semantics derived automatically from language corpora contain human-like biases](#). *Science*, 356(6334):183–186. Publisher: American Association for the Advancement of Science.

- Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. [On the Intrinsic and Extrinsic Fairness Evaluation Metrics for Contextualized Language Representations](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.
- Kaytlin Chaloner and Alfredo Maldonado. 2019. [Measuring Gender Bias in Word Embeddings across Domains and Discovering New Gender Bias Word Categories](#). In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 25–32, Florence, Italy. Association for Computational Linguistics.
- Rodrigo Alejandro Chávez Mulsa and Gerasimos Spanakis. 2020. [Evaluating Bias In Dutch Word Embeddings](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 56–71, Barcelona, Spain (Online). Association for Computational Linguistics.
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. [BERTje: A Dutch BERT Model](#). ArXiv:1912.09582 [cs].
- Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. [RobBERT: a Dutch RoBERTa-based Language Model](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.
- Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. [Harms of Gender Exclusivity and Challenges in Non-Binary Representation in Language Technologies](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hannah Devinney, Jenny Björklund, and Henrik Björklund. 2022. [Theories of “Gender” in NLP Bias Research](#). In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’22*, pages 2083–2102, New York, NY, USA. Association for Computing Machinery.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. [The Pile: An 800GB Dataset of Diverse Text for Language Modeling](#). ArXiv:2101.00027 [cs].
- Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. 2021. [Intrinsic Bias Metrics Do Not Correlate with Application Bias](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1926–1940, Online. Association for Computational Linguistics.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning Word Vectors for 157 Languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. 1998. [Measuring individual differences in implicit cognition: The implicit association test](#). *Journal of Personality and Social Psychology*, 74:1464–1480. Place: US Publisher: American Psychological Association.
- Jacqueline He, Mengzhou Xia, Christiane Fellbaum, and Danqi Chen. 2022. [MABEL: Attenuating Gender Bias using Textual Entailment Data](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9681–9702, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2022. [Debiasing Isn’t Enough! – on the Effectiveness of Debiasing MLMs and Their Social Biases in Downstream Tasks](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1299–1310, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Corinna Kleinert. 2006. [Frauen in Führungspositionen: Karriere mit Hindernissen](#). Research Report 9/2006, IAB-Kurzbericht.
- Angelie Kraft, Hans-Peter Zorn, Pascal Fecht, Judith Simon, Chris Biemann, and Ricardo Usbeck. 2022. [Measuring Gender Bias in German Language Generation](#).
- Mascha Kurpicz-Briki. 2020. [Cultural Differences in Bias? Origin and Gender Bias in Pre-Trained German and French Word Embeddings](#).
- Mascha Kurpicz-Briki and Tomaso Leoni. 2021. [A World Full of Stereotypes? Further Investigation on Origin and Gender Bias in Multi-Lingual Word Embeddings](#). *Frontiers in Big Data*, 4.

- Alina Köchling and Marius Claus Wehner. 2020. [Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development](#). *Business Research*, 13(3):795–848.
- Anne Lauscher, Archie Crowley, and Dirk Hovy. 2022. [Welcome to the Modern World of Pronouns: Identity-Inclusive Natural Language Processing beyond Gender](#). ArXiv:2202.11923 [cs].
- Anne Lauscher and Goran Glavaš. 2019. [Are We Consistently Biased? Multidimensional Analysis of Biases in Distributional Word Vectors](#). In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 85–91, Minneapolis, Minnesota. Association for Computational Linguistics.
- Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. [On Measuring Social Biases in Sentence Encoders](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.
- Chris McCormick. 2020. *The Inner Workings of BERT*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient Estimation of Word Representations in Vector Space](#). ArXiv:1301.3781 [cs].
- Hadas Orgad and Yonatan Belinkov. 2022. [Choose Your Lenses: Flaws in Gender Bias Evaluation](#). In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 151–167, Seattle, Washington. Association for Computational Linguistics.
- Hadas Orgad, Seraphina Goldfarb-Tarrant, and Yonatan Belinkov. 2022. [How Gender Debiasing Affects Internal Model Representations, and Why It Matters](#). arXiv. Version Number: 2.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global Vectors for Word Representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Kurt Pärli. 2018. [Altersdiskriminierung - von der Anstellung bis zur Kündigung](#). *Schulthess Juristische Medien 2018*, pages 1 – 10.
- Krithika Ramesh, Sunayana Sitaram, and Monojit Choudhury. 2023. [Fairness in Language Models Beyond English: Gaps and Challenges](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2106–2119, Dubrovnik, Croatia. Association for Computational Linguistics.
- Sarah Schröder, Alexander Schulz, Fabian Hinder, and Barbara Hammer. 2024. [Semantic Properties of cosine based bias scores for word embeddings](#). ArXiv:2401.15499 [cs].
- Thiemo Wambsganss, Vinitra Swamy, Roman Rietsche, and Tanja Käser. 2022. [Bias at a Second Glance: A Deep Dive into Bias for German Educational Peer-Review Data Modeling](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1344–1356, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Jieyu Zhao, Subhabrata Mukherjee, Saghar Hosseini, Kai-Wei Chang, and Ahmed Hassan Awadallah. 2020. [Gender Bias in Multilingual Embeddings and Cross-Lingual Transfer](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2896–2907, Online. Association for Computational Linguistics.

A WEAT7, WEAT8 German Results

The results of [Lauscher and Glavaš \(2019\)](#) and [Kurpicz-Briki \(2020\)](#) point to the same conclusion with slightly different values. Table 3 shows a comparison of the reported WEAT scores for the FastText embedding for the German WEAT lists. The difference can be attributed to differences in the translations of the WEAT wordlists.

	Lauscher 2019	Kurpicz-Briki 2020
WEAT 7	0.46 (p>0.05)	0.23 (p=0.65)
WEAT 8	0.05 (p>0.05)	0.11 (p=0.83)

Table 3: Comparison of WEAT scores.

B GPT-4 Prompt

The following prompt is used to create the sample sentences for SEAT. This prompt contains plural sample sentences. Target words sometimes do not make sense in the plural form, so we used a shorter version of the prompt, only containing singular sample sentences (the first seven sentences). Using GPT-4 instead of manually coding sentences eliminates the need to define various linguistic elements. These include determining the gender of words (neuter, masculine, or feminine), distinguishing between things and people, knowing the plural forms of words, and deciding whether to use articles in specific sentences (nos. 1-4, 12, 13).

Dies ist (der/die/das) XX.
 Das ist (der/die/das) XX.
 Dort ist (der/die/das) XX.

Hier ist (der/die/das) XX.
 (Der/Die/Das) XX ist hier.
 (Der/Die/Das) XX ist dort.
 (Der/Die/Das) XX ist (ein/eine)
 (Sache/Mensch).
 Es ist (der/die/das) XX.
 Dies sind XX.
 Das sind XX.
 Sie sind XX.
 (Die/) XX sind hier.
 (Die/) XX sind dort.
 XX sind (Sachen/Menschen).

Ersetze in der oben genannten Vorlage die Sätze "XX" mit den untenstehenden Wortlisten. Dies ergibt 14 x 5 Sätze. Passe die Sätze an, damit sie grammatikalisch korrekt sind. Wenn nötig, ändere das Wort ins Plural, damit es zur Vorlage passt. Verwende beim Satz Nr. 7 und 14 "Sache(n)", ausser "Mensch(en)" passt offensichtlich besser ("Sache" wird manchmal nicht passen, verwende es trotzdem). Schreibe keinen code um dies zu erreichen. Gib dies im CSV-Format zurück, jeder Satz auf einer neuen Zeile:

"Dies ist das XX.",
 "Das ist das XX.",
 "Dort ist das XX.",
 usw.

Wortliste:
 Mann, Junge, Bruder, Sohn, Vater

C WEAT Method

The metric is based on the Implicit Association Test (IAT), where subjects are presented with two concepts, for example school subjects (e.g., *Science*, *Arts*), and gender (*Male*, *Female*). Short reaction times to classify e.g., *Science* and *Male* in a given class indicate cognitive proximity of *Male* and *Science*. In the context of static word embeddings, WEAT uses cosine similarity as a proxy for reaction time in the IAT. Cosine similarity measures the cosine of the angle between two vectors, serving as an indicator of their semantic proximity in vector space. In the following example, the association between school subjects (target words) and gender (attribute words) is compared. The attribute and target words are also referred to as stimuli. In

the example, a smaller angle between *Science* and *Male* (represented by a blue dotted line in Figure 1) indicates that these two concepts are closely related. The angle from *Science* to *Female* (represented by a blue dashed line) is then subtracted from the angle *Science* to *Male* (represented by a blue dotted line). This results in an angle that quantifies the degree of relationship between the concept *Science* and the gender attributes *Male* and *Female*. This calculation is then performed for another target word (here: *Arts*) and its relation to gender (marked in green in Figure 1). The output of the calculation of the word *Arts* is compared to its counterpart for *Science*. In a perfectly unbiased embedding, these two angles should be identical. In the provided example this would clearly not be the case, as the two results of the dotted minus the dashed angles are not equal. The described procedure is done for a set of target words (e.g., *programmer*, *engineer*, *scientist*, ... and *nurse*, *teacher*, *librarian*, ...) and a set of attribute words (e.g., *man*, *male*, *he*, ... and *woman*, *female*, *her*, ...). The mean of the angles is used to aggregate the sets. The null hypothesis is that the relative similarity of the two sets of target words to the two sets of attribute words is identical. For the formulas used to compute the effect size and the p-value, we refer the reader to the original paper by Caliskan et al. (2017).

D Correlation Static to Contextualized

The Table 4 shows p-values for different dataset combinations. The consistently low values across combinations of CW1-5, GER1-2, and WEAT7-8 datasets reinforces the observed correlation between static and contextualized word embeddings.

Configuration	p-value
CW1-CW5	0.016
CW1-CW5 + GER1,2	<0.001
CW1-CW5 + GER1,2 + WEAT7,8	0.001

Table 4: Correlation between static and contextualized word embeddings

E CW1-CW5 Wordlists

The Tables 5,6,7,8 and 9 list the wordlists CW1 - CW5 used for the WEAT metric. These lists are also used for the creation of the sentence templates for the SEAT metric via the GPT-4 prompt.

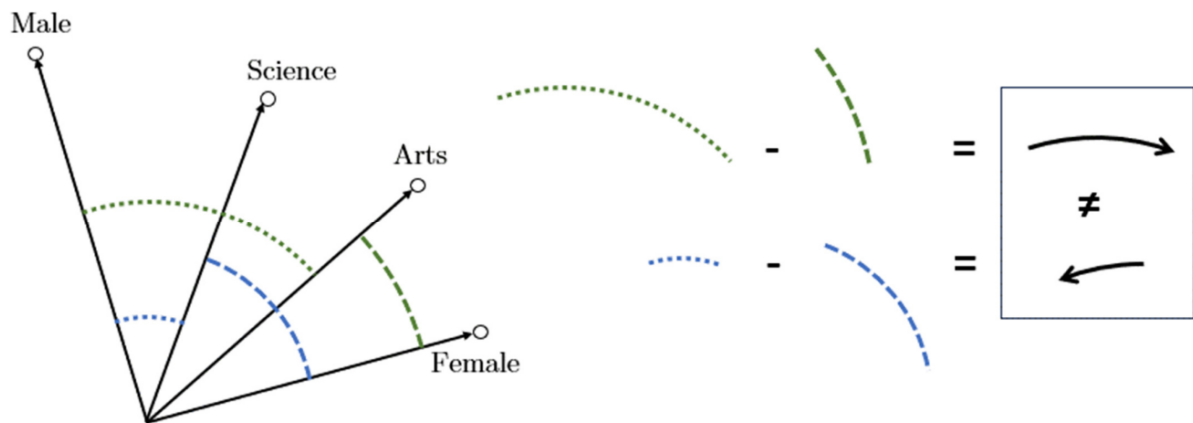


Figure 1: Angles of the attribute words (Male, Female) and target words (Science, Arts), (Own illustration)

Category	Keywords
Productive	Effizienz, Leistungswille, Engagement, Ambition, Zielstrebigkeit, Kompetenz, Motivation, Einsatzbereitschaft, Beharrlichkeit, Eifer
Unproductive	Ablenkung, Unaufmerksamkeit, Ineffizienz, Nachlässigkeit, Desorganisation, Überforderung, Zerstreuung, Desinteresse, Unachtsamkeit, Sorglosigkeit
Business	Geschäftsleitung, Management, Fachkraft, Unternehmen, Gehalt, Büro, Geschäft, Karriere
Children	Elternteil, Kindertagesstätte, Sohn, Tochter, Kinder, Kinderwagen, Schnuller, Windel

Table 5: CW1 wordlists

Category	Keywords
Productive	Effizienz, Leistungswille, Engagement, Ambition, Zielstrebigkeit, Kompetenz, Motivation, Einsatzbereitschaft, Beharrlichkeit, Eifer
Unproductive	Ablenkung, Unaufmerksamkeit, Ineffizienz, Nachlässigkeit, Desorganisation, Überforderung, Zerstreuung, Desinteresse, Unachtsamkeit, Sorglosigkeit
Young	Jugendlicher, Jugendliche, Jugend, Studium, Berufseinstieg, Mittzwanziger, Schülerin, Schüler, Student, Studentin
Old	Älterer, Ältere, Lebensmitte, Erwachsene, Lebenserfahrung, Erwachsener, Rentner, Rentnerin, Senior, Seniorin

Table 6: CW2 wordlists

Category	Keywords
Traditionalists	Bewahrung, Konservatismus, Unflexibilität, Konventionalität, Tradition, Gewohnheit
Communicators	Kommunikation, Vermittlung, Verbindung, Vielfalt, Mehrsprachigkeit, Integration
Native	Einheimischer, Bürger, Staatsangehöriger, Landesangehöriger, Ansässiger, Ortsansässiger
Immigrant	Ausländer, Einwanderer, Immigrant, Migrant, Migrationshintergrund, Gastarbeiter

Table 7: CW3 wordlists

Category	Keywords
Male Hobbies	Gewichtheben, Wrestling, Feuerwehr, Velorennen, Schrauben
Female Hobbies	Backen, Ökologie, Make-Up, Kunst, Ballett
Male	Mann, Junge, Bruder, Sohn, Vater
Female	Frau, Mädchen, Schwester, Tochter, Mutter

Table 8: CW4 wordlists

Category	Keywords
Productive	Effizienz, Leistungswille, Engagement, Ambition, Zielstrebigkeit, Kompetenz, Motivation, Einsatzbereitschaft, Beharrlichkeit, Eifer
Unproductive	Ablenkung, Unaufmerksamkeit, Ineffizienz, Nachlässigkeit, Desorganisation, Überforderung, Zerstreuung, Desinteresse, Unachtsamkeit, Sorglosigkeit
Male	Mann, Junge, Bruder, Sohn, Vater
Female	Frau, Mädchen, Schwester, Tochter, Mutter

Table 9: CW5 wordlists

Exploring Personalized Learning Support through Retrieval Augmented Generation: A Feasibility Study

Petar Mladenov and Luis Pinheiro and Dino Pelesevic and Jasmin Heierli

Zurich University of Applied Sciences, Winterthur, Switzerland

{mladepet, pinhelui, pelesdin}@students.zhaw.ch, heej@zhaw.ch

Abstract

This paper presents a feasibility study on using language models (LMs) in education to generate and evaluate tasks based on Bloom's taxonomy. We developed a prototype combining retrieval-augmented generation and prompt engineering to assess students' cognitive levels. Initial expert feedback highlights our approach's potential, but it emphasizes the need for broader empirical validation. The study provides a foundation for future research into LMs in personalized education and underscores the importance of real-world testing.

1 Introduction

Knowledge dissemination is essential for advancing societies, yet traditional teaching struggles with teacher shortages and diverse student needs (BFS, 2022). The limited availability of support per student is further compounded by the diverse learning abilities and requirements of individual students.

Consequently, digitalization has produced many e-learning aids¹. These tools typically offer interactive tasks, explanations, and sample solutions, but often fail to provide personalized learning support that accurately assesses a student's understanding level. Current e-learning aids, despite offering task-based support, do not fully capture the nuances of a student's comprehension, limiting the effectiveness of personalized learning².

An effective, personalized learning support requires a sound and valid assessment of the student's state of understanding.

¹For example: <https://www.aufgabenfuchs.de>, <https://www.sofatutor.ch>, <https://ilearn.ch>, <http://aufgaben-online.ch>, <https://lernen-mit-spas.ch>, <https://www.schlaukopf.ch>, <https://simpleclub.com>, <https://brilliant.org>

²These services build upon the groundwork laid by earlier work in digital education, which we cannot discuss due to space constraints

This study explores the feasibility of using language models (LM) with Retrieval Augmented Generation (RAG) to enhance e-learning personalization via Bloom's Taxonomy (Krathwohl, 2002). We aim to assess student comprehension more accurately by creating and evaluating tasks across two different comprehension levels, using Bloom's taxonomy not just as a framework, but as a personalization tool.

The remainder of this paper is organized as follows: Sect. 2 outlines how we aim to assess a student's level of understanding and the requirements. Sect. 3 details the implementation of our approach, demonstrating its feasibility. Finally, Sect. 4 discusses the results and potential future work.

2 Requirements

This section details the system's requirements for personalizing the system to assess student understanding of learning materials. We start by describing the desired inputs and outcomes of the system, followed by a list of key requirements.

The objective is to analyze content, identify relevant excerpts, like the one in Figure 1, and generate tasks across two Bloom levels. Using selected

I Beginn der Unternehmenstätigkeit

Das menschliche Leben ist durch Spannungsfelder geprägt: Auf der einen Seite stehen unsere vielfältigen und umfangreichen Bedürfnisse, auf der anderen Seite die knappen Mittel, um diese Bedürfnisse zu befriedigen. Wie jeder Mensch muss auch ein Unterneh-

Figure 1: Example excerpt from an educational resource

excerpts, the LM prompts tasks testing specific Bloom's levels. For instance, one task might assess Bloom's level 1 (Remember), querying a student's ability to recall facts:

Wer ist für die strategische Planung in einem Unternehmen verantwortlich?
A) ...

Another task might evaluate Bloom's level 3 (Apply), examining the ability to apply knowledge

in new scenarios.

Szenario: Sie sind ein Mitglied des Führungsteams der Schweizer Firma "Alpine Goods AG", die qualitativ hochwertige Outdoor-Bekleidung und Ausrüstung produziert und vertreibt...
Frage: Unter Verwendung der SWOT-Analyse (Strengths, Weaknesses, Opportunities, Threats) analysieren Sie die aktuelle Situation von "Alpine Goods AG". Was ist eine der wichtigsten Schwächen, die das Unternehmen beachten und angehen sollte, und welche strategischen Entscheidungen könnten getroffen werden, um diese Schwäche zu adressieren?
A) ...

The generated tasks assess recall of facts (Level 1) and application of knowledge (Level 3). The system corrects responses to determine a student's level of understanding and whether or not they are reaching certain Bloom's levels.

To investigate the feasibility, an implementation must meet these requirements:

- The system shall be able to process different educational resources and use them for assessing the student's learning level for a specific subject.
- The system shall select suitable study material for task generation using a simple keyword.
- The system shall be able to generate tasks that assess different Bloom levels.
- The system shall be able to generate response options for single- or multiple-choice tasks.
- The system shall be able to evaluate the correctness of student responses.

Having defined these requirements, we further restrict our focus to German texts in PDF format and aim to use RAG for these purposes. The next section explains our implementation approach to meet these requirements.

3 Implementation

This section outlines the implementation of a lightweight prototype, designed to be feasible on standard computers and evaluated for potential by an expert. Figure 2 illustrates the system's architecture, featuring four main components.



Figure 2: System architecture.

3.1 Text to vectors

Addressing the need to process diverse educational resources, the prototype uses Chroma³ vector store for storing text as embedding vectors extracted from PDFs using PyPDF2⁴. For the prototype, we did not evaluate the performance of different PDF readers. PyPDF is open source and allows the retrieval of text from PDFs. Texts are chunked to align with the embedding model (paraphrase-multilingual-MiniLM-L12-v2⁵), optimized for semantic search in German. The chunking and embedding processes, though practical, lack extensive technical validation at this stage.

As there is currently not much research around chunking for RAG available (Yepes et al., 2024), we decided to take a practical approach respecting the input length of the embedding model we planned to use. To achieve the desired length, the retrieved texts are chunked with Langchain's RecursiveCharacterTextSplitter⁶ that splits the text into chunks of 1,000 characters. This roughly correspond to 128 tokens, which is the input size of the embedding model we used.

The chunks are then embedded using the paraphrase-multilingual-MiniLM-L12-v2⁷ embedding model that works for short paragraphs of German text (Reimers and Gurevych, 2019). It is small enough to run on a variety of hardware and has been developed for semantic search, specifically. The embedded chunks are then indexed and stored in a Chroma vector store.

Having established the method for converting text to vectors, we next focus on the question generation components for assessing students' understanding levels.

³<https://docs.trychroma.com/>

⁴<https://pythonhosted.org/PyPDF2/>

⁵<https://huggingface.co/cross-encoder/msmarco-MiniLM-L6-en-de-v1>

⁶https://python.langchain.com/docs/modules/data_connection/document_transformers/recursive_text_splitter

⁷<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>

3.2 Vector retrieval

To meet the second requirement of selecting study material by keyword we utilized Chroma’s vector similarity search.

We enhanced traditional methods by using GPT-4⁸ to expand initial terms into five linguistically sound questions, rather than traditionally adding more terms (Carpineto and Romano, 2012).

For example, for an original search term ‘Unternehmenstätigkeit’, we used the prompt

```
Du bist Lehrperson für Betriebswirtschaft. Du bekommst Fragen zum Lehrmittel über Betriebswirtschaft. Schlage bis zu 5 zusätzliche verwandte Fragen vor, um dem Benutzer zu helfen, die Antworten auf seine Frage zu finden...
```

yielding five additional questions such as

```
Was sind die verschiedenen Arten von {given_topic}?
```

These were validated for relevance and appropriateness by an educational expert.

To further ensure the validity of the expanded queries and their retrieved documents, we applied PCA projection. This visualizes the original and expanded queries alongside their results in a 2-D vector space, confirming their alignment and relevance.

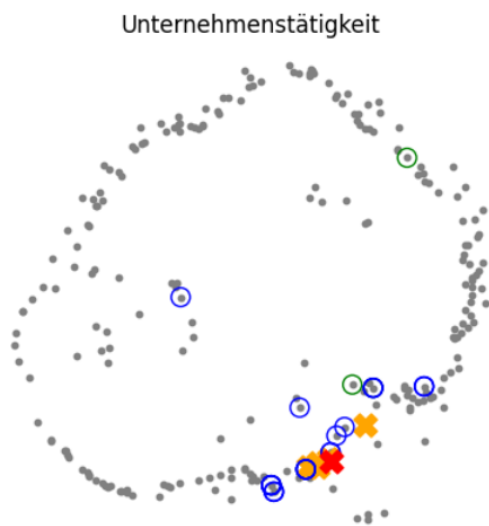


Figure 3: PCA projection of the original (red) and expanded queries (orange) and their respective results (results for original query = green, results for expanded = blue).

The original search term and the five expanded queries then guide the retrieval of the top 5 relevant

⁸<https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo>

chunks from the vector store with Euclidian distance. To avoid information overload and redundancies, we refine our selection to the five most relevant chunks. This is achieved with the ms-marco-MiniLM-L6-en-de-v1 cross-encoder model, which evaluates chunk relevance regardless of origin.

With these refined chunks, we proceed to the question generation as detailed in the next section.

3.3 Task generation

We opted for RAG combined with prompt engineering rather than fine-tuning a model to enable the dynamic generation of tasks for different subjects and learning materials. Fine-tuning would potentially require separate models for different subjects, which increases resource requirements. Furthermore, the repeated fine-tuning of a single model with new materials could lead to inconsistencies, potentially negatively affecting model performance.

Bloom’s taxonomy categorizes educational objectives into six cognitive levels, ranging from simple recall to complex evaluation and creation. As our goal is to evaluate the feasibility of personalization in e-learning, Bloom’s taxonomy is a suitable framework to personalise tasks according to students’ cognitive levels. For the prototype we focused on level 1 (Remember) and level 3 (Apply) tasks, as this allows us to evaluate whether the LM can generate tasks that are personalized to the students’ cognitive levels.

The aim of the task generation is to create tasks aligning with different Bloom’s levels. In the scope of this investigation, we focused on single-choice tasks. Consequently, the task generation must also include the generation of response options. Following the idea of RAG, we combine prompt engineering with text injected into prompts.

Our first attempt to obtain tasks assessing different Bloom’s levels consisted of specifying the Bloom’s level as part of the prompt. While the first results indicated that GPT-4 has inherent knowledge about Bloom’s taxonomy, the generated questions for level 1 were overly simplistic, and those for level 3 lacked comprehensive scenario descriptions — as pointed out by our educational expert.

To address this, we crafted prompts containing keywords and phrases indicative of the desired Bloom level (Krathwohl, 2002). For Bloom level 1, our prompts, such as the one shown next, include verbs such as ‘list’, ‘name’, and ‘describe’, en-

suring simplicity and clarity in the generated tasks.

```
... Deine Aufgabe ist es, eine
Prüfungsfrage auf
Bloom-Niveau 1 zu formulieren, die sich
auf allgemeines Wissen bezieht, das im
Unterricht behandelt wurde. Verwende
einfache Schlüsselwörter
und Verben wie 'sammeln', 'erzählen',
'benennen', 'erinnere', 'was', 'wann',
'wer', 'liste auf', 'zeige', 'gib an'
und ähnliches...
```

The previously retrieved chunks are then injected at the end of such a prompt, followed by an instruction to generate a task solely on this information.

The prompt for the Bloom level 3 task generation also includes instructions to generate a fictitious scenario mentioning theories when applicable to solve the task. While we obtained acceptable results instructing the LM to generate questions and response options for single-choice tasks at once, our educational expert concluded that these response options lacked discriminative power.

We obtained better results by prompting the LM separately for task generation and the generation of answer options. The task generated with the first prompt was included in second prompt. This adds the possibility of specifying separate requirements such as regarding the discriminatory power.

This two-step prompting approach not only enhances the discriminative power of response options but also increases control in tailoring them to specific task requirements, thereby ensuring our prototype's efficacy in engaging students at their personal cognitive level.

3.4 Task correction

The LM evaluated student responses against provided materials, aiming to mimic a teacher's assessment process, as shown in this structured prompt. Accuracy in practical educational settings remains to be tested.

```
Du bist Lehrperson für das Fach
Betriebswirtschaft an einer schweizer
Sekundarschule. Deine Aufgabe ist es,
die Antworten deiner Schüler auf eine
Single-Choice-Frage zu bewerten. Die
Frage lautet: \{question_3\}. Antworte
nur mit 'richtig' oder 'falsch'. Die
Informationen zum Thema sind:
\{retrieved_documents\}. Antwort des
Schülers: \{user_answer\}
```

Note that the expressions in curly brackets are placeholders for the full question including answer options, the retrieved documents and the response the student selects.

4 Conclusion

This feasibility study demonstrates the potential of LMs for personalized education, specifically through developing a prototype that leverages LMs for task generation and response evaluation, targeting the personalized assessment of students' understanding at different Bloom levels.

In alignment with the identified requirements (Section 2), our retrieval-augmented generation approach, which incorporates a Chroma vector store, effectively processes a range of educational resources, selects relevant content, and seamlessly integrates it into LM-generated prompts. The appropriateness of our tasks, customized for specific Bloom levels, was confirmed by an educational expert. Although our current focus has been on two levels of Bloom's taxonomy, the versatility of our approach suggests potential applicability to other means of personalization.

While we focused on two Bloom levels, our results indicate potential for personalization across other dimensions, such as learning styles, interests, and cognitive abilities. Future work will include systematic validation of our techniques for query expansion and cross-coder reranking. Empirical testing with students will also be crucial to evaluate the accuracy of our system in identifying their respective Bloom's comprehension levels. In conclusion, our study not only confirms the feasibility of using LMs in educational settings but also opens avenues for future research, particularly in enhancing personalized learning experiences and understanding student cognitive levels.

References

- BFS. 2022. *Szenarien 2022-2031 für die Lehrkräfte der obligatorischen Schule*. 22806575. Bundesamt für Statistik (BFS), Neuchâtel.
- Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1).
- David R Krathwohl. 2002. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218.
- Nils Reimers and Iryna Gurevych. 2019. *Sentence-bert: Sentence embeddings using siamese bert-networks*. *CoRR*, abs/1908.10084.
- Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. *Financial report chunking for effective retrieval augmented generation*.

SDG Classification Using Instruction-Tuned LLMs

Tobias Fankhauser and Simon Clematide and Martin Volk

Department of Computational Linguistics

University of Zurich

tobias.fankhauser,simon.clematide,martin.volk@uzh.ch

Abstract

This paper investigates the potential of quantized, instruction-tuned Large Language Models (LLMs) for zero-shot classification of scientific abstracts according to the United Nations’ Sustainable Development Goals (SDGs). We introduce the Decompose-Synthesize-Refine-Extract (DSRE) framework, leveraging advanced prompting techniques for both single-label and multi-label classification scenarios. DSRE is designed to enhance the zero-shot capabilities of LLMs for this domain-specific task. We explore the trade-offs between model performance and computational efficiency introduced by quantization. The performance of DSRE and quantized LLMs is benchmarked against fine-tuned LLM baselines and the Aurora system. Our findings demonstrate the potential of instruction-tuned LLMs for zero-shot SDG classification but emphasize the continued value of fine-tuning for optimal performance. Additionally, we consider dataset imbalance and the impact of augmenting datasets.

1 Introduction

In 2015, the United Nations outlined the 2030 Agenda for Sustainable Development, introducing 17 Sustainable Development Goals (SDGs) to address global challenges such as poverty, inequality, and climate change (United Nations: Department of Economic and Social Affairs). The academic community plays a vital role in advancing these goals through research that contributes to the targets defined within each SDG. However, effectively aligning scientific literature with specific SDGs remains a challenge due to the implicit nature of many contributions and the vast scope of subjects.

Recent advancements in Natural Language Processing (NLP), particularly in Large Language Models (LLMs) and generative AI, offer promising avenues for automating the classification of scientific abstracts into relevant SDGs.

This paper explores the application of quantized, instruction-tuned LLMs for zero-shot SDG classification of scientific abstracts. We propose the Decompose-Synthesize-Refine-Extract (DSRE) framework, an advanced prompt decomposition approach tailored for both single-label and multi-label classification tasks. DSRE aims to maximize the zero-shot capabilities of LLMs for domain-specific classification, addressing the need for data-efficient methods in aligning research outputs with SDGs.

We examine the balance between computational efficiency and model performance afforded by quantization and assess the effectiveness of DSRE compared to parameter-efficient fine-tuning approaches and the Aurora system (Vanderfeesten et al., 2022). Our investigation also covers the impact of dataset imbalance on classification accuracy and evaluates the potential of dataset augmentation strategies to mitigate these challenges.

2 Background and Related Work

SDG Classification and Aurora System: SDG classification addresses aligning scientific literature with the United Nations’ Sustainable Development Goals (SDGs). Traditional methods have varied from rule-based systems, highlighting the need for manual keyword refinement (Rivest et al., 2021; Wang et al., 2023), to advanced transfer-based machine learning approaches. Among these, the Aurora system stands out for employing the multilingual BERT (mBERT) for SDG classification of scientific publications (Vanderfeesten et al., 2022). Aurora offers a binary classifier for each SDG, therefore supporting multilabel classification by setting a threshold.

LLMs and Instruction Tuning: Large Language Models (LLMs), such as GPT and BERT, have significantly advanced the field of NLP, offering deep contextual understanding and generation capabilities through extensive pretraining and fine-tuning processes (Radford et al., 2018; Devlin

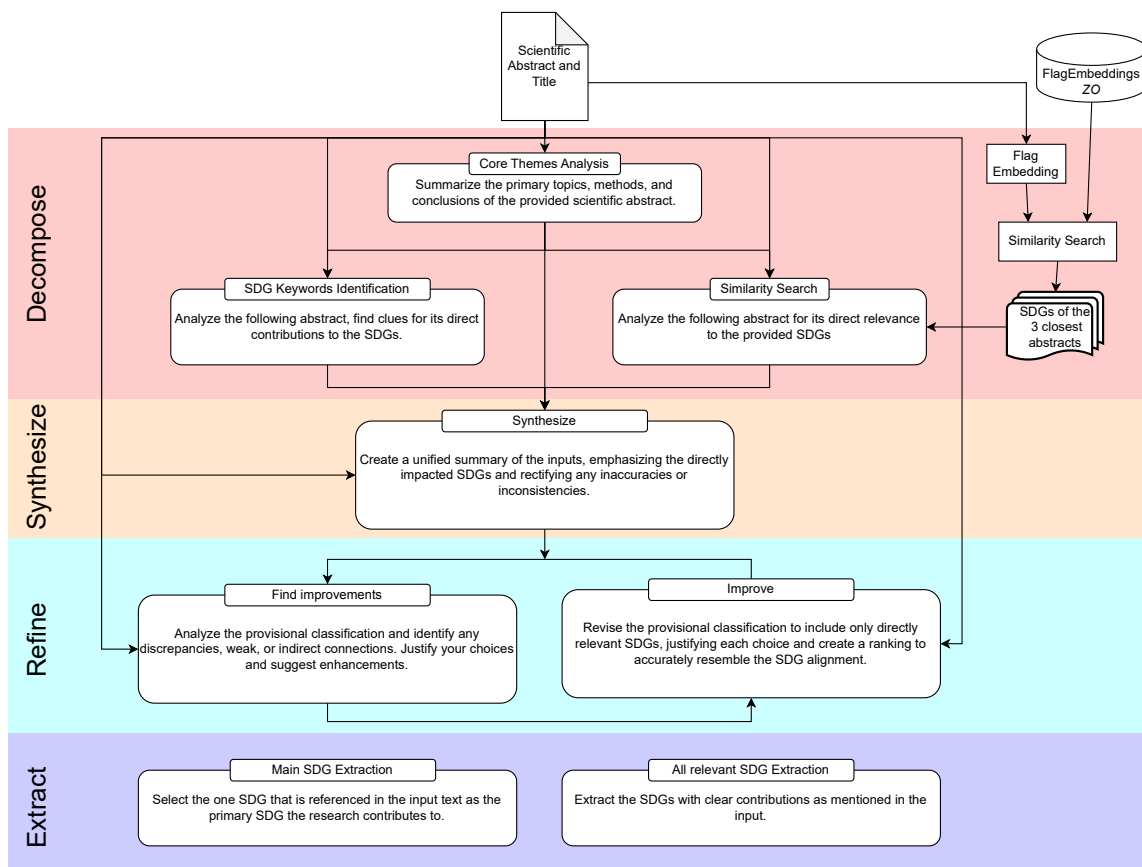


Figure 1: The DSRE prompt decomposition architecture using task decomposition (including semantic similarity), synthesis of results, self-refinement, and final SDG extraction.

et al., 2018). Instruction-tuned LLMs, leveraging datasets of instruction-output pairs, have shown versatility in adapting to specific tasks, without extensive task-specific training (Zhang et al., 2023).

Prompt Engineering techniques have emerged as critical for eliciting desired outputs from instruction-tuned models. These techniques range from manual and automated prompting to advanced methods like Chain-of-Thought (CoT) prompting, which guides models through intermediate reasoning steps (Wei et al., 2023), and In-Context-Learning (ICL) for task adaptation with minimal examples (Liu et al., 2023b).

Advanced Prompting Techniques: CARP, introduced by (Sun et al., 2023), directs the LLM to identify keywords and tones within the text. These are then assessed in a subsequent reasoning step, increasing text classification accuracy.

BSM, developed by (Saha et al., 2023), tackles complex problems by breaking them down into manageable subtasks. Each subtask is independently solved and the results are then merged to produce a coherent output. This method has shown to align LLM outputs with human-like reasoning.

Self-Refinement, proposed by (Madaan et al., 2023), enhances LLM outputs through an iterative process of feedback, and refinement. It continuously refines its responses based on self-generated feedback.

Parameter-Efficient Fine-tuning (PEFT): To address the computational demands of LLMs, techniques like Low Rank Adaptation (LoRA) and Quantized LoRA (QLoRA) have been proposed. LoRA minimizes the number of parameters needed for fine-tuning by introducing low-rank matrices to the attention heads (Hu et al., 2021). QLoRA extends this by applying quantization, significantly reducing the computational and memory requirements, making fine-tuning feasible on consumer hardware (Dettmers et al., 2023).

Quantization methods enhance the efficiency of Large Language Models (LLMs) by reducing the precision of model weights from 32-bit floating points to lower precisions such as 4-bit, which decreases model size and computational demands while minimizing performance loss (Frantar et al., 2022). For the quantization mapping, *quantization constants* are derived from the values of each ten-

Model	Qu	A/R \uparrow	P	F1	RT	Mem
Zephyr 7B beta	-	.608	.693	.615	41.9	9.53
Zephyr 7B beta	Q	.570	.659	.570	1.1	8.73
Zephyr 7B beta	DQ	.551	.644	.547	1.2	8.61

Table 1: Performance comparison highlighting the effects of quantization (Qu) in zero-shot classification. No quantization (-), quantized (Q), and double quantized (DQ) models are compared. Accuracy (A), Recall (R), Precision (P), F1 Score (F1), Runtime in seconds (RT), and Memory usage in Gigabytes (Mem) are reported.

or (Dettmers et al., 2023). These constants are used for restoring the weights to a higher precision, such as 16-bit, during inference and back-propagation in fine-tuning. Storing weights in 4-bit precision while performing computations in 16-bit during inference helps maintain numerical stability and reduces cumulative errors. This preserves the fidelity of the neural network operations even if the initial precision loss has already occurred. Despite the reduced precision, quantized models maintain essential functionalities such as contextual understanding and reasoning, as shown by Liu et al. (2023a).

3 Experiments

In our experiments, we test how the techniques described above influence SDG classification quality.

3.1 Data and Datasets

We use the imbalanced ZORA dataset Z (Meister, 2022), consisting of 384 scientific abstracts (270 train, 114 test items) from the Zurich Open Repository and Archive (ZORA). Each item belongs to a single SDG class (there is no class for not relevant to any SDG).

To compensate for our sparse in-domain data, we additionally exploit the crowdsourced OSDG Community Dataset (OSDG; UNDP IICPSD SDG AI Lab; PPMI), specifically selecting 26k texts (dataset label O) with good inter-annotator agreement (Roady, 2023). The dataset ZO simply concatenates Z and O , whereas $ZO Up$ provides exactly 56 samples for each SDG (except for SDG 17, which is not present in O).

Aurora uses the dataset A stemming from Scopus abstracts retrieved by search queries for “Mapping Research Output to the Sustainable Development Goals (SDGs)” v5.0.2.¹

¹<https://zenodo.org/records/4883250> (accessed on May 15, 2024)

Model	Qu	FT	A/R \uparrow	P	F1
Aurora	-	A	.500	.593	.500
LLAMA 2 7B	Q	-	.556	.626	.542
Zephyr 7B beta	Q	-	.570	.659	.570
DSRE	Q	-	.579	.687	.572
Zephyr 7B beta	-	-	.608	.693	.615
LLAMA 2 7B Chat	DQ	O	.653	.711	.647
LLAMA 2 7B Chat	DQ	Z	.657	.679	.653
LLAMA 2 7B Chat	DQ	ZO	.657	.709	.653
Zephyr 7B beta	DQ	Z	.685	.691	.679
LLAMA 2 7B Chat	DQ	ZO Up	.702	.710	.697
Zephyr 7B beta	DQ	ZO Up	.711	.754	.717

Table 2: Performance comparison on Z test set ($N=114$), highlighting the effects of quantization, prompt decomposition and fine-tuning. Column ‘Qu’ indicates whether no quantization (-), 4-bit quantization (Q) or double 4-bit quantization (DQ) was used. Column Fine-Tuning (FT) indicates which dataset was used to train the model, if any. The table is sorted by column Accuracy/Recall (A/R). We report mean values of weighted macro-averages per class from at least 2 runs.

3.2 Large Language Models

LLAMA 2 7B incorporates optimized multi-head attention mechanisms and employs strategies to enhance training efficiency (Touvron et al., 2023). It achieves new state-of-the-art results, with its 13B variant outperforming larger models like GPT-3 in Zero-Shot Common Sense Reasoning tasks. For our experiments, we use the smallest LLAMA 2 7B variant of this instruction-tuned LLM family.

LLAMA 2 7B Chat is an instruction-tuned chat variant trained using Reinforcement Learning with Human Feedback (RLHF) (Kaufmann et al., 2023). In our experiments, we always use the 4-bit double-quantized version for fine-tuning.²

Zephyr 7B beta is built by distillation techniques (Hinton et al., 2015) from the larger Mistral 7B teacher model (Tunstall et al., 2023). The so-called Distilled Direct Preference Optimization (dDPO) method adjusts the model’s outputs to favor higher-quality responses in a supervised manner, thereby simplifying the alignment process and minimizing the reliance on reinforcement learning techniques or reward models.³

3.3 Quantization Experiments

The results in Table 1 show the expected loss in classification performance with each additional

²<https://huggingface.co/meta-llama/Llama-2-7b-chat-hf> (accessed on May 15, 2024)

³<https://huggingface.co/HuggingFaceH4/zephyr-7b-beta> (accessed on May 15, 2024)

quantization level. However, if maximal performance is not key, runtime reduces by a factor of 40 with 4-bit quantization. Double quantization unfortunately reduces performance without tangible benefit in runtime or memory usage.

3.4 DSRE Experiments

The proposed Decompose-Synthesize-Refine-Extract (DSRE) approach aims to classify abstracts into SDGs by employing zero-shot learning via instruction-tuned LLMs (here always Zephyr 7B beta). This approach does not suffer from limited and imbalanced labeled data. Additionally, it can easily be formulated for multilabel predictions, which might be more adequate for scientific output that contributes to more than one goal.

DSRE is structured into four phases: Decomposition, Synthesis, Refinement, and Extraction. Figure 1 shows our architecture and the prompts.

The *Decomposition* breaks the classification task into smaller tasks and incorporates strategies akin to CARP, which enhance the model’s ability to identify textual clues (Sun et al., 2023). Inspired by the similarity search applied in CARP, DSRE uses Flag Embeddings (BGE) within its Similarity Search component. Flag Embeddings leverage the last hidden state of the [CLS] token for sentence embeddings, providing dense vector representations of texts. Employing BGE allows DSRE to conduct semantic comparisons between the target abstract and a pre-classified corpus (Xiao et al., 2023).

The *Synthesis* merges these detailed analyses into a preliminary classification of the abstract’s relevance to SDGs. This integration draws on methodologies related to BSM for synthesizing information from disparate sources (Saha et al., 2023).

Subsequently, the *Refinement* improves this preliminary classification by addressing inaccuracies and inconsistencies, drawing upon Self-Refinement techniques to improve the classification’s accuracy and reliability (Madaan et al., 2023).

Lastly, the *Extraction* isolates the SDG classifications from the refined output. In our case, it requires fine-tuning with a specialized dataset to accurately extract SDG labels from DSRE outputs, indicating a specific limitation of output control from current LLMs.

3.5 Results

Table 2 compares 8 system configurations, providing further insights on the effects of quantization,

advanced prompt engineering, and fine-tuning.

While our advanced DSRE prompting technique marginally improves classification accuracy within the group of quantized zero-shot approaches, it also introduces significant computational overhead (overall processing time per average abstract is around 46 seconds on an RTX 3080). Only exceptional classification performance would justify such an effort. However, the underlying Zephyr 7B beta is clearly not strong enough for zero-shot SDG classification reasoning. In the next section, we discuss the results of parameter-efficient fine-tuning experiments.

3.6 QLoRA Fine-Tuning Experiments

Both LLAMA 2 7B Chat and Zephyr 7B beta profit from fine-tuning. The lower part of Table 2 shows that a domain shift (and maybe also an annotation policy shift) from OSDG documents to scientific abstracts exists. For fine-tuning, carefully balancing classes by adding OSDG *O* material instead of concatenating all *O* data gives the best results. Parameter-efficient fine-tuning on in-domain training data overcomes the observed performance losses due to quantization by a large margin.

4 Conclusions

For the task of classifying short abstracts into Sustainable Development Goals (SDGs), like those typically encountered in ZORA, utilizing double-quantized Large Language Models (LLMs) via parameter-efficient fine-tuning methods proves to be a superior strategy. This approach notably excels in optimizing the use of computational resources and in reducing processing time. Furthermore, we expect our labelled data to grow within the ongoing project that this work is part of, and these models can immediately profit from this.

The DSRE method may offer an advantage when applied to longer full-text articles. The intermediary outputs generated by DSRE-like procedures could additionally serve to provide AI-generated explanations to end users, enhancing the interpretability of SDG classification decisions.

Future work will focus on deepening multilabel experiments that we already started on a small (N=51) dataset collected in a human annotation campaign using the output of the multilabel DSRE prompt as preannotations.

Acknowledgments This work was supported by the Digitalization Initiative of the Zurich Higher

Education Institutions (DIZH).

References

- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2022. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2023. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*.
- Peiyu Liu, Zikang Liu, Ze-Feng Gao, Dawei Gao, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2023a. Do emergent abilities exist in quantized large language models: An empirical study. *arXiv preprint arXiv:2307.08072*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhunoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- Robin Meister. 2022. Identifying and prototyping a machine learning based SDG-classification for scientific papers. Bachelor’s thesis, University of Zurich.
- OSDG; UNDP IICPSD SDG AI Lab; PPMI. 2023. [OSDG community dataset \(OSDG-CD\)](#).
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. [Improving language understanding by generative pre-training](#). electronic.
- Maxime Rivest, Yury Kashnitsky, Alexandre Bédard-Vallée, David Campbell, Paul Khayat, Isabelle Labrosse, Henrique Pinheiro, Simon Provençal, Guillaume Roberge, and Chris James. 2021. [Improving the Scopus and Aurora queries to identify research that supports the United Nations Sustainable Development Goals \(SDGs\) 2021](#). *Mendeley Data*, 2.
- Jessica Saemi Roady. 2023. Automatic classification of academic papers according to the UN sustainable development goals – an interdisciplinary perspective. Master’s thesis, University of Zurich.
- Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*.
- Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.
- United Nations: Department of Economic and Social Affairs. 2021. [The 17 goals: Sustainable development](#).
- Maurice Vanderfeesten, Robert Jaworek, and Lennart Keßler. 2022. [AI for mapping multi-lingual academic papers to the United Nations’ Sustainable Development Goals \(SDGs\)](#). *Zenodo*.
- Weiwei Wang, Weihao Kang, and Jingwen Mu. 2023. [Mapping research to the Sustainable Development Goals \(SDGs\)](#). *Research Square*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged resources to advance general Chinese embedding. *arXiv preprint arXiv:2309.07597*.
- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

A Appendix

A.1 QLoRA Configuration

```
BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)

LoraConfig(
    lora_alpha=16,
    lora_dropout=0.1,
    r=64,
    bias="none",
    task_type="CAUSAL_LM"
)

SFTConfig(
    per_device_train_batch_size=1,
    gradient_accumulation_steps=4,
    learning_rate=1e-4,
    logging_steps=10,
    num_training_epochs=1,
    bf16=True,
    optim="paged_adamw_8bit"
)
```

Listing 1: QLoRA configuration used for training

A.2 Fine-Tuning Prompts

The SDGs are:

- SDG 1 – No Poverty: Aims to end poverty in all its forms everywhere.
- SDG 2 – Zero Hunger: Aims to end hunger, achieve food security and improved nutrition, and promote sustainable agriculture.
- SDG 3 – Good Health and Well-being: Aims to ensure healthy lives and promote well-being for all at all ages.
- SDG 4 – Quality Education: Aims to ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.
- SDG 5 – Gender Equality: Aims to achieve gender equality and empower all women and girls.
- SDG 6 – Clean Water and Sanitation: Aims to ensure availability and sustainable management of water and sanitation for all.
- SDG 7 – Affordable and Clean Energy: Aims to ensure access to affordable, reliable, sustainable, and clean energy for all.
- SDG 8 – Decent Work and Economic Growth: Aims to promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all.
- SDG 9 – Industry, Innovation and Infrastructure: Aims to build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.
- SDG 10 – Reduced Inequalities: Aims to reduce inequality within and among countries.
- SDG 11 – Sustainable Cities and Communities: Aims to make cities and human settlements inclusive, safe, resilient, and sustainable.

- SDG 12 – Responsible Consumption and Production: Aims to ensure sustainable consumption and production patterns.
- SDG 13 – Climate Action: Aims to take urgent action to combat climate change and its impacts.
- SDG 14 – Life Below Water: Aims to conserve and sustainably use the oceans, seas, and marine resources for sustainable development.
- SDG 15 – Life on Land: Aims to protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss.
- SDG 16 – Peace and Justice Strong Institutions: Aims to promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels.
- SDG 17 – Partnerships for the Goals: Aims to strengthen the means of implementation and revitalize the global partnership for sustainable development.
- SDG 0 – No Goal: Must be chosen if and only if the abstract does not contribute to any of the SDGs.

Listing 2: The list of SDG descriptions provided to the model

A.2.1 Prompt for Training and Inference

System Message:

You are an expert in scientific research, policy analysis and sustainable development. Determine to which Sustainable Development Goal (SDG) the paper described by the abstract contributes the most. The SDGs are: {SDG_List} \n\nExpected Output Format: \n\nThe abstract primarily contributes to SDG [SDG Number] – [SDG Title]. \n\nExample Output: \n\nThe abstract primarily contributes to SDG 1 – No Poverty

User Message:

Analyze the abstract and determine to which Sustainable Development Goal (SDG) the paper described by the abstract contributes the most. \n\nAbstract and Title: \n\n {Abstract_Text} \n"

Listing 3: Prompt to train and infer the main SDG from an abstract

A.3 DSRE Prompts

General Constraints:

Always provide answers that are short, concise and to the point. Keep your answer within 300 words. Ensure clarity and consistency in your responses, avoiding contradictory statements or redundancy.

Listing 4: General Constraint provided to each Prompt, to keep the output short and concise.

A.3.1 Generation Prompt

Assess if the provided prompt meets the following criteria:

- Clarity and Specificity: Is the prompt clear and specific enough to guide the model's response accurately?
- Balancing Creativity and User Intent: Does the prompt find a balance between leveraging the model's creativity and achieving the user's intended outcome?
- Domain Adaptation: Is the prompt tailored to a specific domain or context to enhance relevance and accuracy?
- Handling Ambiguity: Does the prompt include strategies for effectively dealing with ambiguous inputs or situations?

Provided prompt:
"{System and user prompt}"

Listing 5: Prompt used to refine the prompts utilized in the DSRE using ChatGPT

A.3.2 Prompts for SDG Classification

System Message:

You are an AI expert in scientific abstract analysis. {general_constraint}

User Message:

Summarize the primary topics, methods, and conclusions of the provided scientific abstract. Abstract:\n"{Abstract_Text}"

Listing 6: Prompt utilized to extract core themes from the abstract

System Message:

You are an AI expert trained across multiple disciplines relevant to the Sustainable Development Goals (SDGs), including environmental, social, economic, cultural, and technological fields. {general_constraint} {sdg_description}

User Message:

Given the identified core themes:
"{Core_Themes}"

Analyze the following abstract for its direct relevance to the SDGs. Scientific Abstract and Title:
"{Abstract_Text}"

Listing 7: Prompt for analyzing abstracts for SDG relevance based on core themes

System Message:

You are an AI trained across multiple disciplines relevant to the Sustainable Development Goals (SDGs), including environmental, social, economic, cultural, technological fields, and scientific analysis. Critically and objectively analyze texts for their contribution to Sustainable Development Goals (SDGs). Summarize the main goal of the provided SDG and assess whether the abstract directly contributes towards achieving this SDG. Justify instances where the abstract does not directly contribute to the SDG. {general_constraint}

User Message:

Given the identified core themes:
"{Core_Themes}"

Analyze the following scientific abstract for its direct relevance to the SDGs "{closest_abstract_sdgs}, {closest_sdg_centroid}". Scientific Abstract and Title:
"{Abstract_Text}"

Listing 8: Prompt for detailed SDG relevance analysis of scientific abstracts

System Message:

You are an AI specialized in synthesizing and summarizing complex texts related to the same scientific abstract. Your task is to merge various inputs into a single, coherent summary. Focus specifically on the directly impacted Sustainable Development Goals (SDGs). Exclude irrelevant or indirectly affected SDGs, and correct any inconsistencies or inaccuracies in the information. All inputs provided are aspects of the same scientific abstract. {general_constraint}

User Message:

Here are the texts to be summarized, all pertaining to the same scientific abstract:

1. Scientific Abstract and Title:
"{Abstract_Text}"

2. Core Themes of the Scientific Abstract:
"{Core_Themes}"

3. SDG Concepts of the Scientific Abstract:
"{SDG_Concepts}"

4. Additional Information about the Scientific Abstract:
"{Retrieval}"

Create a unified summary of these inputs, emphasizing the directly impacted SDGs and rectifying any inaccuracies or inconsistencies.

Listing 9: Prompt for merging inputs into a coherent summary with a focus on SDGs

System Message:

You are an AI trained in critical analysis and sustainable development goals (SDGs). Your task is to critically analyze the provided abstract against the provisional classification for alignment with relevant SDGs. Highlight any discrepancies, weak, or indirect connections. {general_constraint} Refer to the provided SDG descriptions for accurate comparisons.

{sdg_description}

{general_constraint}

User Message:

Scientific Abstract and Title:
"{Abstract_Text}"

Provisional Classification:
"{Response_Text}"

Analyze the provisional classification and identify any discrepancies, weak, or indirect connections. Justify your choices and provide enhancements.

Listing 10: Prompt for finding improvement in the provisional classification

System Message:

You are an AI tasked with optimizing classifications related to Sustainable Development Goals (SDGs). Revise the provided provisional classification by addressing the identified enhancements. Ensure that your revised classification only includes directly relevant SDGs and clearly justifies the direct relevance of each selected SDG. Adjust the rankings and explanations to reflect a more accurate alignment with the SDGs.

{general_constraint}

User Message:

Scientific Abstract and Title:
"{Abstract_Text}"

Provisional Classification:
"{Defective_Response_Text}"

Identified Enhancements:
"{Identified_Issues_Text}"

Revise the provisional classification to include only directly relevant SDGs, justifying each choice and create a ranking to accurately resemble the SDG alignment.

Listing 11: Prompt for optimizing classifications related to SDGs

System Message:

As a precise AI, your specific function is to identify the single, most pertinent Sustainable Development Goal (SDG) from those mentioned in the input in relation to the abstract. Select the one SDG that is referenced in the input text as the primary SDG the research contributes to based on the content of the abstract. Your response must be formatted as 'SDG X', where 'X' is the number of the most relevant SDG from those mentioned in the input. Ensure your response strictly adheres to this format and excludes any justification or additional information.

User Message:

Input:
"{BSM_Response}"

Listing 12: Prompt for identifying the primary SDG from input

System Message:

Analyze the provided input to identify which Sustainable Development Goals (SDGs) are directly impacted by the scientific abstract. Extract only the SDGs with

clear contributions as mentioned in the input. Present your findings as a list of SDG numbers, separated by commas. Your response should be only this list, without additional justification or explanation.

User Message:

Input:
"{BSM_Response}"

Listing 13: Prompt for identifying all directly impacted SDGs from input

A.4 Extraction Module Fine-Tuning

A.4.1 Example Training Input-Output Pair

Input Text:

Ranked List of SDG Contributions:

1. SDG 15 – Life on Land: This study's main contribution is toward SDG 15, which aims to protect terrestrial ecosystems and promote sustainable use of resources. Accurately estimating population trends for amphibian species is crucial for understanding which species are declining and which require conservation action. By accounting for imperfect detection, this study helps identify species at risk of decline and promotes conservation efforts to protect their habitats.
2. SDG 17 – Partnerships for the Goals: The study emphasizes the importance of collaboration and data sharing, highlighting the need for partnerships to inform conservation policy and manage species declines. By advocating for the routine recording of nonobservation and absence data, the study underscores the significance of partnerships and collaborative efforts for conservation purposes.
3. SDG 13 – Climate Action: While not directly related to climate action, this study's findings regarding species declines could have implications for understanding the impacts of climate change on amphibian populations. Accurate population trend estimates are necessary for understanding which species are declining and which are most in need of conservation action, and this study's insights could inform climate adaptation and mitigation strategies for amphibian species.

The study does not contribute significantly to the other SDGs and should not be ranked for them. If the abstract did not contribute to any of the SDGs, SDG 0 – No Goal – would be assigned.

In conclusion, this study's contributions are primarily aligned with SDG 15 and SDG 17, emphasizing the significance of protecting terrestrial ecosystems and promoting partnerships and collaboration for conservation purposes. Its insights also have indirect implications for SDG 13, highlighting the need for accurate population trend estimates to understand the impacts of climate change on amphibian populations.

Primary SDG Extraction Output: SDG 15
Secondary SDGs Extraction Output: 15, 17, 13

Listing 14: Input-output pairs for fine-tuning the DSRE extraction Module

A.5 Artifacts Availability

The artifacts associated with this research can be found in the repository on

<https://github.com/TobiFank/SDG-Classification-Using-Instruction-Tuned-LLMs/blob/master/README.md>

Triple Detection in German Verb-based Sentiment Inference: The Case of Novel Verbs

Dylan Massey

Department of Computational Linguistics, University of Zurich
Andreasstrasse 15, 8050 Zürich
dylan.massey@uzh.ch

Abstract

This short paper describes the evaluation of two neural models for their ability to detect sources, targets and verbal predicates as a step for enabling the full sentiment inference task, that is: Identifying whether a polar relation (*in-favour-of*, *against*) holds between two entities in a given sentence and which verb mediated the relation. The models are trained and evaluated on a silver standard generated by a rule-based system for sentiment inference on German text. We are mainly interested in the research question whether neural models are able to generalize to novel, previously unseen verb constellations and, thus, might make rule-based approaches superfluous. We find that one examined architecture, a simple transformers-based approach, achieves an F1-score of 85.2% on full triple detection.

1 Introduction

The central task in sentiment inference is to identify the proponents and opponents given a particular text. Polar verbs play a crucial role here, they identify the polar relation that holds between a source and target, which are both realized through particular semantic roles of the verb. In *The police man killed the aggressor with his weapon* there is an *against* relation: the police man as the source acts against the aggressor being the target. The presence of such a relation might be represented as a label (*against*) over the triple (*police*, *kill*, *aggressor*). However, not every instantiation of a verb should be interpreted that way. Take *The knife killed the aggressor*, an instrument-subject verb alternation. Here *knife* is not the source of an *against* relation, because it is not an actor. Adequate extraction of polar relations from sentences requires the consideration of selectional restrictions of verb frames identifying *knife* as inanimate.

In this paper we investigate whether a customized neural model, directly trained on a silver

standard of triples is able to solve this task: finding the source, verb, and target. An exemplary triple is (X, loves, Y), which *might* indicate a favourable attitude of X towards Y. The source is denoted by X and the target by Y. The source is the origin of the (potentially inferred) sentiment described and the target the one towards whom the sentiment is directed. We are especially interested in the question whether the two investigated neural approaches and their underlying models generalise to verbs not seen in the training phase. Since for German no gold standard is available a silver standard was generated by a rule-based system (Klenner et al., 2017) whose precision is reasonable, but which is expected to have low recall due to lexical gaps.

2 Related Work

Sentiment Analysis is concerned with the elicitation of affective signal in text. While text types such as product reviews contain highly opinionated expressions, are short and profit from assignment of a polarity value in a range $[-1, 1]$, other text types such as newspaper articles, can benefit from more granular analysis. Approaches concerned with more granular, intra-textual elicitation of sentiments have been termed fine-grained sentiment analysis, aspect-based sentiment analysis, or more recently sentiment inference and structured sentiment analysis¹.

With the performance of approaches subsuming transformer-based contextual word embeddings (Vaswani et al., 2017; Devlin et al., 2019) tasks related to the elicitation of fine-grained sentiment in text have profited too. A commonly found distinction made by approaches is the separation between entity recognition (ER) and subsequent classification of which relation holds between them. The former has given rise to the subtask of Opinion Role Labelling (ORL). ORL has the goal to

¹For structured sentiment analysis cf. Barnes et al. (2021)

identify source and target given a polar sentence (Bamberg et al., 2022, p. 112). Previous work has also considered the more extensive task of additionally identifying a cue expression inducing sentiments between or towards entities (Yang and Cardie, 2013; Choi and Wiebe, 2014; Katiyar and Cardie, 2016). In the ORL-only setting and more recently in German, Bamberg et al. (2022) achieve state-of-the-art performance on the IGGSA-Steps datasets (Ruppenhofer and Struss, 2016), they too, use a transformers-based approach. Deng and Wiebe (2015) are the first to present a fine-grained corpus of texts for sentiment analysis. Reschke and Anand (2011) introduce the idea of verbs for an implicit sentiment analysis in English.

3 Rule-based Silver Standard

The rule-based system described in Klenner et al. (2017) uses a verb lexicon² for sentiment inference. For the instantiation of a verb, various restrictions must be satisfied. Table 1 shows one polar frame for the German verb *sorgen für* (care for).

1 dependency label	subj	pp-obj
2 lexical restriction	-	prep=für
3 selectional restriction	+animate	+animate
4 polar role	source	target
5 polar relation	in favour	-

Table 1: Frame of *sorgen für* (Eng. care for)

For this reading the restrictions 1, 2 and 3 must hold: particular dependency labels, lexical restrictions (preposition *für*, Eng. *for*) and animacy requirements must be met. Then the polar assignments (4 and 5) can be made, namely that the subject is the source and the object the target of a in favour relation. A dependency parser (Sennrich et al., 2009) and an animacy classifier (Klenner and Göhring, 2022) are used to enforce those restrictions. Due to the restrictive verb instantiation policy, precision of the system is reasonable. We manually evaluated the output of 210 sentences and found a precision of 79.0%, a recall of 78.9%. However lexical gaps (missing verbs) and not modelled polar versions of a verb are expected to affect recall. Here neural models come into play, which might be able to generalise to unseen verbs. However, these models not only should learn applicable

²The lexicon is available from <https://www.c1.uzh.ch/en/texttechnologies/research/opinionmining/sentiment-inference.html>

verbal profiles, but also the restrictions that guide the instantiations (e.g. selectional restrictions) of source and target. For this, a large gold standard is needed. Since no such gold standard for German is available, we propose to create a silver standard on the basis of the output of Klenner et al. (2017), which - as we have argued - has proved to have a reasonable performance. We used the demo system³ of Klenner et al. (2017) to produce a silver standard.

The data which is used to generate the silver standard dataset is from the Swiss Media Database (SMD)⁴. In total 266,647 news articles from major Swiss news outlets within the date range from January 2018 until November 2022 were downloaded, cleaned and passed through the rule-based sentiment inference system. We omit sentences which do not include both, a source and a target and where either or both are pronouns. Both settings would require means of resolution over text surpassing the sentence-level, which is outside the scope of our work. Along with the aforementioned source, target, verb triples, the rule-based system also generates a label that signifies a positive or negative relation between the source and target depending on the constellation between the three entities. We also include an equal amount of "neutral" triples all of which contain verbs however that can potentially be charged and are thus part of the verb lexicon. Including triples that contain polar verbs in neutral constellations can help reduce detection of false positives and therefore lead to a more adequate performance in sentiment inference and analysis systems (Wilson, 2008, p. 181). Although these triples are not per se triples that induce a polar relation and we cannot talk of a source and target in such instances, we keep the terms for simplicity.

4 Neural Models for Polar Triple Detection

We have so far discussed the task of sentiment inference and the importance of identifying the verbal predicate and the roles the predicate casts on its participants as a first step. Contrary to lexicon-based systems, neural approaches handle out-of-vocabulary words at inference and might cope well with unseen verb constellations. For example, if *X loves Y* is within the vocabulary of the lexicon,

³The demo is available under <https://pub.c1.uzh.ch/demo/stancer/index.py>

⁴<https://smd.ch/en/home>

then we might infer that X is positive towards Y , given $X := \text{subj}$ and $Y := \text{obj}$. Similarly, X *adores* Y might not trigger if *adore* is not in the lexicon.

Our interest lies on the polar relations that are verb-mediated and span **between** two proper textual entities. The goal of our neural task is to (a) detect entities (verb and its fillers) on the sentence-level and (b) how the found non-verbal entities relate to each other. More precisely, our goal is to detect all triples ρ from some sentence x where ρ consists of a source s , a verbal predicate v and a target t . In the present paper **we shall only be concerned with step (a), that is the detection of the verbal predicate and its fillers**, since initial experiments revealed that the downstream performance of (b) is greatly affected by the ability to extract salient triples from the sentence (especially the verb).

5 Method

We consider two systems for sentiment inference. System 1 (S_1) is devised by [Zhong and Chen \(2021\)](#) who follow a two-step approach for the task of relation extraction (RE), which we will repurpose for sentiment inference and abbreviate as ERRE⁵. They train entity model and relation classifier independently of each other. The entity-marked sentences serve as input to the relation model. We base our implementation closely on the tutorial provided by [Pal \(2022\)](#), who references [Zhong and Chen \(2021\)](#), but modify the approach to handle ternary relations aligning our task. Both approaches use XLM-R by [Conneau et al. \(2020\)](#) as pre-trained base. For the relation model of the ERRE we use the pre-trained bert-base-german-cased, which has shown performant in German language settings. System 2 (S_2) is proposed by [Samuel et al. \(2022\)](#) for structured sentiment analysis, where the goal is to extract and polarly relate subjectivity cues to sources, targets. We abbreviate System 2 as PERIN.

Initial experiments revealed that final task performance is affected most dramatically by ER performance relative to downstream RE. Thus, the focus of the present paper is on ER performance. Since ERRE relies on two independent models, ER performance can be easily evaluated. Its entity model is based on `AutoModelForTokenClassification` ([Wolf et al., 2020](#)). For PERIN it is not possible to fully decouple entity recognition capabilities

⁵Short for Entity Recognition and Relation Extraction

from the full task since training occurs end-to-end. Therefore we train and on the full dataset (including on in-favour and against relations) but evaluate only ER performance.

To test the generalization capabilities of our proposed neural models we split test, validation and training data twice: once randomly (RAND) and once based on the restriction that all triples that have some verb v can only either all be in the training dataset or in the test dataset (MEVG⁶). The MEVG setting simulates the worst-case scenario where all verbs are novel at inference and reveals models generalisation capabilities to unseen contexts.

RAND contains 28,072 train, 6,004 validation and 6,082 test sentences. Label frequencies were balanced using a disproportionately stratified sampling meaning that we have equal representation of every class - in favour, against, neutral - in our dataset. RAND consists of 460 unique verbs in total. MEVG contains 30,032 training and 5,063 validation and testing sentences. The training dataset contains 333 unique verbs and validation and testing (where we allow overlap) contain 138 unique verbs.

As evaluation metrics, we rely on [Barnes et al. \(2022\)](#), who evaluate the performance on the F1-score, precision and accuracy for each respective element of the n -tuple as well full tuple precision and accuracy. The precision and recall are provided in the appendix [A.2](#). Hyperparameters are addressed in appendix [A.1](#).

6 Results

We evaluate both approaches on their ability to correctly identify the verb and potential sources and targets and conclude that the entity model of the ERRE approach performs best on our generated silver standard.

Table 2 illustrates the F1-scores for the individual entity extraction scores and the performance for the combined triple extraction, $F1_{\rho}$, on the test dataset. The performance drops from RAND to the MEVG setting where only verbs not part of the training dataset are in the test dataset. The drop is higher for PERIN (S_2) than for ERRE (S_1) and it is more drastic for ERRE in verb ($F1_V$) (15.1%) and target detection ($F1_T$) (15.3%) than it is for source detection ($F1_S$) (7.9%). The decrease in (whole) triple classification performance ($F1_{\rho}$) is

⁶Mutually exclusive verbs groups splitting

System	Split	F1 _s	F1 _t	F1 _v	F1 _ρ
S_1	RAND	90.7	88.7	96.8	85.2
S_2	RAND	89.5	85.8	95.9	83.9
S_1	MEVG	82.8	73.4	81.7	65.0
S_2	MEVG	47.3	42.1	47.0	41.8

Table 2: Triple recognition capabilities for source (F1_s), target (F1_t) and verb (F1_v) of S_1 (ERRE) and S_2 (PERIN) on the test dataset in % depending on whether verbs in the test dataset were randomly overlap (RAND) or were mutually exclusive to the training dataset (MEVG).

19.8% for ERRE (from 85.2% to 65%) and 42.1% (from 83.9% to 41.8%) for PERIN. PERIN performs worse than ERRE under unseen constellations. Unexpectedly performance is not only low on verb detection, but similar also on source and target detection performance. ERRE, on the other hand, not only has still (under MEVG) relatively better performance in verb detection (81.7%), but also a triple F1 score of 65%. The loss in triple classification from entity recognition appears rather attributable to the 12.4% drop in target classification (from 85.8% to 73.4%) rather than verb identification.

In order to get a better understanding of the quality of the silver standard and the reproductive power of the neural models, a randomly sampled set of 210 sentences from the silver standard were manually annotated by a single annotator using the Universal Data Tool by Ibarluzea (2021). This still can not be regarded as an analytic gold standard, since these 210 sentences were selected on the basis that at least one verb of the rule-based system’s lexicon was present (100% sentence recall). However, since the objective of the current paper is to evaluate the verb generalisation power of neural models, we argue that this is a reasonable initial setting (we could call it a verb-biased gold standard **gold*) for the sake of investigation. Since the verb was pre-supposed (and shown to the annotator), we excluded it from the evaluation. We only carried out opinion source identification, i.e. how accurate the silver standard is in terms of identification of potential sources and targets. Results are visible in Table 3.

Knowing the limitations (we only have **gold*) we nonetheless can say that the silver standard ap-

System	F1 _s	F1 _t	F1 _{s,t}
S_0	87.3	85.9	78.9
S_1	88.3	84.3	77.2
S_2	88.7	84.5	77.1

Table 3: Comparison of the silver standard (S_0), the ERRE (S_1) and PERIN (S_2) models with manually annotated sentences (**gold*). All numbers are in %.

pears to satisfy performance wrt. opinion role labelling and acts as viable resource for training neural models. The manual annotation of S_0 revealed a F1_s of 87.3%, a F1_t of 85.9% and a F1_{s,t} of 78.9% (F1_{s,t} indicates pairs of source and target). Both neural approaches reproduced these results: Trained on the silver standard, they reach the same performance wrt. to **gold* (a real gold standard, though verb-biased) as S_0 . For more precise results including the detailed precision and recall for all the individual and the combined components we refer to the appendix in section A.2. Nevertheless, the results are - due to the restriction that only the opinion roles of the given verbs are considered - too high and cannot be compared to full opinion role detection as e.g. done in Bamberg et al. (2022). There the results are 10 to 15% lower.

7 Conclusion

The research hypothesis of this short paper was that neural models are able to deal with novel and unseen verbs not encountered during training in the context of sentiment inference. This is crucial where the polar verb directly mediates a polar relation (in favour, against) if used in an affirmative, factual sentence. On the basis of a rule-based system, a silver standard was generated for the training and evaluation of two neural models. The empirical settings comprised a worst-case scenario where the verbs of the training and test set are mutually exclusive. In this setting, the performance of one of the systems, though decreased, still was reasonably good showing that generalization at the verb-level has taken place. This learned neural model is as good as the rule-based system on a small gold evaluation, but can also deal with novel cases which the rule-based system under no conditions could achieve. The rule-based system, thus, is superfluous. With increasingly powerful large language models we devise as future work to inves-

tigate prompting techniques as data augmentation strategy for our current models, as well as a direct approach using large language models.

References

- Laura Bamberg, Ines Rehbein, and Simone Ponzetto. 2022. [Improved Opinion Role Labelling in Parliamentary Debates](#). In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pages 110–120, Potsdam, Germany. KONVENS 2022 Organizers.
- Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. [Structured Sentiment Analysis as Dependency Graph Parsing](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3387–3402, Online. Association for Computational Linguistics.
- Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. [SemEval 2022 Task 10: Structured Sentiment Analysis](#). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1280–1295, Seattle, United States. Association for Computational Linguistics.
- Yoonjung Choi and Janyce Wiebe. 2014. [+/-effectwordnet: Sense-level lexicon acquisition for opinion inference](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, SIGDAT*, pages 1181–1191.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.
- Lingjia Deng and Janyce Wiebe. 2015. [MPQA 3.0: An entity/event-level sentiment corpus](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1323–1328, Denver, Colorado. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Severin Ibarluzea. 2021. [Universal Data Tool](#).
- Arzoo Katiyar and Claire Cardie. 2016. [Investigating LSTMs for Joint Extraction of Opinion Entities and Relations](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.
- Manfred Klenner and Anne Göhring. 2022. [Animacy denoting german nouns: Annotation and classification](#). In *Proceedings of the Language Resources and Evaluation Conference*, pages 1360–1364, Marseille, France. European Language Resources Association (ELRA).
- Manfred Klenner, Don Tuggener, and Simon Clematide. 2017. [Stance detection in Facebook posts of a German right-wing party](#). In *LSDSem 2017/LSD-Sem Linking Models of Lexical, Sentential and Discourse-level Semantics*.
- Sujit Pal. 2022. [Transformer Based Approaches to Named Entity Recognition \(NER\) and Relationship Extraction \(RE\)](#).
- Kevin Reschke and Pranav Anand. 2011. [Extracting contextual evaluativity](#). In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 370–374.
- Josef Ruppenhofer and Julia Maria Struss. 2016. [IGGSA-STEPS: Shared Task on Source and Target Extraction from Political Speeches](#). 29(1):33 – 46. Place: Regensburg Publisher: Gesellschaft für Sprachtechnologie und Computerlinguistik.
- David Samuel, Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2022. [Direct parsing to sentiment graphs](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 470–478. Association for Computational Linguistics.
- Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. [A new hybrid dependency parser for German](#). In *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 115–124.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *Advances in neural information processing systems*, 30.
- Theresa Ann Wilson. 2008. [Fine-grained Subjectivity and Sentiment Analysis: Recognizing the intensity, polarity, and attitudes of private states](#). Doctoral Dissertation, University of Pittsburgh. (Unpublished).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-Art Natural Language Processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2013. [Joint Inference for Fine-grained Opinion Extraction](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. [A frustratingly easy approach for entity and relation extraction](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 50–61. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters

For the entity model part of ERRE we rely on sensible defaults. This amounts to a batch size of 16 for both training and development/test sets, 3 epochs, a learning rate of 2×10^{-5} and a weight decay of .01. Cross-entropy is used as a loss function and Adam serves as the optimizer.

For PERIN we use AdamW as the optimizer, a linear scheduler (without warmup). A weight decay of 1×10^{-2} is set, while the learning rate is set to 2×10^{-2} . The model is trained again for 3 epochs. Cross-entropy is the loss function.

A.2 Performance

Full performances for the silver standard test dataset in comparison to predictions of PERIN:

Source Precision: 0.932
Source Recall: 0.861
Source F1: 0.895
Target Precision: 0.901
Target Recall: 0.819
Target F1: 0.858
Verb Precision: 0.959
Verb Recall: 0.959
Verb F1: 0.959
Tuple Precision: 0.836
Tuple Recall: 0.842
Tuple F1: 0.839

Full performances for the manual annotations in comparison to the predictions of ERRE:

Source Precision: 0.895
Source Recall: 0.920
Source F1: 0.907
Target Precision: 0.903
Target Recall: 0.871
Target F1: 0.887
Verb Precision: 0.969
Verb Recall: 0.966
Verb F1: 0.968
Tuple Precision: 0.853
Tuple Recall: 0.851
Tuple F1: 0.852

Full performances for the manual annotations in comparison to the silver standard:

Source Precision: 0.896
Source Recall: 0.851
Source F1: 0.873
Target Precision: 0.857
Target Recall: 0.861
Target F1: 0.859
Tuple Precision: 0.790
Tuple Recall: 0.789
Tuple F1: 0.789

Full performances for the manual annotations in comparison to the entity model of the ERRE system:

Source Precision: 0.877
Source Recall: 0.889
Source F1: 0.883
Target Precision: 0.833
Target Recall: 0.854
Target F1: 0.843
Tuple Precision: 0.771
Tuple Recall: 0.774
Tuple F1: 0.772

Full performances for the manual annotations in comparison to the entity model of the PERIN system:

Source Precision: 0.877
Source Recall: 0.898
Source F1: 0.887
Target Precision: 0.837
Target Recall: 0.854
Target F1: 0.845
Tuple Precision: 0.771
Tuple Recall: 0.771
Tuple F1: 0.771

Chapter 3

Applied Track

This track introduces abstracts that showcase applied natural language processing (NLP) solutions from companies, developers and researchers. The purpose of this track is to provide a platform for demonstrating NLP solutions and facilitating knowledge sharing among the audience.

Adesso Intelligent Agent - An Addon Ecosystem for Extending and Empowering Rag Systems

Leon Stolpmann and Rafael Dubach and Stephan Symons and Philipp Kuntschik
leon.stolpmann@adesso.ch

Abstract

Retrieval Augmented Generation (RAG) is a technique that combines natural language interaction with a search and retrieval system connected to a data repository such as enterprise content management systems (CMS) [1]. RAG provides its users with a natural language interface for querying, accessing, and manipulating data contents, such as asking questions, summarizing documents, or classifying text. Leveraging the recent advances in Large Language Models (LLMs) and neural/vector search solutions, RAG for enterprise search has emerged as a key industrial use case. However, RAG also introduces several new challenges and limitations such as verification of truthfulness of given answers (e.g. LLMs tend to “hallucinate” at times [2]), content safety and potential reputational damage caused by harmful answers (in case of public or customer facing systems) and resulting limited trust in these systems. There are also legal and legislative challenges that need to be addressed, as unsafe content [3] or data privacy issues (e.g. PII [4]) are currently still legally ambiguous. To address these challenges, we present an AI framework we call adesso Intelligent Agent that can be adapted to specific industry use cases and provides add-on capabilities to individually address these issues. The adesso Intelligent Agent describes a scalable and customizable RAG-Framework that accelerates development and deployment of functioning systems while allowing to rapidly address current and future challenges such as privacy, compliance, and trustworthiness or the expanding needs from business or environment. Our reference architecture encompasses the ability for deployment on the common cloud providers as well as in a private data center. With this submission, we want to set the focus on our Addon Integration Layer. The layer sits between an outward facing API for Web-interfaces or to be used from other software and the RAG CORE-System and allows for observation, management, and Controlled Intervention of requests to the system. The adesso Addon Library describes a comprehensive set of mechanisms that proved effective working with our clients and allow us to quickly incorporate additional functionality to a RAG system while minimizing the impact on system performance. We will show, how our approach allows to continuously monitor, evaluate, and improve the quality and safety of the generated answers. We believe that RAG has great potential for a wide range of industry specific use cases but also poses significant challenges. By leveraging the best practices contained in the presented framework, companies are able to address these challenges and requirements for compliance in AI systems.

An LLM-based Chatbot for Joint Access to Customer and Corporate Data

Teo Ferrari and **Vincent Coriou** and **Rastislav Kovac**
Vuk Vuković and **Daniel Dobos** and **Fatemeh Borran** and **Andrei Popescu-Belis**
teo.ferrari@heig-vd.ch

Abstract

Large Language Models (LLMs) give access to huge amounts of public-domain knowledge and have robust interactive capabilities. When it comes to corporate or personal data, several techniques enable their integration with LLMs: in-context learning, fine-tuning, or retrieval-augmented generation (RAG). In the present work, we demonstrate that two of these techniques can be combined, and design a system that can answer questions about company-level and user-level data at the same time. Specifically, we present the design and evaluation of a customer support system which combines a fine-tuned version of an open-source LLM, for answering questions related to static company-level data, with in-context learning for answering questions about dynamic customer-specific data. This system has been designed with a Swisscom customer support use-case in mind. To inject static knowledge into the LLM, we employed a Parameter Efficient Fine-Tuning (PEFT) technique, specifically Low-Rank Adaptation (LoRA) (Hu, et al. 2021). This is cost-effective and at the same time has low overfitting risks. The static data used for fine-tuning contains company-specific knowledge formulated as question-answer pairs. To enable the LLM to access dynamic, customer-specific data, based on previous studies (White, et al. 2023) and our own experiments, we engineered a prompt which combines instructions concerning the desired behaviour of the chatbot with a structured representation of customer-related information (here, mostly about billing). The evaluation data includes several dozen questions about static and dynamic knowledge, with the system's answers being assessed along three criteria used to evaluate free-form question answering (Sai et al. 2022): relevance, correctness, and fluency. In other words, we assess if the answer is on the same topic and provides the required type of information; then, if the answer is factually correct given the knowledge base (irrespective of its relevance to the question); and finally, if the answer is formulated in correct English and is appropriate in terms of politeness and greetings. Following preliminary assessments of several LLMs, we selected Mistral 7b for our implementation. The evaluation results revealed that fine-tuning a Mistral (Jiang, Sablayrolles, Mensch, et al. 2023) model enabled it to handle static data queries satisfactorily, while prompt engineering ensured effective access to dynamic data. The results confirm the feasibility of a versatile, efficient customer support system through the combination of fine-tuning and prompt engineering, tailored to the specific data sources encountered in customer service scenarios.

Annotation Tool for Dataset Creation

Patrick Giedemann and **Pius von Däniken** and **Jan Milan Deriu**
gied@zhaw.ch

Abstract

The “Annotation Tool for Dataset Creation”, developed as part of the HAMiSoN project (Holistic Approach to Counter Disinformation), represents a significant advancement in the fight against the spread of disinformation on social networks. This web application is specifically designed to facilitate the annotation of transcripts derived from short videos found on social media platforms, which are suspected of containing disinformation. Aimed at empowering annotators, the tool is equipped with a suite of interactive features that simplify the process of navigating, annotating, and understanding the content that requires attention.

The primary objective of the tool is to enhance the annotation process, making it more intuitive and efficient for users. By providing real-time insights into the dataset creation process and the characteristics of the dataset itself, the tool not only enhances the productivity of annotators but also contributes valuable data for researchers involved in the HAMiSoN project. Its ability to handle multimodal data, including video, text, and audio, underscores the comprehensive approach taken by the developers to address the multifaceted nature of disinformation.

Moreover, the “Annotation Tool for Dataset Creation” has been thoughtfully designed with scalability and adaptability in mind, allowing for its application beyond the initial scope of the HAMiSoN project. This flexibility makes it a valuable resource for future research endeavors aimed at curating datasets across various domains. As such, the tool not only serves the immediate needs of the HAMiSoN project but also lays the groundwork for broader applications in dataset creation.

Automatic Identification of Swiss German Dialects Via Speech-to-phoneme Transcriptions

Laura Bolliger and **Safiyya Waldburger** and **Mark Cieliebak** and **Jasmina Bogojeska**
bollilau@students.zhaw.ch

Abstract

Within the domain of multilingual speech recognition systems, differentiating between languages and dialects is crucial. This task is particularly challenging for low-resource languages like Swiss German, which is spoken by comparatively few people. Our project explored various approaches for automatically identifying Swiss German dialects from audio data. The objective was to classify speech samples into one of seven dialect regions.

In a first attempt, the pretrained speech recognition system Whisper was fine-tuned on subsets from two recent Swiss German corpora SDS-200 and STT4SG-350. The number of speakers as well as the number of samples per dialect region were varied, and it was found that a larger number of speakers and a balanced number of samples per dialect and speaker are favorable. Overall, the Whisper-based classification model already achieved acceptable results, but it still had difficulties with certain speakers and dialect regions. Furthermore, the model tended to wrongly classify samples from speakers who came from the border areas of the dialect regions where the dialect features probably were less distinguishable.

In a second attempt we pursued an approach that has been explored very little in the context of the dialect identification task. To our knowledge, it has only been used once before for this task. The main idea behind it is to eliminate all non-linguistic features such as speaker features or noise that could distract a model from learning the dialect features, by first automatically transcribing the speech samples to phoneme sequences using a phoneme recognizer model. After that a classifier model is trained on the phoneme sequences to identify the dialect regions. To find the best performing combination of phoneme recognizer and classifier, several models and algorithms were tried out. For the phoneme recognizer, state-of-the-art pretrained cross-lingual speech-to-phoneme models were used to generate high-quality transcriptions. For the classifier, simpler classical algorithms were compared with more complex deep learning approaches. On the one hand, this led to a much more efficient training process in comparison to the first approach, since the speech samples had to be transcribed only once per phoneme recognizer, and the classifier only had to process phoneme sequences instead of raw audio data. On the other hand, the best model combination outperformed the first attempt.

Can NLP Models and Methods Be Applied to Eeg Data?

Lino Casanova and Andreas Marfurt
lino.casanova@gmail.com

Abstract

Our work introduces an application of Transformer models and transfer learning to electroencephalogram (EEG) data. We start by creating embeddings, then pretrain on large EEG data and finally finetune on the task of sleep stage classification. Related work in the field of sleep phase classification includes the YASA algorithm based on LightGBM. In the processing of EEG data with transformers, Neuro-GPT stands out as a pioneer. The embeddings are created by first segmenting EEG signals into uniform-length parts and utilizing power spectral density (PSD) analysis to derive feature vectors. The embeddings are then constructed from these feature vectors. The model's architecture consists of a BERT encoder followed by a classifier. In a first step, the model was pretrained geusing EEG data from the Temple University Hospital EEG Corpus with a masked language modeling objective. Subsequently, the model was finetuned using only 197 whole-night polysomnographic recordings with annotated sleep stages. The results show that using the PSD analysis to embed EEG data is a valid approach. Furthermore they indicate that pretraining on large EEG datasets enhances performance on downstream tasks. Finetuning on the downstream task of sleep stage classification with 5 classes achieves a mean accuracy of 58%, which is nontrivial but can still be improved. There is a noticeable difference between the accuracy of the "Wake", "REM" and "N3" phases at 69% and that of the "N1" and "N2" phases, which is 41%. To summarize, our study shows that applying NLP methodology to EEG data works almost without adaptations. At the same time, there is a lot of room for improvement and we expect further advancements in this field.

ChaLL - A Chatbot for Language Learners

Manuela Hürlimann and **Luzia Sauer** and **Gerold Schneider**
Johannes Graën and **Jean-Philippe Goldman** and **Janick Michot**
Katsiaryna Mlynchyk and **Ahmet Yavuz Uluslu** and **Irina-Camelia Stroescu**
Jan Deriu and **Michael Geiss** and **Mark Cieliebak**
manuela.huerlimann@zhaw.ch

Abstract

Speaking practice is essential for foreign language learning, but setting up effective and engaging speaking activities in the classroom is challenging. The goal of the ChaLL project is to create a prototype of a digital speaking companion that can be used to develop language learners' interactive speaking skills. Our goal is to develop a voicebot with which learners can lead both informal open-ended conversations and solve specific tasks that train their ability to use speech functions (e.g. expressing an opinion, asking for clarification) necessary to carry out real-life activities (e.g. planning a picnic). Making the voicebot work as a dialogue partner entails solving a number of challenges, such as: - Error-preserving Speech-to-Text, i.e. accurately transcribing the spoken output of young English language learners, including preservation of speaker errors. - Detecting and categorizing errors in the transcripts of the spoken output and providing corrective feedback. - Adjusting/Adapting the bot's response to the learner's output based on the detected English proficiency level and the occurrence of error types. - Developing and aligning different dialogue models that offer an engaging experience, from open-domain chitchat to focused language learning tasks. In this presentation, we will discuss how we collected and annotated the language data necessary to train our models, how we solved the aforementioned NLP challenges, what our system architecture looks like and what design considerations we have made. We will also give including a short demo. We will conclude with the main lessons learnt and an outlook for future developments. The presentation will introduce and report results from the innovation project „Towards a Voice-based Chatbot for Language Learners (ChaLL)”, which is funded by Innosuisse and is jointly developed by PHZH, ZHAW and the University of Zurich in 2023-2024.

Chatbot Evolution at Swisscom Customer Support

Fatemeh Borran

fatemeh.borran@swisscom.com

Abstract

Large language models (LLMs), such as OpenAI's GPT-x, Meta's LLaMA, or Google's LaMDA are revolutionizing the world, especially since ChatGPT benefited from an immense press and social media coverage. These models claim to understand and generate human-like natural language conversations. In addition to producing convincing responses across different domains, the context awareness of these models makes them even more powerful. One can influence the response by providing contextual information and then ask the model to answer specific question in that context. Despite being one of the greatest advances in the field of Natural Language Understanding, using these models in production is NOT recommended. This is mainly due to unreliable content generated by these models and their non-deterministic nature. Thus, the question that we answer in this study is - how can industries benefit from those models? Conventional chatbots typically involve an intent recognition module and predefined flows specifically tailored for selected business scenarios. While these chatbots provide a considerable degree of control over flow execution, building a comprehensive customer care support across all business scenarios is often a laborious task. Conversely, generative bots relying on Large Language Models require little implementation effort at the expense of lack of control. When it comes to developing a customer support bot tailored for a particular case, one must either excel in prompt engineering or possess the necessary data and infrastructure for fine-tuning open-source GPT models, which is not affordable for most companies. The constraints on the token size and the prolonged response time of cloud based GPT models hinder the ability to encompass all business knowledge within a prompt. Retrieval Augmented Generation (RAG) combines both retrieval-based and generative methods to improve the performance of conversational AI systems, mainly, increasing quality by using most relevant information and reducing hallucination. RAG typically involves utilizing a Knowledge Base (KB) for retrieval and incorporating this information into the generative process. While RAG is a powerful approach in chatbot development, there are certain limitations, especially when it comes to using customer and dynamic data. In this case study, we show how Swisscom leverages LLMs to create values for productive systems. Following RAG methodology, we create a KB comprising detailed description of business scenarios along with their corresponding resolutions. We instruct LLM to call external APIs, use customer data, and execute specific actions as required in different situations. For a given customer request, we (1) retrieve data from the KB and (2) retrieve customer specific data (after asking customer to login), then (3) we use LLM model to (i) generate an answer and (ii) define next best action. By adopting this approach, we transition from the world of predefined scenarios in conventional chatbots to more scalable chatbot world with the capacity of handling unlimited scenarios with little implementation effort.

Climate Change Chatbot

Roger Waldvogel and Albert Weichselbraun
roger.waldvogel@fhgr.ch

Abstract

There are still many people who question the sole human responsibility for climate change. According to a study conducted by the University of St. Gallen in 2020, this includes around 39 per cent of the Swiss population. As part of my master's thesis (Roger Waldvogel), I was able to reconfirm the relevance of this problem in 2023. Such doubts can significantly hinder the implementation of necessary measures to mitigate climate change at a political, social and individual level.

This project aims to significantly deepen the population's awareness and knowledge of climate change. On the one hand, it aims to provide well-founded information to people who are unsure about the impact of human activities on climate change. In this way, doubts that have been an obstacle to taking effective measures in their own sphere of influence can be dispelled. On the other hand, the project aims to increase the knowledge of those who are already convinced about climate change. This enables them to argue more cogently in discussions - whether online or face-to-face - and thus refute misinformation.

We want to break new ground in the communication of climate knowledge. Current developments in the field of artificial intelligence enable new, effective and scalable forms of knowledge transfer. We are therefore developing a chatbot to utilise the advantages of this technology for communicating climate knowledge.

Effective knowledge transfer, which is perceived as interesting and stimulating by the addressees, usually combines textual elements with multimedia content. The chatbot will therefore integrate visual media such as images and videos as well as helpful links into its answers. The chatbot will also ask users questions to determine their level of knowledge. This will make it possible to propose targeted measures to mitigate climate change and close knowledge gaps.

Creating A Virtual Receptionist to Lighten Your Employees' Workload: A Case Study with the Canton of Jura and Subsidies for Training.

Emmanuel de Salis and Jonathan Guerne and Hatem Ghorbel
emmanuel.desalis@he-arc.ch

Abstract

The process for getting a subsidy for a formation is lengthy and complex, and people can often get confused by the application's wording or the questions asked. People can also get frustrated when their application for a subsidy is denied for simple reasons that they did not know beforehand. The "Guichet virtuel" (Virtual counter) of the canton of Jura provides a simple form which allow people to see if they are eligible or not for a funding, before filling the whole application.

Nonetheless, a lot of people skip this form and either calls the subsidies department by phone or fill the whole application. This situation wastes a lot of time for the subsidies department staff and creates a lot of frustrations for the applicants.

To resolve this process, we propose a prototype of a virtual receptionist in the form of an LLM-driven chatbot, acting as a first contact before calling the staff or filling the whole application. This chatbot can discuss with a user and understand his or her profile, asking question until the eligibility is verified. This allows users to quickly get an answer just by chatting and avoiding the need to call the subsidies department staff for the simpler questions.

Using a Mistral-based model, we created an architecture where the receptionist is able to discuss with an applicant and extract relevant information from their answer. Then, the model evaluates the maximum information gain for each potential missing information and asks the best question to get a final answer as quickly as possible. Contrary to most approaches, the proposed pipeline benefits from a strong validation process also meant to avoid direct contact between the applicant and the LLM.

This talk will present the problem, how we resolve it with the creation of the chatbot including some technical details, and the implications if the audience wants to create something similar for their company.

Dialect Identifications with Large Language Models

Vani Kanjirangat and **Ljiljana Dolamic** and **Fabio Rinaldi**
vani5019@gmail.com

Abstract

Language identification is the task of classifying utterances into languages, where languages are regarded as discrete classes (one language, one class). Dialects of a language can be quite overlapping, sharing linguistic similarities, which make the problem more challenging to tackle, where even the existing Large Language Models (LLMs) struggle. In the project, we aim to focus on the multilingual capabilities and limitations of existing LLMs, by using dialect identification as the main task. Most LLMs are trained on a huge amount of data, predominantly in English, while it is claimed that their performance is good enough in other languages too. This could be true with high-resource languages but not with others. In this project, we will be focusing on medium and low-resource languages such as Arabic, Swiss-German, Italian, Indo-Aryan, etc. Our experiments on fine-tuned pre-trained encoder-based models (multilingual and monolingual) have shown the supremacy of language-specific models in these tasks. We intend to compare the performance of LLMs (> 10B parameters) with these pre-trained models to analyze their multilingual abilities (non-English).

Harnessing LLMs for Generating Patient Discharge Reports

Norman Süsstrunk and Albert Weichselbraun
norman.suesstrunk@fhgr.ch

Abstract

Administrative processes in healthcare are important, but time-consuming and costly for all parties involved. The developed system aims to enhance the efficiency of documentation processes in psychiatric clinics using artificial intelligence (AI). The goal is to reduce the workload of qualified medical personnel, improve report quality, and reduce overall costs for psychiatric clinics. The project is being developed in collaboration with the Psychiatric Clinic Graubünden (PDGR). Technologically, the Report Recommender is based on harnessing Large Language Models (LLM's) combined with Multi-Document Summarization (MDS). This involves creating a summary from a collection of documents related to a specific topic. This process encounters challenges similar to summarising a long document, as the issue of limited context length becomes more pronounced with multiple documents. Understanding the relationships between these documents is crucial for an effective summary. One common strategy for MDS is concatenating all documents into one large text and applying techniques designed for single-document summarization. However, this approach requires the model to process very long sequences. Therefore, a two-stage process, similar to that used for long document summarization, is commonly employed (Liu* et al., 2018). State-of-the-art approaches also use hierarchical architectures or graph-based methods to capture inter-document relations (Liu & Lapata, 2019). At the same time, MDS approaches increasingly aim to utilise pre-trained encoder-decoder models such as BART, T5, or PEGASUS (Pasunuru et al., 2021). Further, identifying and eliminating redundant information while retaining relevant content is a complex task, particularly when dealing with large volumes of text. Extractive and abstractive summarization share inherent similarities; both are designed to extract the main information from an input natural language text. Current research (Zhang et al., 2023) indicates that the two tasks could mutually enhance each other. Extractive summarization selects key sentences and information from the original text, while abstractive summarization, with the assistance of large language models, rewrites and condenses information coherently. This approach aims to maintain the integrity of medical data while ensuring that the summary is concise and easy to understand. Therefore, a hybrid approach combining information extraction and abstractive summarization will be developed and evaluated.

Interlex – A Search Engine to Explore the Interconnectedness of Swiss Legal Texts

Selena Calleri and **Michelle Wastl** and **Bojan Peric** and **Andreas Abegg**
selena.calleri@gmail.com

Abstract

The Swiss legal landscape is tied to working with text by using non-digital approaches due to its multilingual and federalist nature. Digital approaches must overcome the plurality of formats, contents, and languages in order to provide useful solutions to legal practitioners. We present Interlex, a tool which tries to overcome these chasms by leveraging NLP technologies and knowledge from linguistics and law to make Swiss state and canton level court decisions accessible through their interconnectedness. The tool is built on a collection of 600'000+ web-crawled court decisions. The texts have been preprocessed to form a standardized corpus of Swiss court decisions that is regularly updated. The preprocessing of the texts included manual layout analysis of every court, scraping the text from different data formats, regex-based cleaning and paragraph splitting. In a next step, a fine-tuned BERT model has been used for legal sentence-boundary-detection before the sentences were embedded using a multilingual BERT-based sentence encoder to facilitate semantic similarity analysis in multiple languages. The data is then stored in a dynamically expandable database, which is regularly updated with new court decisions. This database serves as a backend for the Interlex web application. Interlex focuses on identifying so-called textual building blocks (TBBs) which are “prefabricated argument blocks.” They are often repeated throughout court decisions and usually require years of practitioner experience to be identified. Apart from yearlong experience the only other approach to find these TBBs is using books or articles with precedents (dt. “Präjudizien”) which indicate potentially relevant passages yet imply a) the continuous maintenance of the material, b) tedious manual labor to create it by multiple authors, and c) access to the collection. Our tool recognizes relevant and potential TBBs on sentence level by using four different levels of similarity: i) exact copy, ii) strongly similar wording, iii) similar meaning with edits, and iv) similar meaning but different words. These surface form and semantic level similarities are automatically detected by leveraging multiple edit differences and similarity scores. This approach differs from manual selection as every phrase is considered a potential TBB and then defined as such if it meets the interconnectedness criteria. Interlex allows for 2 modalities: either searching in full text for relevant keywords and exploring the found passages in the full text, or directly exploring TBBs based on a search term and finding the most relevant sentences, defined by their interconnectedness. This required a parametrization of TBBs, as for the first time a data-based approach was applied to identify them systematically in a comprehensive corpus of court decisions. The exploratory nature of the tool and the ambiguous definition of TBBs make the tool’s evaluation complex. We employed sample-based qualitative analyses to evaluate data cleanliness, embedding model selection, and the interconnectedness score.

Kickstarting Legal Multi-label Classification Experimentation

Claudia Schulz and **Martina Forster** and **Prudhvi Nokku** and **Stavroula Skylaki**
clauschulz1812@gmail.com

Abstract

Multi-Label Classification (MLC) is a common task in the legal domain, where more than one label may be assigned to a legal document. A wide range of methods can be applied, ranging from traditional ML approaches over fine-tuned Transformer-based architectures to zero-shot LLM prompting. Depending on the data characteristics, such as available training data, text length and number of labels, different approaches may yield the best results. In addition to prediction performance, another important consideration in practice is prediction speed and cost of the different suitable approaches.

Experimenting with different baseline approaches to find the most promising one for a given legal MLC task is usually time-intensive. To cut down the baseline testing time on new projects, we designed a MLC baseline suite that allows to seamlessly train and evaluate a variety of different MLC models in one go. This includes traditional similarity methods like TF-IDF as well as BERT-style models such as RoBERTa, Bi-/Cross-Encoders, and T5, and allows to compare domain-specific with general-domain models.

We tested our baseline suite on two public legal datasets, POSTURE50K and EURLEX57K, and compare the results with state-of-the-art LLM prompting approaches. To explore the comparative advantage offered by different approaches in relation to the dataset properties, we varied the amount of training data and the number of labels in these datasets, simulating different types of datasets. Our results highlight performance-speed-cost trade-offs.

Language Models Explore the Linguistics of Chess

Lars Schmid and **Jerome Maag** and **Mark Cieliebak** and **Pius von Däniken**
schmila7@students.zhaw.ch

Abstract

In this research, we explore how to train Large Language Models (LLMs) to generate valid chess moves, extending their use from traditional Natural Language Processing to structured, rule-based language systems. We trained the openly available GPT-2 architecture on chess notation datasets containing up to one million games. We then evaluated model performance using the following metrics: “Average Number of Correct Plies”, which assessed the model’s ability to generate moves for both players by recording the average number of correct moves across 100 simulated games; “Hard Position Accuracy,” which evaluated the model’s capability to handle 67 challenging positions, assessing the percentage of scenarios successfully solved; and “Legal Piece Moves Accuracy,” which specifically evaluated the model’s ability to generate valid moves for a given piece, based on its position on the board. This last metric also aimed to assess the model’s capability in modelling the state of the game board. The evaluations revealed a significant correlation between dataset size and model effectiveness, particularly emphasizing the importance of integrating the Beginning of Sequence (BOS) token to enhance syntactical correctness and reduce errors. Despite challenges with some complex chess positions, the best model achieved an average accuracy rate of up to 50 correct moves per game, demonstrating the adaptability of LLMs to rule-based systems like chess and opening new possibilities for AI in structured domains.

Leveraging LLMs to Enhance Clinical Record Analysis and Retrieval

Lorenzo Ruinelli and **Amos Colombo** and **Oscar William Lithgow Serrano**
Andrea Franchini and **Joseph Cornelius** and **Sandra Mitrovic** and **Fabio Rinaldi**
lorenzo.ruinelli@eoc.ch

Abstract

We describe the current status of a project which aims to explore the potential of large language models (LLMs) in enhancing the analysis and retrieval of clinical records. The pilot application will enable a more efficient interaction of clinical practitioners with the documents related to the medical history of a patient. Instead of painstakingly consulting dozens of documents in order to gather the information necessary to properly treat the patient, the application will enable a Question/Answer interaction paradigm. A doctor will be able to ask for precise information through a dialogue interface, and drill down to what he/she needs.

The project is structured as a collaboration between the Natural Language Processing (NLP) group of the Dalle Molle Institute of Artificial Intelligence (IDSIA USI SUPSI) and the “Information and Communications Technology area” of the Ente Ospedaliero Cantonale (EOC ICT). A very preliminary exploration of the possibility to use an in premise LLM has been conducted in 2023 at EOC with external consulting from IDSIA.

The first goal consists of efficiently identifying and categorizing pertinent medical concepts, diagnoses, procedures, and medications mentioned in free text narratives within clinical records.

The second goal consists of developing tools that can analyze unstructured clinical data to generate insights and summaries for clinicians, potentially aiding in diagnosis, treatment planning, and personalized medicine approaches.

A third goal is the semi automated generation of discharge letters. Initial results concerning the semi automated generation of discharge letters with LLMS, starting from de identified versions of the patient’s documentation, have already been obtained, and will be presented at the conference.

Medicrawl: An ML Based System to Automate the Search of Illegal Products in Ecommerce

Nicolas Perez Gonzalez

nicolas.perezgonzalez@swissmedic.ch

Abstract

Swissmedic is the National Authorisation and supervisory authority for drugs and medical products. It ensures high-quality, safe and effective medical products. As part of its mandate, Swissmedic regularly oversees online markets on the search for illegal medicinal products. This search is time consuming as well as resource intensive. In order to optimize this workflow, MediCrawl was developed in the innovation team within Swissmedic. MediCrawl is a platform developed in the Cloud that allows a centralized analysis of products of interest found online. These products are collected via crawling and are later presented to the Swissmedic specialists. Due to the large number of incoming products of interest, a series of filters have been developed including a Machine Learning based filter based on the text description of these. With a dataset of roughly 60,000 cases, we trained multiple classifiers based on BERT embeddings, which identify the binary signal and labels if the case is relevant for Swissmedic. In development, our classifier identified relevant products with a precision of 0.85 and recall of 0.92. 3 months later, with new unseen data, precision dropped to 0.61 while the relevant score of recall remain high at 0.94. After two years of development, our results with MediCrawl are promising and a more permanent implementation in Swissmedic is underway as well as an international collaboration for this tool. Further research will be focused on improving scores with new methodologies as well as the benefits of moving training into a MLOps pipeline.

Navigating the Commodity Market with Language Models

Himmet Kaplan and **Albert Weichselbraun** and **Martin Tschudy**
kapa@zhaw.ch

Abstract

Commodities are of utmost concern to the global economy. Still, their availability is by nature exposed to worldwide distributed issues such as logistics (theft, perish, delivery accidents), geopolitical developments (corruption, war, reforms, sanctions), and environmental influences (weather). While traditional sources of information like inventory reports, weather forecasts, and satellite data can provide insights, they often lack accessibility, timeliness, and reliability. As an alternative news publications, offer a rich, diverse, and timely source of information on events impacting commodities. Yet, the sheer volume of news data of approximately 50,000 publications daily on commodities, presents a challenge in extracting actionable intelligence.

Recent advancements in Natural Language Processing led to the emergence of Large Language Models (LLMs) which offer a promising solution to manage this vast amount of text data. However, empirical observations reveal that LLMs fall short in classifying how specific news affects the availability of certain commodities. central problem is LLMs' ethical, moral, and environmental evaluation of news. These evaluations may not accurately capture a commodity's availability or market value. For example, LLMs could negatively rate news about an oil platform accident, causing injuries or deaths, due to ethical concerns. However, such an incident leads to a decrease in supply, thereby having a positive effect on the oil price. This contradicting classification suggests that the complex interplay between supply and demand is not always reliably captured by conventional LLMs. This is because they primarily focus on the general sentiment rather than the asset-specific implications, even with optimized prompts.

To address this problem, we introduce NewsForce, a research project that combines language models with economic models. This enables the development of commodity-specific language models, such as CrudeBERT, which is designed for the crude oil market. As a result, the language model was able to more accurately capture the complex interplay between supply and demand in the oil market, thus pragmatically estimating the availability of crude oil. The classifications from the LM developed by NewsForce show a notably higher degree of relationship to the WTI oil prices relative to the classifications of a financial LM.

This outperformance highlights the significance of merging economic principles with language models. Further experimentation shows that our method is adaptable to other non-precious commodities. These include coffee and iron, showcasing its flexibility and potential to transform commodity market analysis by enabling timely identification of risks from the global news traffic.

NLP in Support of Pharmacovigilance

Fabio Rinaldi and **Lorenzo Ruinelli** and **Roberta Noseda**
Oscar William Lithgow Serrano and **Sandra Mitrovic**
fabio@idsia.ch

Abstract

Pharmacovigilance (PV) is the science and activities related to the detection, assessment, understanding, and prevention of adverse effects of drugs, or of any other drug related problem. It is a critical component of drug safety, and it is essential for the safe and effective use of medicines.

In Switzerland Healthcare professional who suspect an adverse drug reaction in their patients are legally required to report it to one of the five PV center. In those centers, PV experts manually assess the validity of received alerts to further process only retained “active” spontaneous safety reports, thus making causality assessment and sending the information to Swissmedic.

In this project, we aim to investigate the potential of state of the art NLP methods applied to the automated processing of clinical records in order to provide reliable signals for the potential presence of adverse drug reactions (ADRs) in the context of pharmacovigilance. Concretely, if the project is successful, the partner hospital will be able to significantly improve the detection of ADRs, and once a similar approach is adopted at other Swiss hospitals, it could have a major downstream impact on public health.

The project is structured as a collaboration between the the regional PV Center of Southern Switzerland, based at the Ticino Cantonal Hospital (EOC), and the NLP group of the Dalle Molle Institute of Artificial Intelligence in Lugano. The PV center will be responsible for collecting and annotating the datasets necessary for the project.

We are starting from an initial (imperfect) version of the annotations obtained through the method of distant supervision. Specifically, the data present in the existing reports will be automatically mapped (as far as possible) onto the original discharge letters from which they were manually extracted.

The problem is framed as two distinct NLP tasks. The first task consists in distinguishing discharge letters which contain ADRs (which should be further processed) from those which do not, it is thus a simple classification problem. The second task consists in the identification of the elements that need to reported to swissmedic, such as drugs administered to the patient, their dosages, comorbidities, age and sex, etc., it is thus a named entity recognition problem.

The project aims at establishing a solid baseline for both tasks, using traditional, efficient, and low cost technologies, before exploring the cost-benefit trade-off of employing LLMs.

Online Bert-based Topic Modelling

Eric Gericke and **Nicolas Jamet** and **Tian Guo** and **Martin Schüle**
geci@zhaw.ch

Abstract

Topic modelling is a machine learning method for identifying and extracting relevant topics from large amounts of text. It allows patterns and trends to be identified in the data that would otherwise be difficult to spot and large amounts of unstructured data to be organized. With news data, for example, it enables relevant topics to be identified and analysed in real time, helping to find information on specific topics quickly and easily. There are a number of different types of topic models such as the classic Latent Dirichlet Allocation (LDA) [1] and variants thereof. In recent years BERT-based models have become popular as they provide a more accurate and contextual text representation allowing for improved topic identification and categorisation. With so-called dynamic topic modelling we can track topics over time thereby analysing not only the current topics but also their development over time. Thus, dynamic topic modelling may enable a better understanding of complex temporal patterns and can be used to predict future trends and topic developments. However, the constant flux of new data, e.g. in news stream settings, poses a challenge as the standard dynamic topic modelling methods usually need to use the entire dataset resulting in significant difficulties as data volumes grow. The solution to this problem is to move to online models that are able to incrementally process and integrate new data, thereby avoiding the need to retrain the model with the entire dataset. In this contribution we demonstrate a BERT-type dynamic topic modelling approach which can reliably track topics over time without the need to merge datasets or to re-train models on the full dataset.

Orbis2 - A Natural Language Processing Benchmarking Framework That Supports Drill Down Analyzes

Norman Süsstrunk and **Roger Waldvogel** and **Andreas Murk**
André Glatzl and **Albert Weichselbraun**
norman.suesstrunk@fhgr.ch

Abstract

Competitive benchmarking of natural language processing (NLP) systems has contributed considerably towards improving the performance of NLP methods. Orbis2 is an open-source benchmarking framework designed towards addressing the need of (i) evaluating natural language processing systems, and (ii) obtaining insights that help in further enhancing them. The framework was developed to support the development of information extraction components in multiple Innosuisse-funded projects such as CareerCoach, IMAGINE, Job-Cockpit and Future of Work. Traditional evaluation tools like GERBIL focus on aggregated statistics such as accuracy, precision, recall, and the F1 metric which indicate the overall performance of the evaluated systems. Although these aggregated metrics are well-suited for comparing systems, they provide little help in understanding evaluation results. Orbis2 addresses this shortcoming by enabling drill down analyzes, which contextualize evaluation results (e.g., by visualizing correct and incorrect annotations within their textual context), therefore, helping researchers in better understanding the strengths and weaknesses of their systems, and in systematically addressing them. The framework integrates with existing NLP annotation tools such as Label Studio and Doccano, enabling users to seamlessly import and utilize corpora from these platforms. Orbis2 also supports a variety of evaluation types, including named entity classification, named entity linking, named entity recognition, and page segmentation. Work on complex slot-filling evaluation tasks is currently underway. The Orbis2 developer team, has invested significant efforts towards enhancing user experience. Its design allows users to easily navigate, compare and analyze metrics without the need to switch between multiple interfaces. Orbis2 is licensed under the Apache 2.0 license, publicly available on GitHub (github.com/orbis-eval), and encourages contributions from the community, to further improve and innovate in the area of visual benchmarking.

Personalized and Interactive Education in Migraine Using Artificial Intelligence

**Fabio Rinaldi and Oscar William Lithgow Serrano and Andrea Franchini
Chiara Zecca and Giulia Mallucci and Alberto Cordella**
fabio@idsia.ch

Abstract

Migraine is a prevalent neurological condition that affects approximately 1 in 7 individuals worldwide. It places a substantial burden on both individuals and societies, encompassing healthcare utilization, medication expenses, physical and psychological disability, diminished quality of life, and decreased productivity.

Efforts to enhance public awareness regarding migraine, its influence on daily life, and the available treatment alternatives are frequently insufficient due to restricted resources or inadequate emphasis. Furthermore, educational initiatives must be tailored to varying levels of education, cultural backgrounds, and languages, thereby guaranteeing the dissemination of information across a wider spectrum of individuals.

The project aims to develop an AI powered educational platform for personalized migraine education and assess its effectiveness and safety through a randomized clinical trial. Our project includes two parts. In part I, a platform utilizing Large Language Models (LLMs) has been tailored to generate personalized content, answer queries, and provide support on migraine. Two approaches (A and B) are being considered.

Approach A involves adapting a pre trained commercial LLM to the project goals using a combination of prompt engineering approaches, while approach B will further involve fine tuning an open source LLM. Approach A is currently under exploration due to its feasibility within the project's timeframe. However, it has limitations in control and data privacy. After initial testing, approach B will be pursued for optimized information control and customization.

Part II involves a 6 month randomized, clinical trial with 200 migraine patients. Group 1 will use the AI platform, and Group 2 will receive standard education. Learning effectiveness, clinical metrics, and adverse events will be compared between groups. The project's innovation lies in its unique AI driven education approach, contributing to fill a gap in migraine education and ultimately improve migraine patients' quality of life and reduce migraine related costs.

Presenting Bust - A Benchmark for the Evaluation of System Detectors of LLM-generated Text

Joseph Cornelius and **Oscar William Lithgow Serrano** and **Sandra Mitrović**
Ljiljana Dolamic and **Fabio Rinaldi**
joseph.cornelius@idsia.ch

Abstract

The rapid advancement of Large Language Models (LLMs) like GPT-4 presents a growing challenge in distinguishing between human and machine-generated texts, heightening the risks of fraud and misinformation. Studies indicate that most adults struggle to differentiate between human and machine-generated content, underscoring the urgent need for reliable detection systems. To address this, there is a pressing requirement for a flexible benchmarking dataset that can effectively evaluate detection systems across various tasks.

In this work, a comprehensive approach was adopted, involving: 1) the creation of a dataset comprising instructions and responses obtained by different generator models; 2) the evaluation of different detectors' performance across different tasks; 3) the accomplishment of meta-analysis and development of surrogate models to provide in-depth insights into dataset characteristics and the potential to simulate detectors behaviour. The resulting benchmark dataset comprises over 25,000 texts derived from 3,180 instructions and synthetic responses generated by seven different instruction-tuned generators across ten tasks sourced from three different datasets. Five detectors (both close- and open-source), with varying training corpora and employing different detection strategies, were evaluated to assess their performance across these ten tasks.

The study revealed notable performance differences among detectors across different tasks. Furthermore, the detectors exhibited varying capabilities in detecting text produced by different generators. Surrogate models highlighted the difficulty in explaining the most performant detector, frequently relying on unexpected textual features.

Our (publicly available) automated pipeline integrating all analyses, facilitates detector selection based on text style and specific use-cases. This benchmarking effort marks a significant step forward in addressing the escalating challenges posed by the proliferation of machine-generated text, providing valuable insights for the development of more robust detection systems in combating fraud and misinformation.

Promise: Model-driven Stateful Prompt Orchestration for Persuasive Conversational Interactions

Wenyuan Wu and **Jasmin Heierli** and **Max Meisterhans**
Adrian Moser and **Andri Färber** and **Mateusz Dolata**
Elena Gavagnin and **Alexandre de Spindler** and **Gerhard Schwabe**
wenyuan@ifi.uzh.ch

Abstract

Natural language-based interactions are gaining importance within health information systems, showing promise in achieving key medical objectives. For example, they support a more patient-centered approach in healthcare by adapting to the unique needs of each patient. When properly integrated, these interactions can streamline patient-physician collaboration, leading to more targeted and effective consultations and treatments. Such consultations involve conversational interactions that usually comprise several consecutive, alternating, possibly nested strands of conversation, and in which persuasive strategies are used in a targeted manner. Consequently, if a system is to have a similar treatment benefit between consultations, it must be able to handle such complex and persuasive interactions.

While the development of increasingly powerful language models (LMs) raises expectations for more sophisticated conversational behaviors, directing LM behavior remains challenging, making fast, iterative experimentation difficult. Despite recent advances in prompting techniques that provide a more efficient method of controlling behavior, prompts alone cannot ensure consistent LM behavior in complex interactions.

To facilitate the investigation of the feasibility and utility of complex and persuasive conversational interactions, we developed PROMISE (PRompt-Orchestrating Model-driven Interaction State Engineering). This framework provides support for the rapid design, implementation, and experimental validation of such interactions by breaking down complex interactions into more specific prompts that increase the predictability of LMs and leverage their conversational capabilities.

PROMISE has been effectively applied in projects such as to improve health literacy or therapy adherence. It enables personalized conversational interactions by integrating patient data from electronic health records, facilitating empathetic and persuasive conversations, and dynamically adapting conversational behaviours to achieve intended interaction outcomes. It is available at GitHub (<https://github.com/zhaw-iwi/promise>) and includes an API supporting the design and implementation of conversational interactions. It also provides support for web-based interactions, including transparent persistence, a REST API, and a ready-to-use default chat interface. More recently, we added the means to dynamically adapt persuasion strategies throughout complex interactions.

Reducing Complexity of Scientific Literature By Automated Simplification and Translation

Nico Colic and Fabio Rinaldi

nico.colic@idsia.ch

Abstract

Scientific literature encodes a wealth of knowledge relevant to various users. However, the complexity of scientific jargon makes it inaccessible to all but domain specialists. It would be helpful for different types of people to be able to get at least a gist of a paper. Biomedical practitioners often find it difficult to keep up with the information load; but even lay people would benefit from scientific information, for example to dispel medical misconceptions. Besides, in many countries, familiarity with English is limited, let alone scientific English, even among professionals. All this points to the need for simplified access to the scientific literature. We thus present an application aimed at solving this problem, which is capable of summarising scientific text in a way that is tailored to specific types of users, and in their native language. For this objective, we used an LLM that our system queries using user selected parameters. We conducted an informal evaluation of this prototype using a questionnaire in 3 different languages. We will present the architecture of the system as well as the results of our user evaluation. A live demo is possible.

STT Explorer: A Look Beyond Wer

Katsiaryna Mlynchyk and **Gabriel Hunziker** and **Lars Mosimann**
Ralph Scheu and **Mark Cieliebak**
mlyn@zhaw.ch

Abstract

STT Explorer introduces a new dimension to evaluating and inspecting STT (Speech-to-Text) results. While traditional metrics like Word Error Rate (WER) give a general idea of performance, they often miss the subtleties of STT output. STT Explorer is meant to fill this gap.

Through visual representation, the STT Explorer allows users to easily compare the performance of different STT engines by examining the alignment between reference and predicted texts. Users can upload STT results in JSON or CSV format, or input them manually. The tool then automatically aligns the texts, enabling a detailed examination of each sentence. Mistakes are color-coded for clarity, and alignments are displayed both succinctly (as dots) and in full (as words).

Current version of the tool utilizes a phonetic-based alignment algorithm, supports m-to-m alignment, meaning multiple words can align with multiple words if phonetically sensible. Additionally, the tool accommodates the alignment of multiple hypotheses from various STT engines. Detailed corpus statistics are also provided for further analysis.

Swiss AI Initiative - Collecting Large Amounts of High-quality Data for Training Large Language Models

Jan Deriu and **Maud Ehrmann** and **Emanuela Boros**
Maximilian Böther and **Christiane Sibille** and **Ihor Protsenko**
Marta Brucka and **Imanol Schlag** and **Elliott Ash**
deri@zhaw.ch

Abstract

The Swiss AI Initiative, a consortium led by ETH Zurich and EPFL, consists of over 70 professors in Switzerland. Its goal is to develop and research large-scale large language models (LLM) for the Swiss population, leveraging the CSCS's Alps supercomputer. Central to this initiative is the commitment to curating high-quality datasets reflective of Swiss cultural and linguistic diversity. High-quality data is crucial for the effective pre-training of LLMs. Current open-source LLMs mostly utilize extensive datasets compiled from web sources, often subjected to minimal quality control. This approach results in datasets containing low-quality texts from social media and other unreliable platforms, embedding significant biases within the LLMs that necessitate alignment. Empirical evidence suggests that integrating high-quality texts into the training regimen enhances LLM performance across various parameter scales [1]. A further complication arises from the reliance on copyrighted content, including newspapers and books, which often embroils open-source initiatives in legal complexities, hindering the release of models under permissive licenses. This environment also contributes to publishers' reluctance to grant research access to their content. This talk will present the overall vision of the Swiss AI Initiative with a particular emphasis on the challenges in data acquisition, storage, and curation.

Understanding Customer Sentiment with NLP from Sparse Labelled Data

Daniel Perruchoud and Joseph Weibel

daniel.perruchoud@fhnw.ch

Abstract

Although sentiment analysis of texts is generally considered a solved problem, implementing a solution for real-world applications can pose challenges. On one hand, public data typically differ significantly from messy texts of real-world applications, especially if authored by various people. On the other hand, solutions often rely on pre-trained models primarily available in English, and even if models are available in other languages, they tend to have lower quality. Finally, most solutions require labelled data, whose acquisition costs businesses want to minimize. This project systematically investigates methods to address and mitigate these challenges, specifically focusing on how and to what extent the amount of labelled data can be reduced. We develop a sentiment analysis solution for a small Swiss bank using transformer-based models. We use binary classification, since the primary focus is the identification of early signs of negative customer experience, which the bank wants to address. For that, we raise over 10'000 client advisor notes of individual customer contacts, with 300 texts manually annotated by two employees. The notes consist of short German sentences (length mean/std. dev.: 162/105 chars) without fixed structure. For comparison, we also apply our approach to several publicly available datasets. To identify sentiment in these texts, we tried 49 strategies using LLM in-context learning via Mistral 7B, Mixtral 7x8B, Llama 2 and 3, and different prompting strategies including Zero-Shot, Few-Shot, Chain-of-Thought and Reasoning. The experiments show, that given an appropriate prompt, smaller models achieve similar performance as larger models. However, larger models generally encounter fewer issues with different prompt styles. Furthermore, we utilize the k-nearest neighbours (kNN) algorithm and sentence-transformer embeddings for text similarities to get sentiment labels. We vary amounts of labelled data to find optimal parameterization. The most accurate labels are obtained when considering at least 150 samples, at least three neighbours and weighting the neighbours' labels based on similarity (F1: ≈ 0.80). Sentiment identified by LLM is accurate, but also computationally expensive in terms of compute time and infrastructure required for inference. Fine-tuning BERT models with weak labels from LLMs is a method to transfer this knowledge into simpler models. We apply self-supervised learning by fine-tuning various pre-trained BERT models with 10'000 weak labels. Best results are achieved with weak labels based on a Few-Shot prompt with six hard labels/examples and Mistral 7B or Llama 2 7B. The results are slightly worse (F1: 0.76) than using the LLM directly for classification (F1: 0.78), but the number of parameters is reduced drastically from 7B to 0.11B. If instead weak labelling with weighted kNN (k=8) is used for BERT fine-tuning, 40 hard labels are needed to reach the same quality. Also, only one-fifth (32 instead of 160) of the hard labels are necessary to achieve an equally good model than just training on all hard labels (F1: 0.73). By using all 160 hard labels for finding similar texts, the quality of the fine-tuned BERT model can even be further improved (F1: 0.82). Applying our approach to several publicly available English datasets including financial news headlines, product and movie reviews, we consistently found results superior to the ones reported above for the bank. This indicates that published results cannot be directly transferred to real-world scenarios. One reason is that publicly available pre-trained models for German show lower performance. Additionally, quantitative analyses show that the bank's client advisor notes employ a broader vocabulary than e.g. financial news headlines and therefore require more data to generalize. Moreover, labelling is not straightforward for humans, as a large portion of the texts does not contain sentiment and abundant neutral texts complicate the identification of negative texts. Through our systematic evaluation, we were able to examine the factors contributing to the quality differences between the datasets and determine which weak-labelling strategy yields the best results in each case. We plan to publish our code so that tests can be conducted for other real-world scenarios.

Virtual Kids: Using LLMs for Role Playing in Professional Communication Training

Don Tuggener
tuge@zhaw.ch

Abstract

The Virtual Kids project develops a learning platform for criminal investigators to train the interrogation of children who are, potentially, victims of sexual abuse. These interrogations must follow specific guidelines and rules of communication, such as avoiding suggestive questions, to be valid for prosecution and to protect the children from (re-)experiencing trauma.

The learning platform consists of multiple virtual characters of children that have their own story and different demographic properties. The trainees select a case and communicate with the characters via speech technology. The virtual characters are rendered in 3D and respond dynamically in real-time.

A specific contribution of the project is that the interrogations do not follow a pre-determined path consisting of multiple-choice branches. Rather, the trainees are free to formulate and ask their own questions in their own language.

In our presentation, we will showcase how forensic psychologists prepare the cases, how we detect unwanted questioning techniques, and how we alter the virtual characters' behavior accordingly in a dynamic fashion using Large Language Models (LLMs).

What Can We Discover About Panic and Anxiety from Bloggers in Quora and Reddit?

Sandra Mitrović and Oscar William Lithgow Serrano
sandra.mitrovic@idsia.ch

Abstract

The recent surge in research interest underscores the undeniable significance of mental health, particularly in the realm of social media analysis. While existing studies have predominantly focused on depression and suicide, there is a growing recognition of the equally critical role of anxiety and panic disorders. Panic, characterized by sudden and overwhelming fear, and anxiety, marked by persistent unease and worry, are distinct yet interconnected phenomena. Despite their shared traits, literature confirms the validity of distinguishing between generalized anxiety and panic.

Anxiety is frequently linked with depression, while panic remains relatively understudied. However, understanding panic is crucial due to its association with heightened risks of severe psychological issues such as acute suicidality and agoraphobia, as well as its potential relevance in identifying individuals with post-traumatic stress disorder (PTSD). Yet, current literature lacks computational approaches to differentiate between panic and anxiety in social media data.

To address this gap, a dataset comprising 1,930 user posts from platforms like Quora and Reddit was collected, and a comprehensive analysis was conducted. The research aimed to answer key questions regarding the (dis)similarity between panic and anxiety-annotated posts, the efficacy of classifiers in discerning panic from anxiety, and the insights gained from a qualitative natural language processing (NLP)-assisted analysis.

By leveraging various analytical approaches, including lexical analysis, emotion assessment, and machine learning techniques, the study not only contributes new insights but also provides the dataset, analysis, and code for future research endeavors. Practical applications of the findings include potential clinical utility in distinguishing between anxiety and panic triggers, guiding emergency medical responses, and encouraging individuals to seek support.

Ultimately, this research aims to enhance mental health understanding and practice by enabling the screening of social media posts displaying indications of anxiety or panic. Through these efforts, it is hoped that a more comprehensive understanding of mental health issues in online discourse will be achieved, ultimately contributing to improved mental health outcomes in society.

Whisper Fine-tuning for Swiss German: A Data Perspective

Claudio Paonessa and **Vincenzo Timmel** and **Manfred Vogel** and **Daniel Perruchoud**
claudio.paonessa@fhnw.ch

Abstract

In our recent exploration of fine-tuning OpenAI’s Whisper speech-to-text model for Swiss German, we built a data processing pipeline to transform readily available sentence-level datasets to long-form audio to be fully compatible with the Whisper model. Our pipeline ensures the preservation of the segmentation capabilities of the model and prevents the model from losing its ability to handle audio with arbitrary length. With additional High German data to preserve the German language and weakly-labeled real long-form data, annotated through the original Whisper Large V2 model, we achieve a new state-of-the-art (SOTA) model for Swiss German speech to High German text translation. The original OpenAI Whisper model shows large variations in performance across the different Swiss dialects, ranging from WER of 17.63 for Central Switzerland to 29.31 for the Valais dialect. Our model significantly improves those error rates and we measure a much more narrow range from 10.73 for Central Switzerland to 13.68 for the Bern dialect. To evaluate its performance on real long-form audio, we curated a test dataset from Swiss German TV shows with human-annotated subtitles. The novel test dataset consists of 20 hours of material from selected TV shows, i.e., Einstein, Puls, Impact Investigativ, SRF Kids News, and SRF ohne Limit.

Chapter 4

Shared Task

This chapter introduces the “SwissText 2024 Shared Task: Automatic Classification of the United Nations’ Sustainable Development Goals (SDGs) and Their Targets in English Scientific Abstracts” submitted by Simon Clematide, Martin Volk, Tobias Fankhauser, Lorenz Hilty, and Jürgen Bernard, and the system papers of four research teams that participated in the task.

Shared Task

Automatic Classification of the United Nations' Sustainable Development Goals (SDGs) and Their Targets in English Scientific Abstracts

Simon Clematide, Martin Volk, Tobias Fankhauser, Lorenz Hilty and Jürgen Bernard

UZH

Description

In the current era of global challenges, the United Nations' Sustainable Development Goals (SDGs) serve as a universal call to action, addressing critical issues such as poverty, inequality, climate change, environmental degradation, and peace and justice. The alignment of scientific research with these goals is pivotal for measuring and enhancing the impact of academia on these global objectives.

The Swisstext 2024 Shared Task proposed by members of the Departments of Computational Linguistics and Informatics at the University of Zurich targets an innovative and significant undertaking: the automatic classification of scientific abstracts (in English) with respect to SDGs and their specific targets. This task not only aligns with the increasing importance of interdisciplinary research towards sustainable development but also represents a crucial step in integrating advanced Natural Language Processing (NLP) within the realm of sustainability research. By automating the classification process, this Shared Task aims to facilitate the identification and analysis of research towards the SDGs, thereby fostering a targeted and efficient approach in addressing the world's most pressing challenges.

This Shared Task evolved in the context of the UZH Digital Society Initiative project "SDG Research Scout" financed by the Digitalization Initiative of the Zurich Higher Education Institutions (DIZH).

Task 1: Classification at the Level of the 17 SDGs

Motivation

The primary aim of this Shared Task is to evaluate the capability of automated systems in classifying a scientific abstract under the most appropriate SDG. A particular emphasis is placed on the system's proficiency in identifying the best fitting SDG when the specific goals are underrepresented in the dataset, thus ensuring a balanced and comprehensive understanding of the research landscape in relation to the SDGs.

Objective

This Shared Task involves the classification of scientific abstracts based on their relevance to one of the 17 Sustainable Development Goals (SDGs). The objective is to determine the single, most relevant SDG for each abstract. Additionally, if an abstract does not pertain to any of the SDGs, it should be classified under a distinct ‘non-relevant’ category.

Task Details

SDG Single Label Classification: Participants are required to assign one primary SDG to each abstract. This involves identifying the most relevant SDG or classifying the abstract as non-relevant if it does not align with any SDG.

Training Material

A dataset of at least 400 labeled abstracts from the UZH publication repository Zora (zora.uzh.ch) for SDG classification will be provided, facilitating the development of models to predict the most applicable SDG for each abstract. This dataset is unbalanced regarding the 17 SDGs and the topics represented therein.

Evaluation Metrics

Primary Metric: Accuracy. This will assess the correct prediction of the primary SDG for each abstract.

Secondary Metric: Average F1 Score per SDG. This metric is designed to evaluate the balanced performance across all SDGs, providing insight into the model’s ability to accurately classify abstracts under each goal, including those that occur less frequently in the training data.

Additional Resources

Participants are permitted to use any external resources (such as additional services, query collections, or annotated documents such as the large OSDG-CD dataset) to enhance and improve their systems. However, manual correction of system output of the test data is not allowed. It is mandatory for system description papers to clearly specify any additional resources used.

Test Conditions

The participants will get a test set with abstracts. They have to predict the numeric SDG label. The participants will have 2 days to compute and submit the results.

Task 2: Multi-label Classification at the Level of SDG Targets

Task 2 in the Swisstext 2024 Shared Task delves into the more granular aspect of the United Nations' Sustainable Development Goals (SDGs) by focusing on the multi-label classification of SDG targets. Each of the 17 SDGs is underpinned by specific targets that detail the objectives needed to achieve the respective goal. For instance, SDG 3, "Good Health and Well-being," includes targets like reducing maternal mortality and ending epidemics of certain diseases. This task requires participants to predict the most relevant target(s) for a given abstract, based on its classified main SDG from Task 1. If no specific target is relevant for the SDG, a label "unspecific target" should be predicted.

Motivation

This Shared Task is designed to challenge the systems' abilities in fine-grained classification and to encourage the development of innovative solutions in the field of NLP, particularly in scenarios where extensive labeled data is not available. Accurate prediction of SDG targets in scientific abstracts enhances the specificity and usefulness of research classification in the context of sustainable development, fostering a more nuanced understanding of research contributions towards global goals.

Objective

Participants are tasked with predicting the primary target for an abstract associated with a specific SDG. Additionally, where applicable, they should identify other relevant targets within the same SDG. This task aims to deepen the understanding of how specific aspects of research align with detailed objectives of the SDGs.

Task Details

Primary Target Prediction: Given an abstract and its main SDG, the main task is to identify the most relevant target within that goal.

Additional Targets Prediction: If relevant, participants should also predict any number of additional targets that apply to the same SDG for the given abstract.

If the primary label is "unspecific target", no additional targets should be predicted.

Training and Development Material

No specific training data will be provided for target-level classification.

Participants can use the labeled data from Task 1 to select the SDG.

A development set consisting of 50 abstracts with annotated targets will be available for tuning and testing the approaches.

Approach

Given the limited amount of development data, participants are encouraged to explore zero-shot or few-shot learning techniques, particularly prompt engineering, to effectively approach this task. These advanced NLP methodologies are anticipated to be central in participants' solutions due to the task's nature and the data constraints.

Evaluation Metric

The F1 Score will be used to evaluate multi-label target predictions in Task 2. This metric effectively balances precision and recall, assessing model accuracy through true positives, false positives, and false negatives against a manually corrected ground truth.

Test Conditions

The participants will get a test set with abstracts and a provided SDG. They have to predict the main numeric target label and any number of additional target labels for the given SDG of the abstract. The participants will have 2 days to compute and submit the results.

Schedule, System Descriptions and Planned Workshop

System Description Papers

System description papers by participants are required in order to gain insights into which approaches work best for this problem.

SwissText Workshop “NLP for Sustainable Development Goals Monitoring”

A 2-hour workshop, allowing both online and on-site participation, is planned. This workshop aims to bring together task participants and interested stakeholders to delve into the workings of the participating systems. It will focus on identifying the strengths and weaknesses of the proposed approaches, fostering a collaborative environment for open discussion. The workshop will also be a platform to discuss future directions for the research community, targeting enhancements in the automatic classification of SDG-related aspects in scientific work. Attendees will have the opportunity to engage in knowledge sharing, contributing to a collective understanding and shaping the next steps in this application area of research.

Schedule

1. Welcome + Shared-Task Overview: Simon Clematide + Tobias Fankhauser
(15-20 min)
2. Jürgen Bernard (UZH): The DZI Project “SDG Research Scout” and its
visualisation aspects (20 min)
3. - Short Break -
4. Presentation of the Shared Task contributions (15 min each)
 1. Fernando de Meer Pardo, Hanna Hubarava and Vera Bernhard
(UZH): System Description Paper for SwissText 2024 Shared Task
1: Classification at the Level of the 17 SDGs
 2. Manuel Bolz, Andreas Loizidis, Kevin Bründler (UZH): SwissText
- Shared Task (SDG Classification) - Task 1
 3. Adrian M.P. Brasoveanu, Albert Weichselbraun, Lyndon J.B. Nixon
and Arno Scharl (MODUL
University Vienna/webLyzard): An Efficient Workflow Towards
Improving Classifiers in Low-Resource Settings with Synthetic Data
 4. Norman Süsstrunk, Albert Weichselbraun, Andreas Murk, Roger
Waldvogel and André Glatzl (Chur): Scouting out the Border:
Leveraging Explainable AI to Generate Synthetic Training Data
for SDG Classification
5. Final discussion «NLP for Sustainability» (15 min)

An Efficient Workflow Towards Improving Classifiers in Low-Resource Settings with Synthetic Data

Adrian M.P. Braşoveanu^{1,2}, Albert Weichselbraun^{3,4}, Lyndon J.B. Nixon^{1,2} & Arno Scharl^{1,3}

¹ Modul University Vienna, Am Kahlenberg 1, 1190, Vienna, Austria

² Modul Technology GmbH, Am Kahlenberg 1, 1190, Vienna, Austria

³ webLyzard technology gmbh, Liechtensteinstraße 41/26, 1090 Vienna, Austria

⁴ University of Applied Sciences of the Grisons, Pulvermühlestrasse 57, 7000 Chur, Switzerland

{adrian.brasoveanu,lyndon.nixon,arno.scharl}@modul.ac.at

albert.weichselbraun@fhgr.ch

Abstract

The correct classification of the 17 Sustainable Development Goals (SDG) proposed by the United Nations (UN) is still a challenging and compelling prospect due to the Shared Task’s imbalanced dataset. This paper presents a good method to create a baseline using RoBERTa and data augmentation that offers a good overall performance on this imbalanced dataset. What is interesting to notice is that even though the alignment between synthetic gold and real gold was only marginally better than what would be expected by chance alone, the final scores were still okay.

1 Introduction

Automated SDG classification, the main topic of the SwissText 2024 Shared Task 1, is one of the most interesting research topics in light of the United Nations Agenda 2030. The goals offer a holistic approach to global challenges covering various issues from hunger and lack of healthcare to energy security and well-being. Alignment with these goals helps create targeted policies and investments in critical sectors while simultaneously enhancing accountability and measurements. The goals are the core of promoting global partnerships and cooperation that drive innovation and guide educational and awareness efforts. Additionally, goals are broken down into specific targets (typically between 8 and 12) to make them actionable for public and private organizations around the globe.

The correct classification of the 17 Sustainable Development Goals (SDG) proposed by the United Nations (UN) is still a challenging and compelling prospect, especially when coupled with an imbalanced dataset, as it happens in the SwissText 2024 Shared Task 1. The main objective was to classify scientific abstracts based on their relevance to one of the 17 SDGs or tag them as non-relevant if they cannot be aligned with any of the SDGs. In a

certain sense, due to the addition of non-relevant cases, the task incorporated both classification and alignment, making it more difficult than a pure classification task.

The primary goal of the Shared Task was to evaluate the accuracy and overall performance of automated systems for classifying scientific abstracts with the appropriate SDG classes. An additional class was added to cover situations for which it was difficult to select a proper SDG class. This split between relevant and non-relevant classes was a novel element for this type of classification. A second element that made the task worthy of pursuing was the fact that the training data was quite imbalanced. The task asked for methods to help classify documents in low-resource settings.

The classes with the highest number of examples were the non-relevant class (SDG 0 with 156 examples, approximately one-third of all examples), followed by SDGs 13 (40 examples) and 15 (49 examples). On the lower end of the spectrum, four classes included less than five examples: SDG 4, 11 and 17 with four examples each, and SDG 14 with as few as three. These imbalances between the classes and the lack of training data led to the idea of using data augmentation techniques to improve the results. The selected approach helps mitigate the impact of class imbalance and can work across different datasets.

The rest of the paper is organized as follows: method and related work are presented in Section 2, Section 3 presents the evaluation results, which are then discussed in Section 4. The paper concludes with future work and insights on enhancing the process of automatically classifying imbalanced datasets.

2 Method

The classification required for this Shared Task is built a little bit differently than the usual SDG classification, as it used 18 classes instead of the

expected 17, the additional class being used for non-relevant examples. It was not explicitly defined if these non-relevant examples should be examples that could fit into multiple categories or simply examples that do not belong to any SDG categories. Obviously the second category was much larger than the first one, therefore the non-relevant examples we included in our synthetic datasets mostly belonged to this category. As already mentioned, the high number of non-relevant examples contributed significantly to the severe imbalances found in the dataset.

Based on literature, BERTopic (Grootendorst, 2022) coupled with RoBERTa (Liu et al., 2019) or SetFit (Tunstall et al., 2022) seem to be provide quick and good solutions for any type of classification, especially when MPNet embeddings are used (Sayed et al., 2023).

Due to the fact that the dataset is imbalanced, only several solutions are deemed to perform well, zero-shot or few-shot learning (Brown et al., 2020) or data augmentation (Feng et al., 2021) being the first two that come to mind. Our paper focuses on data augmentation, and given the limited time allocated to the Shared Task, it seemed a rather good option.

We started by looking for a set of topics that could trigger the various SDGs. A recent study by Amel et al. (Amel-Zadeh et al., 2021) suggested a list, but while we examined it, we realized that there was some overlap between the SDGs, as it became quite clear that topics like "innovation", "energy", or even "economic aspects" tend to span across multiple SDGs. While their approach indicates higher scores are possible when applying this method, we wanted a fully automated approach that required no additional lists. This was the main reason we decided to focus on data augmentation.

Since the launch of ChatGPT, the number of articles on data augmentation techniques has increased exponentially. Reviewing all of them in such a short timespan would have been impossible. Therefore we limited ourselves to a study about pre-ChatGPT data augmentation techniques (Feng et al., 2021) and surveying modern articles about the impact of reasoning strategies on NLP task (Wadhwa et al., 2023). We surveyed two ideas related to data augmentation: i) using LLMs to generate synthetic data and ii) using existing datasets for augmentation. Each approach came with its own set of issues. For the LLM approach, it was clear

that the data quality would be an issue, as it could increase or decrease over time. For the second approach, we felt strongly that open datasets like OSDG (Pukelis et al., 2020, 2022) could provide a good solution. While the quality of the datasets is certainly much higher than the current generation of LLMs, and their construction is well covered through their open papers and a series of notebooks, we have quickly uncovered some issues. The annotation process for these datasets followed strict rules, but they did not include out-of-domain cases or non-relevant cases.

After carefully examining both approaches, we decided to test the LLM data augmentation with GPT 4.0 for this shared task, as we wanted to see whether it could help us achieve good results quickly. Drawing upon data augmentation also made the results more interesting, since the chosen approach can be easily adapted to other datasets.

We tested several models before submitting the three requested runs. The examined models included: BERT (Guisiano et al., 2022), SVMs (Morales-Hernández et al., 2022), and RoBERTa (Liu et al., 2019).

The best performing approach extended a RoBERTa base model (125M parameters, similar to BERT) with dropout and fully connected layers. Adding the dropout layer set to 0.5 helped us prevent overfitting. The fully connected layer mapped the model output to the 18 classes. The model was evaluated using a 5-fold stratified cross-validation strategy to ensure a similar number of samples of each target class in each fold. For each fold, the model was trained for five epochs using the AdamW optimizer with a learning rate of 1E-5. The experiments were run in Google Colab Pro using L4 GPUs.

3 Evaluation Results

The synthetic dataset was generated using GPT 4.0 and classic prompts, which included 12 examples after the prompt (e.g., "Please annotate the following documents with their corresponding SDG class") (Wadhwa et al., 2023).

Since the labelled test results were not available upfront, we created a labelled test set using the same procedure we used for the augmentation. While this test set differed significantly from the dataset published after the Shared Task's conclusion, we considered it necessary to help select the best runs for the submission. A discussion related

Table 1: Results for the primary metric — correct prediction for primary SDG ordered by accuracy. P, R, F1 represent precision, recall and F1 metrics.

Run	Accuracy	Macro P	Macro R	Macro F1	Weighted P	Weighted R	Weighted F1
run3	0.49	0.65	0.66	0.56	0.74	0.49	0.53
run1	0.46	0.61	0.58	0.51	0.69	0.46	0.49
run2	0.40	0.58	0.61	0.51	0.64	0.40	0.42

Table 2: Results for the secondary metric - average F1 score per SDG. P, R, F1 represent precision, recall and F1 metrics.

Run	Accuracy	Macro P	Macro R	Macro F1	Weighted P	Weighted R	Weighted F1
run3	0.52	0.68	0.67	0.59	0.75	0.52	0.55
run1	0.50	0.66	0.63	0.58	0.69	0.50	0.53
run2	0.43	0.62	0.63	0.56	0.65	0.43	0.45

to these differences is included in Section 4.

One of the three submitted test runs was found to offer the best average performance on this imbalanced dataset. Although all submissions scored above average in both evaluation settings, the variability between runs suggests that improvements can be made.

Table 1 showcases the performance obtained by the submitted runs (named run31 to run33) for the primary metric, which optimizes for accuracy. Table 2 presents the results obtained for the secondary metric, i.e., the average F1 score per class. One of our runs (run33) obtained the best performance for the secondary metric from all submitted runs. The runs were submitted using the cover name: test_roberta_base_synth_TASK1_RUN31 to RUN33, which included the name of the dataset (test), model (Roberta base), method (synth), task (TASK1), and run (RUN31, RUN32 and RUN33). The evaluation reports can be found on the Shared Task’s GitHub folder ¹.

4 Discussion

We see at least several avenues for improving the data augmentation strategies. Perhaps the most obvious one is using a modern reasoning strategy like Chain-of-Thought (CoT) (Wei et al., 2022). Adding a justification for each example generated by the LLM would have further improved the quality of the synthetic dataset and, therefore, led to even better classification results (Wadhwa et al., 2023).

¹https://github.com/ZurichNLP/sdg_swisstext_2024_sharedtask/tree/main/evaluation

Such techniques are known to work better for LLMs or larger Transformer models, which tend to generalize better. Consequently, we didn’t use this approach in our experiments, as we relied upon the smaller roberta-base model (125M parameters) due to time restrictions.

Given the counts for the relevant (SDGs 1 to 17) and non-relevant (SDGs marked as 0) classes, the classification results would have been considerably better if we had started with a binary classifier to separate relevant from non-relevant classes.

As outlined in Table 3, more than half of the test examples were non-relevant. This severely skewed the results since LLMs tend to overfit. In fact, an LLM will not be able to reproduce this setting a priori unless it is made more transparent through a detailed prompt (e.g., by adding a line like: “Please be aware that half of the examples I will ask you to annotate will be non-relevant”).

An evaluation of the augmented data revealed that the synthetic dataset was missing the non-relevant class. The LLMs failed to produce examples for the non-relevant class despite being instructed to provide examples for all 18 classes (which includes class 0 for non-relevant cases). This error suggests that data augmentation is still the way to go, as even with all the errors that followed, the results were still the most balanced.

We assume that in a real-world setting, non-relevant entries will likely be even more prominent than in the provided dataset, which amounted to approximately one-third of the provided training examples and one-half of the provided test examples. In addition, overlaps between various classes

are probably also more likely in a production setting.

While the length of the abstract was not considered a key parameter for our prompt, it is important to notice that the generated abstracts from the synthetic dataset were, on average, shorter than the ones from the real repository used for collecting the abstracts for this task. This suggests that the prompts need to be further refined to consider this aspect.

Table 3: Alignment between Gold and Synth Gold SDG Counts.

SDG	Gold	Synth Gold	Difference
0	83	12	71
1	3	2	1
2	4	5	1
3	3	25	22
4	6	10	4
5	4	7	3
6	4	5	1
7	3	6	3
8	5	24	19
9	5	12	7
10	4	8	4
11	4	4	0
12	6	4	2
13	2	8	6
14	5	3	2
15	5	7	2
16	3	1	2
17	7	2	5

5 Future Work

Future work will focus on improving the data augmentation strategies. The top priority will be creating synthetic datasets that are closer to the train and test distributions. Some other datasets should be based on classic distributions (e.g., multinomial, Poisson, log-normal, etc.). Pairing existing SDG datasets with non-relevant examples generated by LLMs could be another viable strategy to improve the training data. We also plan to test on multiple SDG datasets using the same approach.

Limitations

A main limitation of the presented approach was its failure to generate synthetic examples for the

zero (non-relevant) class. Likely, even a few non-relevant examples in the synthetic dataset would have further improved the results. Another major shortcoming is that we have not considered various data distributions for this particular set of runs. This limitation will be addressed in future work, as already mentioned. Processing speed was another significant problem since a typical 5-fold stratified cross-validation with 5 epochs per fold took over 10 minutes to run, which could be considered a bit too much given the size of the training and test datasets (430 and 156 examples).

References

- Amir Amel-Zadeh, Mike Chen, George Mussalli, and Michael Weinberg. 2021. Nlp for sdgs: Measuring corporate alignment with the sustainable development goals. *Columbia Business School Research Paper*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. [A survey of data augmentation approaches for NLP](#). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 968–988. Association for Computational Linguistics.
- Maarten Grootendorst. 2022. [Bertopic: Neural topic modeling with a class-based TF-IDF procedure](#). *CoRR*, abs/2203.05794.
- Jade Eva Guisiano, Raja Chiky, and Jonathas De Mello. 2022. [Sdg-meter: A deep learning based tool for automatic text classification of the sustainable development goals](#). In *Intelligent Information and Database Systems - 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28-30, 2022, Proceedings, Part I*, volume 13757 of *Lecture Notes in Computer Science*, pages 259–271. Springer.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

- Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Roberto Carlos Morales-Hernández, David Becerra-Alonso, Eduardo Romero Vivas, and Joaquín Gutiérrez. 2022. [Comparison between SVM and distilbert for multi-label text classification of scientific papers aligned with sustainable development goals](#). In *Advances in Computational Intelligence - 21st Mexican International Conference on Artificial Intelligence, MICAI 2022, Monterrey, Mexico, October 24-29, 2022, Proceedings, Part II*, volume 13613 of *Lecture Notes in Computer Science*, pages 57–67. Springer.
- Lukas Pukelis, Núria Bautista-Puig, Mykola Skrynik, and Vilius Stanciasukas. 2020. [OSDG - open-source approach to classify text data by UN sustainable development goals \(sdgs\)](#). *CoRR*, abs/2005.14569.
- Lukas Pukelis, Núria Bautista-Puig, Guste Statulevičiute, Vilius Stanciasukas, Gokhan Dikmener, and Dina Akylbekova. 2022. [OSDG 2.0: a multilingual tool for classifying text data by UN sustainable development goals \(sdgs\)](#). *CoRR*, abs/2211.11252.
- Mohamad Al Sayed, Adrian M. P. Brasoveanu, Lyndon J. B. Nixon, and Arno Scharl. 2023. [Unsupervised topic modeling with bertopic for coarse and fine-grained news classification](#). In *Advances in Computational Intelligence - 17th International Work-Conference on Artificial Neural Networks, IWANN 2023, Ponta Delgada, Portugal, June 19-21, 2023, Proceedings, Part I*, volume 14134 of *Lecture Notes in Computer Science*, pages 162–174. Springer.
- Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. [Efficient few-shot learning without prompts](#). *CoRR*, abs/2209.11055.
- Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. [Revisiting relation extraction in the era of large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15566–15589. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

MeHuBe at SwissText 2024 Shared Task 1: Ensembling and QLoRA with Retrieved Citations for Fine-Grained Classification of Sustainable Development Goals

Fernando de Meer Pardo **Hanna Hubarava** **Vera Bernhard**
University of Zurich University of Zurich University of Zurich
fernando.demeerpardo@uzh.ch hanna.hubarava@uzh.ch vera.lara.bernhard@uzh.ch

Abstract

This study is a contribution to the SwissText Shared Task 2024, aimed at an improvement of the automatic classification of scientific abstracts related to the United Nations' Sustainable Development Goals (SDGs). Using Semantic Scholar API, we augment the small and highly imbalanced training set by retrieving additional abstracts from citations of the original dataset. The enriched dataset is then used to fine-tune a number of BERT-based models as well as the Mistral-7B model, which is fine-tuned in a parameter-efficient way with QLoRA. Experimentation with various ensembling strategies reveals a positive influence of prediction confidence, with the best ensembling strategy yielding the accuracy of 77% and macro F1 of 77%. The submission occupied place 10 when evaluated on the Shared Task's test set.

Keywords: Sustainable Development Goals, SDG, NLP, QLoRA, BERT, ensembling, text classification, data augmentation

1 Introduction

This study is a team submission to Task 1 of the SwissText Shared Task 2024¹. The Shared Task is a part of the Swiss Text Analytics Conference. The goal of the shared task is to improve upon the existing ways of automatic classification of scientific abstracts related to the United Nations' Sustainable Development Goals (SDGs)².

Natural Language Processing (NLP) has demonstrated significant potential in addressing various socio-economic challenges, from healthcare and education to climate change: For example, BERT models have been successfully used for identifying geographic entities in climate literature which enables monitoring evolution of climate issues (Mallick et al., 2024). Additionally, larger LLMs

¹see Appendix A for our code implementation and the software specifications used for this submission.

²see <https://sdgs.un.org/goals>

like Mistral-7B has been effectively applied to tasks such as medical dialogue systems (Zhao et al., 2024).

By leveraging the tools offered by Natural Language Processing, the Shared Task aims to build a better approach towards identification and analysis of academic research pertinent to sustainable development. This, in its turn, should help to both quantify and leverage the impact of academia on the progress in achieving these global goals.

The key contributions and findings of this study are the following:

- We study the effect of augmenting the SDG training data with weakly labeled citations fetched via the Semantic Scholar API.
- We fine-tune and evaluate 4 different BERT variants and an adapter for Mistral-7B with QLoRA.
- We experiment with various ensembling techniques for all the models we fine-tuned.

2 Dataset

2.1 Dataset Description

Task 1 focusses on the classification at the level of the SDGs, where each abstract is to be mapped to a single SDG number (class label). There are 18 classes in total: 17 SDGs and an additional class zero ("non-relevant"). The latter is reserved for the articles that should not be classified as relevant to any of the SDGs, posing a challenge and demanding a creative solution.

The final version of the train set includes 430 entries in jsonl format from the UZH publication repository Zora. It contains Zora ID, title, abstract, URL and a target SDG number. The dataset is extremely unbalanced, with nearly half of the train set represented by the four largest classes (life on land, good health and well-being, climate action, and non-relevant).

In addition to the train set provided by the Task organizers, there exists the OSDG Community Dataset (OSDG et al., 2021), which appears to be a popular resource in the research on automatic SDG classification (Pukelis et al., 2022).

2.2 Dataset Modifications

The train set history on GitHub³ reveals two rounds of modifications between the task release and the submission deadline. The modifications of the train set during the Shared Task had a twofold impact: Firstly, they caused the participants to reconsider gold standard labels for the modified records and the features which could be most informative. Secondly, they pointed towards a potential lack of consensus among annotators regarding some problematic classes.

Crucially, the ad hoc modifications raised a concern whether the final split into the train and test was done truly randomly, i.e. preserving the original class distribution and in accordance with the same annotation guidelines. A chi-square test confirmed our suspicion that the train and test set are extremely unlikely to come from the same distribution (Chi2: 63.6, p-value: 2.63×10^{-7}). See Figure 2 and Appendix D for a more detailed analysis.

3 Methods

3.1 Data Augmentation

In order to acquire more data to fine-tune our models with, we resorted to the Semantic Scholar API (Kinney et al., 2023). For each original record, we gathered all of its citations and generated new records by concatenating their titles and abstracts. Each new record was then assigned the same label as the work it referenced, without any manual supervision.

This weak labeling approach carries some risks. Although scientific papers usually cite thematically related manuscripts as part of their related work, there remains a possibility that such overlaps may be unrelated to the SDG labels (see Section 6).

We hypothesize that following along "citation paths", we can better capture the trends within SDGs. For example, out of the 17 records labeled with SDG 7 (renewable energy) in the provided train set, 15 concern solar energy and water splitting, only one publication is on the topic of adaptive energy consumption, while the remaining record is

³https://github.com/ZurichNLP/sdg_swisstext_2024_sharedtask

related to cloud networks. Crucially, the majority of those records were published in 2018-2019, ignoring the developments and themes of other years and decades. By including the publications that build upon previous research, we strive to harvest the evolution of topics over time and thus mitigate the limitations of the train set.

3.2 Models

3.2.1 BERT Variants

The following BERT variants were selected for fine-tuning:

- **Multilingual BERT**: BERT pretrained on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective (Devlin et al., 2018).
- **SciBERT**: BERT pretrained on a corpus of 1.14M papers, 3.1B tokens built through the Semantic Scholar API. Additionally, it has its own vocabulary (scivocab) fit to match the training corpus (Beltagy et al., 2019).
- **Aspect-Based SciBERT**: BERT pretrained to perform pairwise document classification on a dataset of citing and cited papers originating from the ACL Anthology (Ostendorff et al., 2020).
- **BioBERT**: BERT pre-trained on large-scale biomedical corpora consisting on PubMed abstracts and PMC full-text articles (Lee et al., 2019).

The selection of these models took into account the type of data used in their pretraining, which should make the models efficient at processing scientific and technical texts in our task.

3.2.2 Mistral-7B with QLoRA

Due to promising results from manual zero-shot experiment using Mistral-7B on the SDG classification tasks, we decided to leverage the semantic knowledge of this large language model and fine-tuned it on the classification task (Jiang et al., 2023). Since our available resources were limited to a single T4 GPU (see Appendix C), we opted for the parameter efficient fine-tuning approach QLoRA (Dettmers et al., 2024): It injects trainable low rank adapters into a frozen, quantized large language model and enables memory efficient fine-tuning by introducing a 4-bit float data type, using double quantization and preventing memory spikes.

3.2.3 Ensembling

We experimented with the following ensembling methods:

- **Majority Voting:** The class voted for by the majority of the models in the ensemble is selected. In the case of a tie, the first one of the tied classes is chosen as a default.
- **Majority Voting with Tie Breakup via Logits:** Same as above, except ties are broken via adding the logits of the tied classes and choosing the one with the largest sum.
- **Soft Voting:** Adds all the logits of all the models and chooses the class with the largest sum.

4 Experiments

To be able to validate the performance of fine-tuned models, we performed a stratified partition across the SDG labels of the SwissText dataset into a train/test split with 1/3 and 2/3 of the records, respectively. The stratification ensures all SDG classes are present across the splits and their proportions are preserved.

In the following subsections, we describe the setup of each of the different experiments we carried out and eventually led us to our final methodology. We include the results of each experiment and the respective discussion in Section 5.

4.1 Data Augmentation Experiment

In order to estimate the effect on performance of the additional records described in Section 3.1 we fine-tuned the **Multilingual BERT** model with the following sets of records:

- **SwissText records:** Train split of the SwissText original records as described in Section 4.
- **Enlarged SwissText:** All the citations of SwissText records obtained via the Data Augmentation procedure as described in Section 3.1.
- **Original and Enlarged SwissText and OSDG:** Combination of the two datasets described above along with all of the OSDG records with an agreement score bigger than 0.5 and their citations.

Both enlarged datasets are heavily unbalanced, as the total number of citations is not equal across SDGs. In order to have balanced training datasets, we randomly sampled at most 1000 records for each SDG label, leading to datasets of 17k records in both instances.

4.2 Experiments with BERT Variants

We fine-tuned all BERT variants described in Section 3.2.1 for 5 epochs. In all cases we observed an increase in accuracy for each epoch. We report the results of the 5th epoch for all models. See Appendix B for the full set of fine-tuning hyperparameters employed.

4.3 Experiments with QLoRA

We fine-tuned an adapter for Mistral-7B with QLoRA for 4 epochs on the full combination of records. The hyperparameters and infrastructure specification can be found in Appendix C.

4.4 Experiments with Ensembling

As specified in Section 3.2.3, we ensemble all the fine-tuned models listed in 3.2.1 and 3.2.2. Having observed a positive correlation between prediction confidence and F1-score (see Figure 1), we opted for the inclusion of confidence (logits) into some of the ensembling experiments.

5 Results & Discussion

In this section, we discuss the results of each of the experiments previously discussed. All of the scores correspond to our test split (containing 2/3 of the train records) of the SwissText records described in Section 4⁴.

5.1 Main Results

5.1.1 Data Augmentation Experiment

Our hypothesis, as described in 3.1, posited that additional labeled data could be retrieved via citations. This hypothesis finds support in the performance statistics, as measured on 2/3 of the original SwissText data, held out as our test set: As shown in Table 1, the overall accuracy increased by 23 percentage points and the macro F1 score even

⁴The SwissText dataset was modified with multiple records being relabeled during the course of development. As a consequence all scores should only be considered as orientative, some scores were obtained in the original and others on the relabeled version of the dataset, see Appendix D for details. We did not re-run all experiments due to time and hardware constraints.

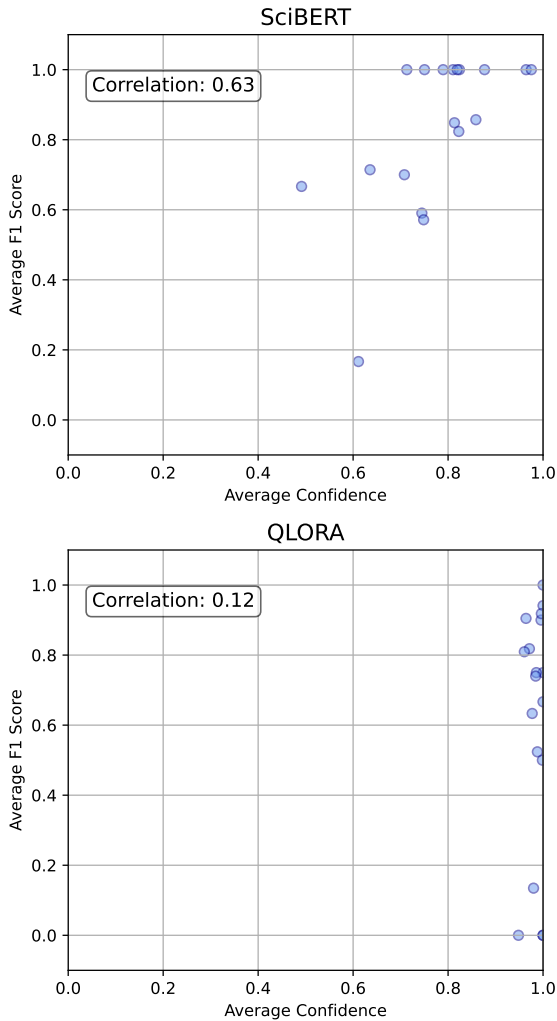


Figure 1: A positive correlation between average per-class F1 score and confidence of SciBERT suggest it could be beneficial to take logits into account during ensembling. QLoRA’s extreme confidence in all predictions indicates it could outvote other models in the ensemble, if its logits are taken into account.

by 53 percentage points when the training set was augmented. Adding the OSDG records⁵ and their citations further improved the overall accuracy by additional 8 percentage points.

5.1.2 Experiments with BERT variants

The four different BERT variants described in 3.2.1 were trained on the augmented dataset (original, enlarged SwissText, OSDG). Table 1 demonstrates that SciBERT performed the best, achieving an accuracy of 73% and macro F1 of 72%. The other BERT models followed closely, with Aspect-based SciBERT even surpassing SciBERT in terms of average precision.

⁵With an agreement score > 0.5.

5.1.3 Experiments with QLoRA

Despite the promising results of QLoRA reported by Dettmers et al., 2024, fine-tuning Mistral-7B on the augmented dataset did not achieve the performance level of BERT variants, lagging 21 percentage points behind the accuracy of SciBERT (see Table 1). This performance gap may be due to our limited resources for more extensive experimentation with QLoRA (see Section 6).

5.1.4 Experiments with Ensembling

Since the fine-tuned BERT variants outperformed Mistral-7B fine-tuned with QLoRA, they were ensembled via majority voting, resulting in slight increase of F1 by 1 percentage point compared to SciBERT, the best performing individual model (see Table 1). The experimental addition of QLoRA to the ensemble increased the accuracy to 75% and the macro F1 to 74%.

Leveraging the correlation between prediction probability and F1 score (as shown in Section 4.4) by breaking up ties via probabilities, the F1 score and accuracy rose to 76%-77%, depending on the inclusion of QLoRA.

Since QLoRA tends to predict with overly high confidence (see Figure 1), we applied soft voting only to an ensemble of the BERT variants, achieving performance comparable to majority voting with logits.

In sum, the experiments involving ensembling (see Table 1) demonstrate that ensembling and incorporating logits into the voting strategy is effective.

5.2 Error Analysis

As pointed out in Section 2.1, the provided train dataset is highly unbalanced. Some classes, either due to topical overlaps or a lack of detailed annotation guidelines (see Section 2.2), are particularly challenging to classify both for human annotators and the models (see confusion matrices in Section F of the Appendix).

Classes 0 (non-relevant), 8 (decent work and economic growth), 9 (industry, innovation and infrastructure), 16 (peace, justice and strong institutions) and 17 (global partnership for sustainable development goals) yield the lowest results across all models and ensembling configurations. SDG 9 (industry, innovation and infrastructure), for example, most commonly gets confused with SDG 12 (responsible consumption and production), reflecting

Model	Acc.	Pre.	Rec.	F1
Data Augmentation Experiment				
SwissText	0.37	0.08	0.06	0.03
SwissText big	0.60	0.51	0.70	0.56
SwissText + SwissText big + OSDG	0.68	0.59	0.70	0.62
Experiments with BERT variants				
mBert	0.68	0.59	0.70	0.62
SciBERT	0.73	0.68	0.83	0.72
Aspect SciBERT	0.72	0.71	0.76	0.70
BioBert	0.69	0.62	0.71	0.63
Experiment with QLoRA				
Mistral-7B	0.52	0.46	0.61	0.48
Experiments with Ensembling				
Majority w/o QLoRA	0.73	0.72	0.79	0.73
Majority with QLoRA	0.75	0.75	0.81	0.74
*Majority + Logit w/o QLoRA	0.76	0.76	0.83	0.77
*Majority + Logit with QLoRA	0.77	0.74	0.83	0.76
*Soft voting w/o QLoRA	0.76	0.75	0.82	0.76

Table 1: Results of all experiments. *Acc.* stands for overall accuracy, *Pre.*, for the precision averaged over all labels *Rec.*, for the recall averaged over all labels, *F1* for macro F1, i.e. F1 averaged over all labels. *SwissText big* is used synonymously for SwissText enlarged. **Bold** signifies the highest values within one experiment. The * marks the configurations chosen for the final submission. All experiments were evaluated on the stratified 2/3 of the original SwissText dataset, held out as our test set.

the semantic overlaps between industry, production and consumption. The confusion between the non-relevant class and SDG 3 (good health and well-being), analysed in more detail in the context of dataset modifications (see Section D of the Appendix), is also visible in model classification results, suggesting a more transparent distinction between medical science and general health topics would benefit the annotation.

5.3 Final Submission

Our final submission comprised three prediction runs with the following components:

- *MeHuBe_RUN1*: Ensemble with QLoRA using majority voting with logits.
- *MeHuBe_RUN2*: Ensemble without QLoRA using majority voting with logits.
- *MeHuBe_RUN3*: Ensemble without QLoRA using soft voting.

The above ensembles showed the most promising results when fine-tuned on the enriched dataset +

1/3 of the train data and evaluated on 2/3 of the train data, see Table 1.

The results of our runs on the test data of the shared task are shown in Table 2. With this results, we achieved place 10 in the Shared Task. The performance of our models on the shared task dataset was markedly lower than on the training dataset, reaching accuracies around 40% and F1 score of maximally 47%.

One reason for this performance gap may be the notable difference in non-relevant data proportions between the two sets. While non-relevant abstract built around a third of the original train set, they represented nearly a half of the test set (see Figure 2). As noted in Section 3.1, classifying class 0 was particularly susceptible to annotation changes, with examples in Appendix D demonstrating incomprehensible decisions, suggesting broader issues with class 0. Furthermore, our models exhibited difficulties with class 0 already during the training phase, as documented in Section 5.2.

Our approach showed a better performance in

Run	Acc.	F1
Evaluation 1: main SDG		
<i>MeHuBe_RUN1</i>	0.39	0.42
<i>MeHuBe_RUN2</i>	0.38	0.41
<i>MeHuBe_RUN3</i>	0.38	0.41
Evaluation 2: secondary SDG		
<i>MeHuBe_RUN1</i>	0.42	0.47
<i>MeHuBe_RUN2</i>	0.42	0.45
<i>MeHuBe_RUN3</i>	0.41	0.45

Table 2: Results on the shared task test data. *Acc.* stands for overall accuracy, *F1* for macro F1, i.e. F1 averaged over all labels.

the evaluation 2 where secondary SDGs were also taken into consideration (see 2. This indicates that our approach occasionally selected secondary SDGs, a phenomenon observed in our error analysis (see Section 5.2), which identified topical overlaps in the dataset. Across the three runs, there was no notable performance disparity; they exhibited comparable results, mirroring our observations during training data evaluation.

6 Limitations

The scope of technical experimentation was restricted by limited computational resources. This pushed us towards fine-tuning compact BERT-based models, as well as parameter-efficient fine-tuning of Mistral-7B with QLoRA. With each QLoRA epoch taking around 18 hours, we were forced to refrain from exploring the effect of various dataset types and hyperparameters on the model performance.

As a consequence of the extremely unbalanced dataset, no cross-validation could be employed, since the smallest classes contained very few records and could not be split into k-folds. No hyperparameter optimization took place when fine-tuning BERT-based models, which should be taken into account if conducting further experiments.

The study would have also benefited from a thorough qualitative analysis of the records obtained via data augmentation. We hypothesize that the results could have been further improved through better preprocessing as well as pruning of the enriched dataset from irrelevant entries. Importantly, dataset enrichment with studies citing the given record only yields papers published later on. To gather trends and topics which preceded the record, one would also need to collect the publications

referenced by that given record.

Due to a late publication of the gold standard labels shortly before the paper submission deadline, no qualitative analysis of the test set errors could be carried out. Lastly, the lack of access to the annotator guidelines limited our understanding of the definitions of classes. An analysis of errors and modifications of the train set points towards inconsistencies in labeling of some abstracts, e.g. those related to medicine and healthcare.

7 Conclusions

Our submission aims at implementing automatic classification of scientific abstracts related to the UN Sustainable Development Goals (SDGs). Leveraging an experimental data augmentation technique, specifically by integrating citations from the original dataset, we enriched the training data and subsequently improved classification performance. Ensembling various fine-tuned BERT-based models, including Mistral-7B fine-tuned with QLoRA, notably improved accuracy and F1 scores on the training data. Using QLoRA provided an interesting proof of concept of fine-tuning larger language models, yet its utility requires further experimentation in a setting with greater computational power. The modest performance of our approach highlights the complexity of SDG classification, particularly identifying abstracts as relevant and non-relevant and modeling it as a single-label task.

Acknowledgements

This paper was written not only as a submission for the Shared Task at SwissText 2024 but also as part of the course Machine Learning for Natural Language Processing 2 at the University of Zurich. We therefore thank the the University of Zurich for providing both the computational infrastructure and valuable feedback throughout the project, and the University of Applied Sciences (ZHAW) for providing supplementary GPU access.

References

- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A Pretrained Language Model for Scientific Text](#). *arXiv e-prints*, page arXiv:1903.10676.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."

- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). *arXiv e-prints*, page arXiv:1810.04805.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David W. Graham, F.Q. Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Christopher Newell, Smita R Rao, Shaurya Rohatgi, Paul Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, A. Tanaka, Alex D Wade, Linda M. Wagner, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine van Zuylen, and Daniel S. Weld. 2023. [The semantic scholar open data platform](#). *ArXiv*, abs/2301.10140.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [BioBERT: a pre-trained biomedical language representation model for biomedical text mining](#). *arXiv e-prints*, page arXiv:1901.08746.
- Tanwi Mallick, John Murphy, Joshua David Bergerson, Duane R Verner, John K Hutchison, and Leslie-Anne Levy. 2024. Analyzing regional impacts of climate change using natural language processing techniques. *arXiv preprint arXiv:2401.06817*.
- OSDG, UNDP IICPSD SDG AI Lab, and PPMI. 2021. [Osdg community dataset \(osdg-cd\)](#).
- Malte Ostendorff, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm. 2020. [Aspect-based Document Similarity for Research Papers](#). *arXiv e-prints*, page arXiv:2010.06395.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.
- Lukas Pukelis, Nuria Bautista-Puig, Gustė Statulevičiūtė, Vilius Stančiauskas, Gokhan Dikmener, and Dina Akylbekova. 2022. [OSDG 2.0: a multilingual tool for classifying text data by UN Sustainable Development Goals \(SDGs\)](#). ArXiv:2211.11252 [cs] version: 1.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Zihao Zhao, Sheng Wang, Jinchun Gu, Yitao Zhu, Lanzhu Mei, Zixu Zhuang, Zhiming Cui, Qian Wang, and Dinggang Shen. 2024. [Chatcad+: Towards a universal and reliable interactive cad using llms](#). *IEEE Transactions on Medical Imaging*, pages 1–1.

A Software and Code

The code to train and evaluate our results can be found on GitHub: <https://github.com/vera-bernhard/SwissTextSDG>

In addition, the following Python packages have been used:

- PyTorch, pytorch-transformers (Paszke et al., 2017)
- Hugging Face’s transformer, accelerate, peft, bitsandbytes (Wolf et al., 2019)
- NLTK (Bird et al., 2009)

B Hyperparameters of BERT Variants Training

We fine-tuned the following pretrained Bert variants from Hugging Face (Wolf et al., 2019) with the hyperparameters as specified in Table 3:

bert-base-multilingual-uncased,
dmis-lab/biobert-base-cased-v1.2,
allenai/scibert_scivocab_uncased,
malteos/aspect-acl-scibert-scivocab-uncased

Fine-tuning each model for 5 epochs on the enlarged datasets required 1 hour of GPU time, utilizing a Tesla T4 GPU (16GB).

C QLoRA Training: Hyperparameters and Infrastructure

We had access to a single Tesla T4 GPU (16GB) and trained QLoRA with the hyperparameters in Table 4. The first training experiment took ~74h for 4 epochs, while training on the full dataset for the same number of epochs required ~95h.

Hyperparameter	Value
Learning Rate	2e-5
Learning Rate Scheduler	Warmup-LinearScheduler
Batch Size	32
Adam Epsilon	1e-8
Number of Epochs	5
Max Gradient Norm	1.0
Max Sequence Length	265

Table 3: Hyperparameters for BERT Variants Training

Hyperparameter	Value
General Training Parameters	
Learning Rate	2e-5
Learning Rate Scheduler	WarmupLinearScheduler
Batch Size	1
Adam Epsilon	1e-8
Number of Epochs	4
Max Gradient Norm	1.0
Max Sequence Length	265
PEFT Parameters	
Pretrained Model	mistralai/Mistral-7B-v0.1
Lora alpha	16
Lora dropout	0.1
Rank of low-rank	2
Total # Parameters	7,111,659,520
Trainable # Parameters	925,696

Table 4: Hyperparameters for QLoRA Training

D Dataset Modifications

The modifications in early March encompassed nine label changes, while major modifications of over 300 lines took place in April. The problematic classes which caused the largest number of modifications are the following: SDG 0 (non-relevant), 3 (good health and well-being), 5 (gender equality), and 16 (peace, justice and strong institutions). A notably large share of modifications concerned the non-relevant class (SDG 0), with seven out of nine first-round modifications related to this class.

Record 130 of the last train set edition serves as an example of the problematic classes and modifications: “Newspaper coverage of female candidates during election campaigns: evidence from a structural topic model”). The article is clearly concerned with gender inequality and bias, yet it was modified to be labeled as SDG 0 (non-relevant). Record 23 exemplifies another case of the modification towards non-relevant class. The article titled "Internal auditory canal volume in normal and mal-

formed inner ears" is a medical publication on the topic of hearing abnormalities, yet it was modified as belonging to SDG 0.

E Correlation between Confidence and F1-score

A positive correlation between average per-class prediction confidence (logits) and macro F1-score could be observed in all BERT-based models (see Figures 1 and 3). As specified in Section 4.4, this observation led us to include confidence into some ensembling configurations, resulting in an improved performance.

F Confusion Matrices

The confusion matrices presented below are those of the best-performing model (SciBERT) and ensembling strategy (majority voting with QLoRA and logits). The models were trained on the enriched dataset and evaluated on the remaining 2/3 of the train set.

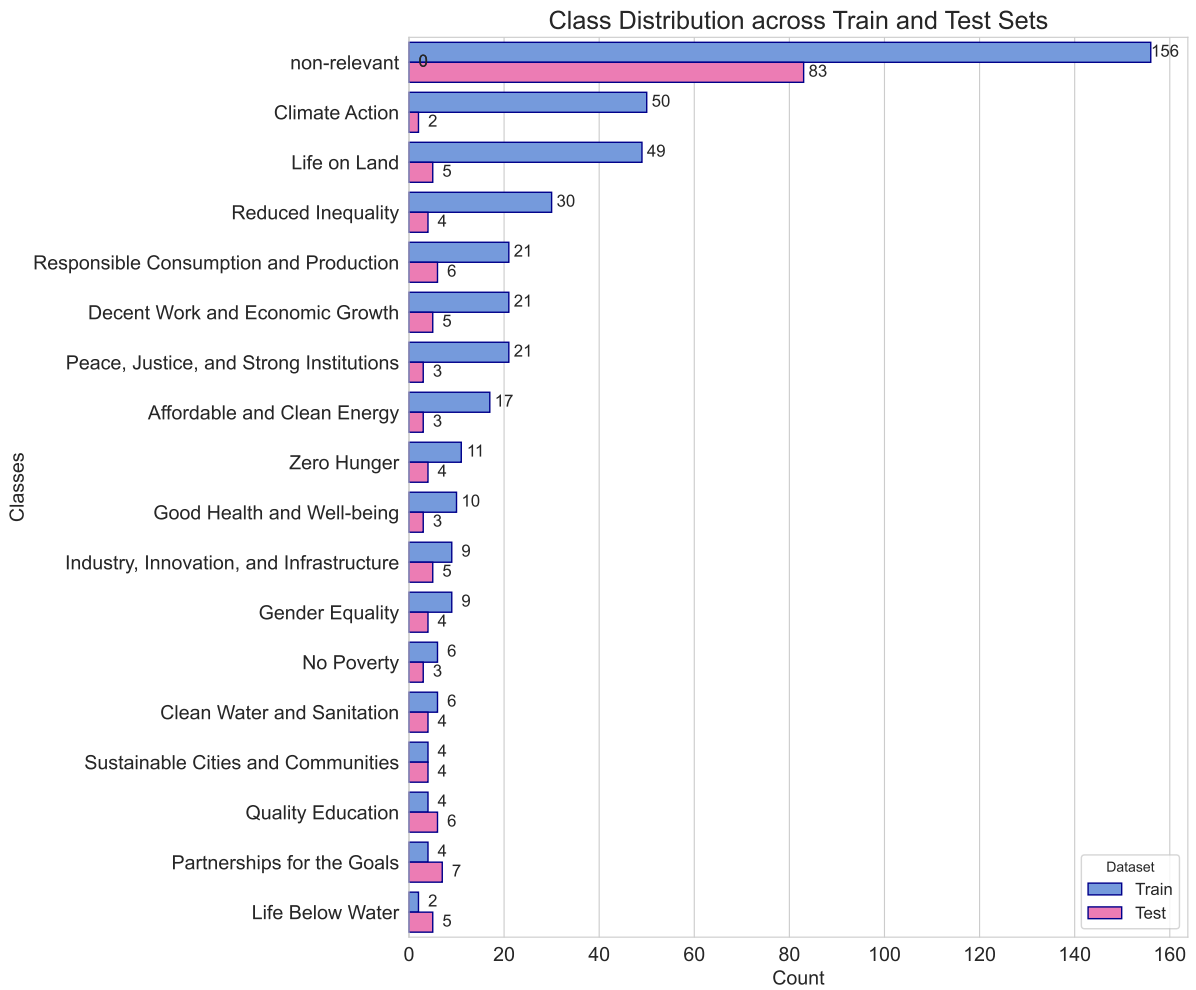


Figure 2: Class sizes in the train and test sets provided by the Shared Task, in decreasing order of class size in the train set.

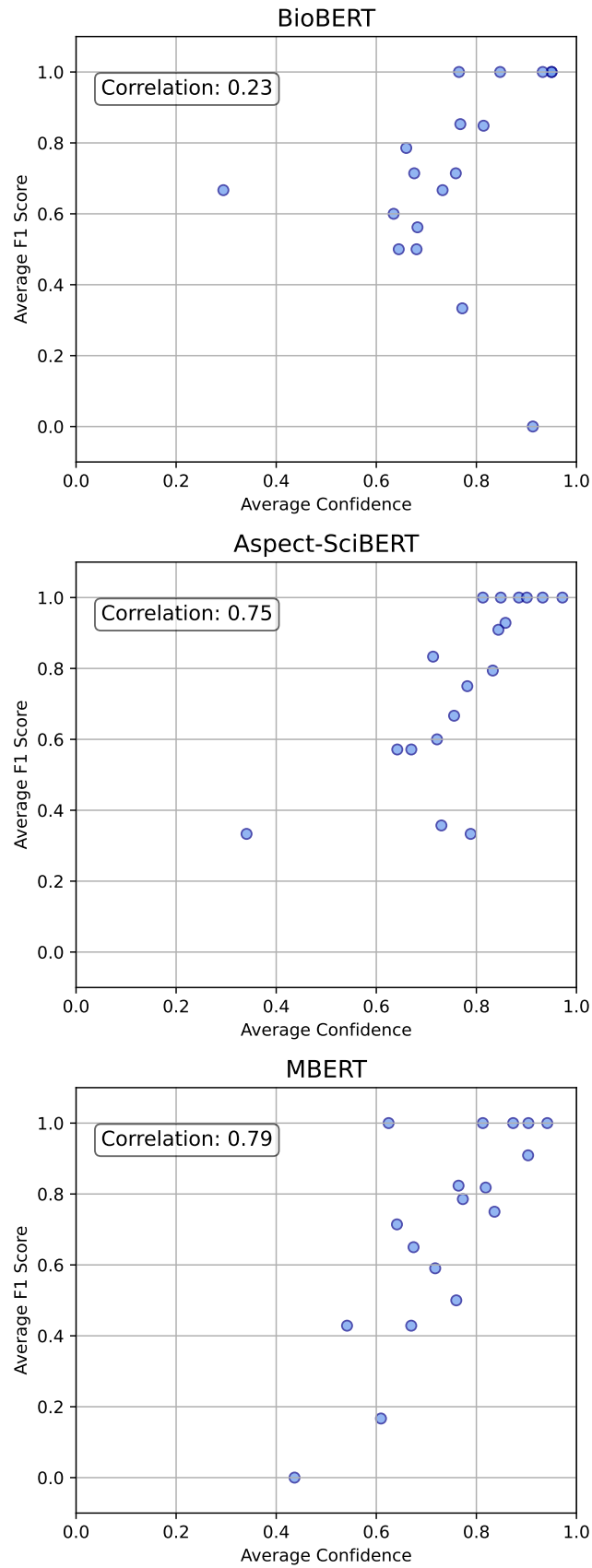


Figure 3: A positive correlation between model's prediction average per-class confidence and F1-score can be observed in all BERT-based models.

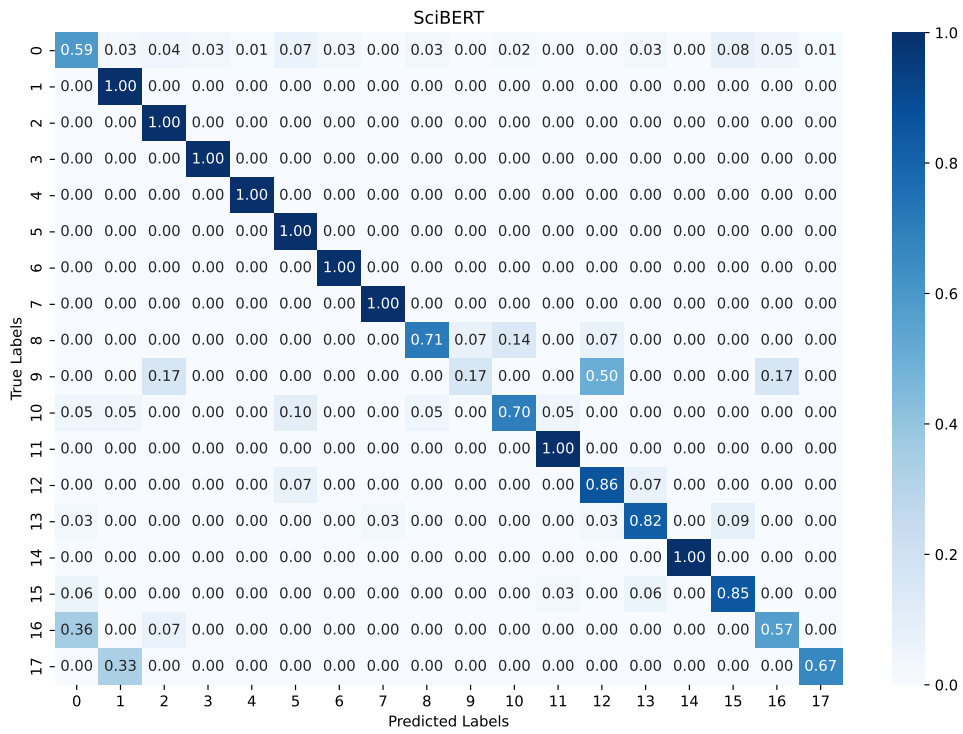


Figure 4: Normalized confusion matrix of SciBERT, the best performing individual model.

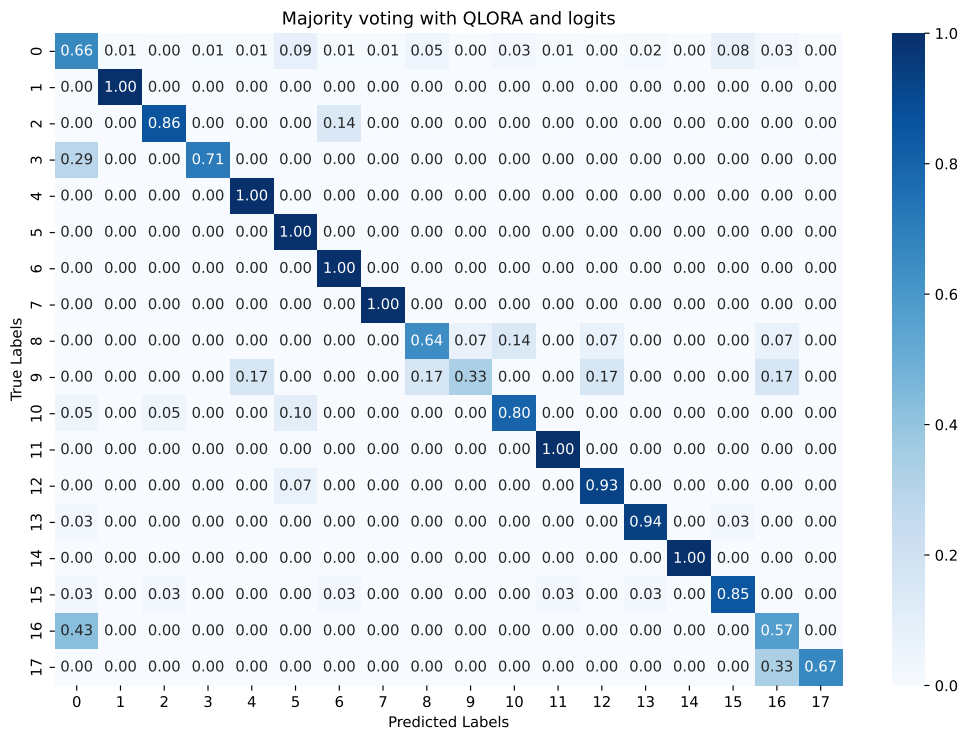


Figure 5: Normalized confusion matrix of the best ensembling strategy, including all BERT-based models and QLoRA and taking into account logits.

Scouting out the Border: Leveraging Explainable AI to Generate Synthetic Training Data for SDG Classification

Norman Süsstrunk and Albert Weichselbraun and Andreas Murk

Roger Waldvogel and André Glatzl

University of Applied Sciences of the Grisons

7000 Chur, Switzerland

{firstname.lastname}@fhgr.ch

Abstract

This paper discusses the use of synthetic training data towards training and optimizing a DistilBERT-based classifier for the SwissText 2024 Shared Task which focused on the classification of the United Nation’s Sustainable Development Goals (SDGs) in scientific abstracts. The proposed approach uses Large Language Models (LLMs) to generate synthetic training data based on the test data provided by the shared task organizers. We then train a classifier on the synthetic dataset, evaluate the system on gold standard data, and use explainable AI to extract problematic features that caused incorrect classifications. Generating synthetic data that demonstrates the use of the problematic features within the correct class, aids the system in learning based on its past mistakes. An evaluation demonstrates that the suggested approach significantly improves classification performance, yielding the best result for Shared Task 1 according to the accuracy performance metric.

1 Introduction

The United Nation’s Sustainable Development Goals (SDG) cover 17 interlinked global objectives that aim at achieving a better and more sustainable future. The SDGs address a wide range of issues, including poverty, inequality, climate change, environmental degradation, peace, and justice, emphasizing that development must balance social, economic, and environmental sustainability. The SwissText 2024 Shared Task 1 requested researchers to design systems that assign scientific abstracts to the most appropriate SDG, or to a non-relevant category, if no SDG applies. The shared task organizers provided a dataset of over 400 labeled abstracts which has been highly unbalanced in regard to the class distribution (Figure 1).

The challenge within this shared task has been developing a classifier based on a highly unbalanced dataset of 18 classes (17 SDGs + the non-

relevant category) which can lead to significant model biases towards the majority classes and poor performance on the minority classes.

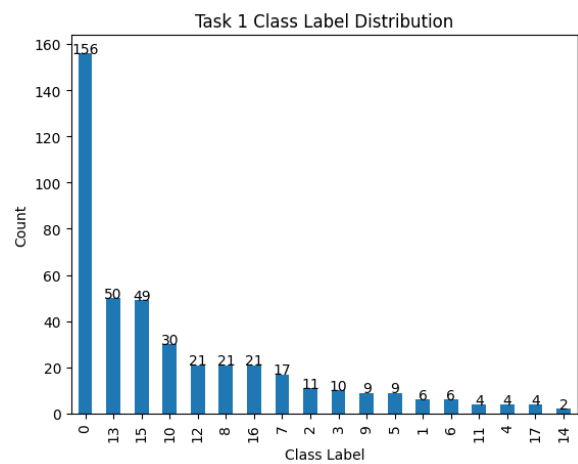


Figure 1: Label distribution within the training dataset for task 1

The rest of this paper is organized as follows: Section 2 outlines the method introduced in this work. Afterward, Section 3 presents and discusses evaluation results. The paper concludes with Section 4 which is followed by a short discussion of limitations.

2 Method

Figure 2 outlines the process used for training and optimizing the SDG classifier. At first, we draw upon GPT-3.5 (chatgpt.com) and Llama 3 (llama.meta.com/llama3/) to generate synthetic training data for all minority classes with the aim to better balance the dataset (Section 2.1).

We then train a transformer-based sequence classifier on both the training and synthetic dataset, and use it to classify the publicly available test dataset (Section 2.2). Finally, we apply explainable AI techniques to identify terminology within the incorrectly classified documents that has contributed significantly to misclassifications. Using an LLM

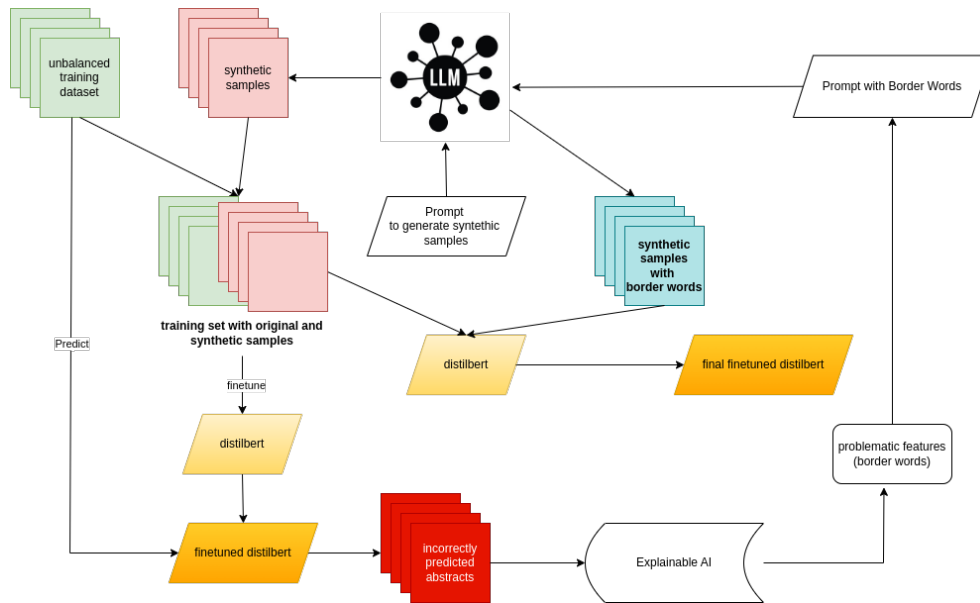


Figure 2: Process used for training the SDG classifier

allows to generate additional synthetic examples that contain this problematic terminology together with the correct class label. This additional synthetic training data aim at enabling the classifier to better learn the distinction between the affected classes, since it provides samples that have been inspired by prior mistakes and are aligned along the class boundaries.

2.1 Creating synthetic training data with GPT-3.5 and Llama 3

The first step utilizes GPT-3.5 and Llama 3 to generate synthetic training data for the minority classes, thus mitigating class imbalances and improving the overall performance of the text classification model. The following prompt was used to create the synthetic data:

```
You are a helpful assistant designed to generate synthetic data.

Create a JSONL file with 10 rows of data
.
The data comes from the United Nations' Sustainable Development Goals.

This is an example row from my current data

{"ID": "oai:www.zora.uzh.ch:126666",
 "TITLE": "Identifying phrasemes...",
 "ABSTRACT": "In corpus linguistics...",
 ... , "SDG": 0}
```

These are the SDGs that are available for the data:

- 0: "Non-Relevant",
- 1: "No Poverty",
- 2: "Zero Hunger",
- 3: "Good Health and Well-being",
- 4: "Quality Education",
- 5: "Gender Equality",
- 6: "Clean Water and Sanitation",
- 7: "Affordable and Clean Energy",
- 8: "Decent Work and Economic Growth",
- 9: "Industry, Innovation, and Infrastructure",
- 10: "Reduced Inequality",
- 11: "Sustainable Cities and Communities",
- 12: "Responsible Consumption and Production",
- 13: "Climate Action",
- 14: "Life Below Water",
- 15: "Life on Land",
- 16: "Peace, Justice, and Strong Institutions",
- 17: "Partnerships for the Goals"

Make sure that the text makes sense (i.e., the title and abstract are coherent) and that the SDG is one of the 18 options listed above. Also only respond with the resulting JSONL file.

Figure 3 summarizes the label distribution with the added synthetic samples, therefore, outlining the impact of the additional data on class imbalances.

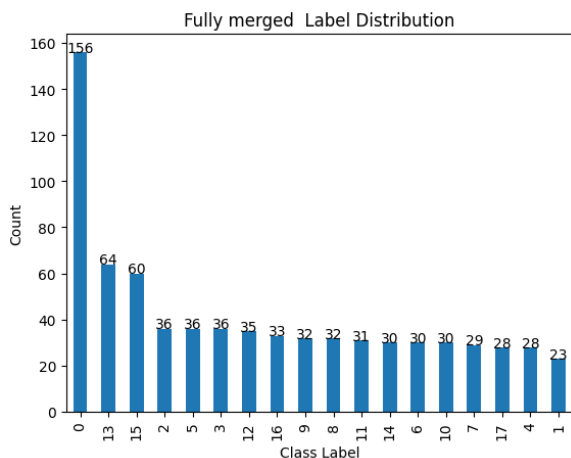


Figure 3: Label distribution after extending the gold standard with synthetic data generated by ChatGPT 3.5 and Llama 3

2.2 Transformer classifier

The proposed classification approach draws upon the Hugging Face library’s *AutoModelForSequenceClassification* class in conjunction with the *distilbert-base-multilingual-cased* model (Wolf et al., 2019). This approach leverages the pre-trained DistilBERT transformer model (Sanh et al., 2020), i.e., a distilled version of BERT, which is well-suited for rapid prototyping, since it provides quick training times in conjunction with a good performance for various natural language processing tasks.

The model was fine-tuned on the training and synthetic data using the AdamW optimizer and a learning rate scheduler. Cross-entropy loss function was utilized, as it is standard for multi-label classification tasks.

In addition, we draw upon Optuna (Akiba et al., 2019), a hyperparameter optimization framework, to identify the best hyperparameters for our model. The search space was defined using the `optuna_hp_space` function, specifying ranges for key hyperparameters such as learning rate, batch size, and the number of epochs. Optuna’s efficient search algorithms, such as Tree-structured Parzen Estimator (TPE), were utilized to explore this space and determine the optimal set of hyperparameters (Table 1).

Table 1: Hyperparameter configuration

Hyperparameter	Value
learning_rate	4.53e-05
per_device_train_batch_size	16
per_device_eval_batch_size	3
num_train_epochs	30
weight_decay	0.01

2.3 Scouting out the border

We developed an approach for identifying incorrectly classified abstracts, and extracting terminology that significantly contributed to the misclassification (i.e., problematic terms or border words), to create additional synthetic training data which are tailored towards addressing the classifier’s weaknesses. The hypothesis is that incorporating these synthetic samples into the training set will further enhance classification accuracy, by providing samples that are well-suited towards learning class boundaries.

2.3.1 Boundary scouting process

The boundary scouting process involves the following steps:

1. *Identify incorrectly classified abstracts*: Applying the developed DistilBERT classifier (Section 2.2) to the test data yields a set of incorrectly classified abstracts.
2. *Extraction of problematic terms*: The *SequenceClassificationExplainer* tool which is part of the Transformer Interpret package¹ is used for analyzing the incorrectly classified abstract. The package draws upon research by Janizek et al. (2020) and Sundararajan et al. (2017) which leverages attribution methods to assign importance scores to individual tokens in the input sequence. These scores indicate the contribution of each token to the model’s prediction, and help in understanding the model’s decision-making process by highlighting text that contributed most to the predicted class (i.e., the terminology responsible for misclassifications).
3. *Synthetic Sample Generation*: We use the identified problematic terminology in conjunction with Llama3-8B-8192 to generate synthetic abstracts that demonstrate the use of the

¹<https://pypi.org/project/transformers-interpret/>

problematic terminology (i.e., border words) in the correct class. For example, if an abstract has been misclassified as “Climate Action” (SDG 13) rather than “Affordable and Clean Energy” (SDG 7) due to the use of the phrase “solar energy”, we would ask the model to generate synthetic examples that use the phrase “solar energy” in the context of SDG 7. Automating this process yields additional training data that specifically address the classifier’s current weaknesses.

4. *Model Retraining*: Retrain the classifier with the original and synthetic samples.
5. *Evaluation*: Evaluate the performance of the retrained classifier to assess improvements.

2.3.2 Example

The following example demonstrates the use of the proposed approach based on an abstract that has been misclassified by the initial classifier model (Section 3):

- *Gold standard label*: 4 (“Quality Education”)
- *Predicted label*: 8 (“Decent Work and Economic Growth”)

Figure 4 in the paper’s appendix shows the classified example text together with the interpretation obtained from the Transformer Interpret package with tokens that contributed significantly to the incorrect classification (labour, ter, market, differenti, academic, the) marked in green.

The model likely considered the text’s focus on the labor market and economic implications of education as more relevant to SDG 8. Words related to economic growth and employment outcomes provided strong signals that outweighed the educational content, despite the text’s clear relevance to the quality of education.

We, therefore, use the following prompt to instruct the LLM to generate a synthetic abstract that belongs to the correct class:

Invent a title and an abstract of a research paper about Sustainable Development Goals (SDG) that has

- the subject = Quality Education
- the abstract should contain and focus the content around following words extensively: ['labour', 'ter', 'market', 'differenti', 'academic', 'the']

Do not include the words Sustainable Development Goals (SDG) in the abstract or the title.

The model then returned the following output (shortened):

Title: Bridging Academic Pathways and Labour Market Needs: Analyzing the Impact of Quality Education on Economic Differentiation

Abstract: This paper examines the intricate relationship between quality education and its impact on labour market differentiation. By analyzing academic programs and their alignment with the evolving needs of the labour market, this study highlights the critical role of education in fostering economic growth and social stability...

The generated synthetic abstracts have been added to the training set and used to retrain the transformer classifier. Section 3 outlines the performance gains obtained through this process.

3 Evaluation

We submitted two evaluation runs. One in which the DistilBERT classifier has been trained on the test and synthetic dataset (Syn; submission name: *NLPChur_TASK1__merged_synthetic_data_task1_report_goldlabel.txt*), and a second one which used in addition the synthetic abstracts generated based on the problematic words (Syn+; submission name: *NLPChur_TASK1__merged_synthetic_data_bad_words_task1_report_goldlabel.txt*).

Table 2 presents the overall classification performance of both approaches. As outlined in the table, the classifier trained on the improved synthetic dataset that has been extended based on the method introduced in Section 2.3 (Syn+) outperforms the classifier trained on the initial synthetic dataset (Syn) in every single evaluation metric.

Table 3 presents the per class classification performance for the classifier trained on the Syn+ dataset. The presented results indicate that although the overall performance improved significantly with the boundary scouting process, there are still classes where the classifier clearly failed to produce viable results. Investigating and mitigating

Table 2: Overall classification performance on Task 1 (correct prediction of the primary SDG) for the classifier trained on (i) the training and synthetic dataset (Syn), and (ii) the training, synthetic dataset and the synthetic data created based on the border words (Syn+).

Metric	Syn	Syn+
Accuracy	0.46	0.52
Macro Precision	0.49	0.53
Macro Recall	0.51	0.60
Macro F1 Score	0.44	0.51
Weighted Precision	0.59	0.65
Weighted Recall	0.46	0.52
Weighted F1 Score	0.49	0.55

these shortcomings will be an interesting direction for future work.

Table 3: Per label classification performance on Task 1 (correct prediction of the primary SDG) of the classifier trained on the Syn+ dataset.

SDG	f1	precision	recall
0	0.59	0.48	0.77
1	0.67	0.67	0.67
2	0.89	1.00	0.80
3	0.18	0.33	0.12
4	0.25	0.17	0.50
5	0.73	1.00	0.57
6	0.86	0.75	1.00
7	0.67	1.00	0.50
8	0.06	0.20	0.04
9	0.50	0.60	0.43
10	0.40	0.75	0.27
11	0.67	0.50	1.00
12	0.60	0.50	0.75
13	0.36	1.00	0.22
14	0.89	0.80	1.00
15	0.91	1.00	0.83
16	0.00	0.00	0.00
17	0.00	0.00	0.00

4 Outlook and Conclusions

This paper introduced an approach for creating and optimizing a transformer-based Sustainable Development Goals (SDG) classifier that was used in SwissText Shared Task 1 that focuses on identifying the majority SDG class for scientific abstracts. We leverage LLMs and explainable AI to generate synthetic training data that aims at (i) mitigating

class imbalances, and (ii) aiding the classifier in learning class boundaries. The system obtained the top accuracy for the Shared Task 1, demonstrating the method’s potential.

Future work will focus on further improving the system’s performance by adding a binary classifier to distinguish between abstracts that contain references to SDGs and those that do not. Additionally, efforts will be directed towards enhancing the methodology for generating synthetic training data. This includes improving the handling of subtokens to ensure more accurate and representative synthetic samples. Finally, we plan to investigate cases where the respective classes did not benefit from the improved synthetic dataset and research strategies to address this issue.

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. *Optuna: A next-generation hyperparameter optimization framework*. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD ’19*, page 2623–2631, New York, NY, USA. Association for Computing Machinery.
- Joseph D. Janizek, Pascal Sturmfels, and Su-In Lee. 2020. *Explaining Explanations: Axiomatic Feature Interactions for Deep Networks*. ArXiv:2002.04138 [cs, stat].
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. In *The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*. ArXiv: 1910.01108.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. *Axiomatic Attribution for Deep Networks*. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR. ISSN: 2640-3498.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. *Huggingface’s transformers: State-of-the-art natural language processing*. *CoRR*, abs/1910.03771.

A Transformer Interpret

Figure 4 illustrates the interpretations obtained from the Transformer Interpret package, which highlights tokens that contributed significantly to the classification label marked in green, and tokens that provided contrary information in red.

1

Word Importance

[CLS] This paper analyses whether tertiary education of different types, i.e., academic or vocational tertiary education, leads to more or less favourable labour market outcomes. We study the problem for Switzerland, where more than two thirds of the workforce gain vocational secondary degrees and a substantial number go on to a vocational tertiary degree but only a small share gain an academic tertiary degree. As outcome variables, we examine the risk of being unemployed, monthly earnings, and variation in earnings (reflecting financial risk). We study these outcomes at career entry and later stages. Our empirical results reveal that the type of tertiary education has various effects on these outcomes. At career entry, we observe equal unemployment risk but higher average wages and lower financial risk for vocational graduates. At later career stages, we find that these higher average wages disappear and risk of unemployment becomes lower for vocational graduates. Thus, by differentiating the tertiary system into vocational and academic institutions graduates face a variety of valuable options allowing them to self-select into an educational type that best matches their individual preferences. [SEP]

Figure 4: Example explanation provided the Transformer Interpret package

Swiss Text - Shared Task (SDG Classification) - Task 1

Manuel Bolz
University of Zurich
manuel.bolz@uzh.ch

Andreas Loizidis
University of Zurich
andreas.loizidis@uzh.ch

Kevin Bründler
University of Zurich
kevin.bruendler@uzh.ch

Abstract

This workshop paper presents the methodology and results of our participation in the Swiss-Text Shared Task 2024, focusing on classifying scientific abstracts into Sustainable Development Goals (SDGs). To address data sparsity and class imbalance, we employed synthetic data generation using large language models, including GPT-4, Mixtral-8x22B, and Llama-3-70b. We utilized a domain-adjusted version of AttrPrompt (Yu et al., 2024) to generate 16,600 synthetic abstracts, leveraging models such as GPT-4, Mixtral-8x22B, and Llama-3-70b to fine-tune pre-trained SciBERT (Beltagy et al., 2019) and Muppet-RoBERTa (Aghajanyan et al., 2021) models. Our findings indicate that synthetic data significantly enhances model performance, though the optimal data generation model varies with the classifier. Notably, SciBERT (Beltagy et al., 2019) consistently outperformed Muppet-RoBERTa (Aghajanyan et al., 2021) across various metrics. The most human-like synthetic texts, generated by GPT-4, yielded the best performance. Our approach achieved third place in the shared task, demonstrating the potential of synthetic data in improving classification accuracy for complex, multiclass settings.

1 Introduction

To address critical global issues such as climate change, poverty, and inequality, all United Nations (UN) Member States have adopted the 2030 Agenda for Sustainable Development, encompassing 17 Sustainable Development Goals (SDGs) with diverse humanitarian, environmental, and developmental objectives. Each SDG includes several sub-targets representing the different facets of the 17 main SDGs. To facilitate the classification of research abstracts into SDGs, the University of Zurich’s Sustainability Team has curated the ZORA dataset consisting of labeled abstracts, where each abstract is given one of the 17 SDG labels or 0

(null class), if the abstract does not relate to any of the SDGs. Such classification aids in understanding research trends, identifying knowledge gaps, and ultimately informing policy decisions aimed at addressing these pressing global issues.

Previous attempts at SDG document classification utilized the labeled data from the OSDG Community Dataset (OSDG et al., 2023). For example, Sadick (2023) has fine-tuned a BERT-based text classification model trained on OSDG data, available on Huggingface. However, the model currently only supports the first 16 goals and does not contain a null class. Extending this, Roady (2023) explored various data configurations and language models to classify SDG labels in scientific abstracts with variable success, primarily caused by data sparsity, class imbalance, and vague class definitions, while also omitting a null class.

This paper investigates whether synthetic data generated by large language models (LLMs) can enhance model performance in multiclass classification tasks characterized by sparse and imbalanced data with poorly separated classes. While transformer-based models have shown promising results in text classification, they frequently struggle with generalization, particularly when confronted with limited data for certain classes and label noise.

Our approach builds on previous findings that synthetic data can improve classification accuracy on multiclass settings, particularly when certain classes are rare (Kochanek et al., 2023; Møller et al., 2024). We aim to employ LLM-generated synthetic data to expand the training dataset, thus improving the model’s capacity to learn from varied and representative examples across all SDG classes. By systematically evaluating the effectiveness of this approach when human-labeled data is not only sparse and imbalanced, but also suffers from label noise, we seek to contribute insights into improving the robustness and generalization capabilities

of SDG document classification models.

We apply a domain-adjusted version of AttrPrompt (Yu et al., 2024) to increase representation of underrepresented SDG classes. AttrPrompt enriches a prompt with a range of domain-specific attributes to generate synthetic data points and has demonstrated superior performance to simple class-conditional prompts. We test three models— GPT-4, Mixtral-8x22B, and Llama-3-70b—to generate 16,600 synthetic abstracts each to fine-tune pre-trained SciBERT (Beltagy et al., 2019) and Muppet-Roberta (Aghajanyan et al., 2021) models. We evaluate the performance of each model against a baseline model trained on the OSDG and ZORA dataset.

We find that the SciBERT model trained on synthetic data generated by GPT-4 performs best reaching an accuracy of 0.47. In both models, including synthetic data moderately increases the accuracy and generally, SciBERT outperforms Muppet-Roberta. However, accuracy stays overall low therefore leaving room for alternative approaches.

2 Methodology

2.1 Synthetic Data

To generate the synthetic data, we applied a three-step prompt to generate the attributes configuration for each SDG:

1. *Which 20 web-of-science research domains will most likely be related to the UN SDG goal number {sdg_id}: {description}?*
2. *Fill in the following structure for studies on the UN SDG goal number {sdg_id} with 10 diverse sub-topics per research domain: {json_structure}*
3. *Analyze the following research domains likely to contain studies on the UN SDG goal number {sdg_id} in terms of completeness. If there is a web-of-science research domain missing that could contain such studies, please generate these domains including 10 diverse sub-topics. Return the generated content in a json structure as shown in the following input: {json_structure}*

This process ensures a comprehensive set of potential research areas, each with multiple sub-topics, resulting in approximately 23 research areas and 230 sub-topics per SDG.

Further, we specify attributes such as *length*, *style*, and *abstract start*, which are described in the appendix. We utilize GPT-4, Llama-3-70b, and Mixtral-8x22B to generate synthetic abstracts by randomly combining these attributes from 34,500 potential combinations using the following prompt:

Write an abstract of a scholarly article from the Web of Science database concerning {main_topic}. Ensure the abstract:

1. *Aligns subtly with the themes of the UN SDG goal {sdg_goal}, though without explicit mention of the goal itself;*
2. *Focuses on '{subtopic}';*
3. *Starts by {abstract_start}*
4. *Is between {length} and {int(length) + 60} words in length*
5. *Reflects a study that {style}*

We excluded SDG 16 from synthetic data generation due to its over-representation in the OSDG and ZORA datasets. Post-generation, all synthetic abstracts were cleansed of any LLM-specific artifacts such as "*Here is a potential abstract:*" or "*(narrative hook)*", commonly found in outputs from Llama-3-70b and Mixtral-8x22B.

2.2 Null Class

To ensure robustness and validate the specificity of classification, we generated a null class consisting of abstracts unrelated to any SDG topics. This process involved several steps. Initially, we utilized GPT-4 to generate topics that are as unrelated as possible to any SDG by querying it with SDG labels and definitions. We then used these unrelated topics to scrape paper abstracts from Semantic Scholar.

Next, we conducted topic modeling on the collected abstracts. The text data was preprocessed, and we applied TF-IDF vectorization to extract relevant features. Using Non-negative Matrix Factorization (NMF), we identified prominent topics for each SDG and determined the most significant words associated with these topics.

To create the null class, we identified and excluded any abstracts containing specific keywords revealed by the topic modeling.

2.3 Data and Splits

The data used for training comes from three primary sources: the OSDG dataset, the synthetic

dataset detailed in the previous subsection, and the given shared task training set. The synthetic dataset included 1,000 samples for every class but 16 and 17. Class 16 was excluded as mentioned before, and for class 17 we generated 1,600 samples to compensate for it not appearing in the OSDG dataset. Synthetic data for the classes for each model amounts to 16,600 total samples¹. We generated 2185 samples for the null class.

To internally evaluate the models and choose the best hyperparameters, we used a 80/20 stratified train/test split, ensuring that at least two abstracts per class from the given dataset were included in the test set, maintaining class balance and representation. For the final submission as well as the experiments shown here, we used a 95/5 split for training, and the released test set for evaluation.

3 Experiments

3.1 Models

We considered two transformer models: SciBERT and Muppet (Massive Multi-task Representations with Pre-Finetuning) RoBERTa. SciBERT (Beltagy et al., 2019) is a variant of BERT pre-trained on a large corpus of scientific text, making it particularly suitable for academic and research-based tasks. Muppet (Aghajanyan et al., 2021) is a pre-finetuned variant of RoBERTa, trained using multi-task learning to enhance its performance across various natural language processing tasks.

3.2 Finetuning

In all settings, we apply a dropout of 0.1 and optimize cross-entropy loss using Adam (Kingma and Ba, 2017). We experiment with different hyperparameter settings and fine-tune the models for 2 to 5 epochs using batch sizes of 16 and 32, and a learning rate of 5e-6, 1e-5, 2e-5, or 5e-5 with a slanted triangular schedule (Howard and Ruder, 2018), which is equivalent to the linear warmup followed by linear decay (Devlin et al., 2019). For each dataset and BERT variant, we pick the best learning rate and number of epochs on the development set and report the corresponding test results. We found that the setting that works best across most datasets and models is 2 epochs, a batch size of 16, and a learning rate of 2e-5. While task-dependent, optimal hyperparameters for each task are often the same across BERT variants.

¹We were only able to generate 16,518 samples using Mixtral.

3.3 Results

Model	Accuracy	F1 Score
muppet-no-synth	0.38	0.27
muppet-llama	0.38	0.34
muppet-gpt-4	0.42	0.41
muppet-ensemble	0.40	0.41
muppet-mixtral	0.39	0.40
scibert-no-synth	0.38	0.33
scibert-llama	0.43	0.47
scibert-gpt-4	0.47	0.44
scibert-ensemble	0.45	0.43
scibert-mixtral	0.45	0.45

Table 1: Accuracy and Avg. F1 Score per Model

Table 1 showcases the performance of each model variant in terms of accuracy and F1 score. The Muppet variants showed relatively similar performances in terms of accuracy, with the Muppet-GPT-4 achieving the highest accuracy and F1 score at 0.42 and 0.41, respectively. Compared to the baseline model Muppet-no-synth, most variants show a slight improvement in accuracy.

On the other hand, SciBERT generally performed better, particularly the SciBERT-GPT-4 for the highest accuracy at 0.47 and the SciBERT-Llama for the highest F1 Score. The consistently higher performance across different datasets suggests a the SciBERT model is better suited for tasks concerning scientific text, likely benefiting from its training on a scientific corpus. Overall, SciBERT models generally outperformed Muppet models on the same data, indicating a possible advantage in handling task-specific nuances.

4 Conclusion

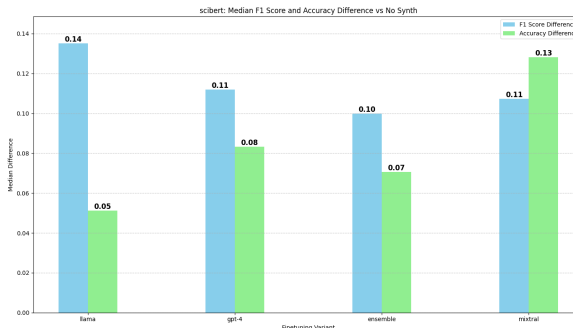


Figure 1: Median Improvement of Accuracy and F1 Score of the SciBERT model by Adding Synthetic Data

The results demonstrate that synthetic data significantly enhances performance in the multi-class

classification task. However, determining the most effective model for data generation remains inconclusive. Notably, the Muppet model exhibited the highest boost in F1 score when trained with synthetic data generated by the GPT-4 model. In contrast, SciBERT achieved better performance with synthetic data from llama. This variation suggests that the optimal choice of synthetic data generation model might be contingent upon specific model architectures and their inherent characteristics.

The most coherent and human-like synthetic texts, as assessed by the authors, were generated by GPT-4. In line with preliminary expectations, the GPT-4 generated data did yield the best classification performance for both Muppet and SciBERT in terms of accuracy. This indicates that the quality of synthetic data, in terms of human-likeness and coherence, could correlate with improved model performance. However, the effectiveness of synthetic data appears to be influenced by how well the generated data aligns with the specific characteristics and requirements of the target model.

Our findings contradict the general wisdom that "there is no data like more data." The ensembling of synthetic data from different models did not result in the largest F1 boost, suggesting that simply increasing the volume of synthetic data does not automatically enhance performance. It underscores the importance of the quality and compatibility of the synthetic data with the specific model being used.

Furthermore, our experiments underline the potential of leveraging large language models to mitigate issues of data sparsity and class imbalance in multiclass classification. The generated synthetic data contributed to noticeable improvements across several evaluation metrics, indicating its viability as a supplementary resource in training robust classification models.

While our study highlights the benefits of synthetic data, it also opens avenues for further research. Future work could explore a broader range of language models for synthetic data generation and investigate the underlying factors contributing to the varying performance boosts across different models. Additionally, a deeper examination of the attributes and configurations used in synthetic data generation could offer insights into optimizing these processes for enhanced classification outcomes.

Overall, our findings advocate for the integration of synthetic data into training pipelines, espe-

cially in scenarios with limited labeled data. This approach not only augments model performance but also aligns with the growing trend of using advanced language models to address complex challenges in natural language processing tasks.

Acknowledgements

The authors would like to thank the organizers of the SwissText Shared Task 2024 and the UZH Sustainability Team for providing the data.

References

- Armen Aghajanyan, Ankit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. **Muppet: Massive multi-task representations with pre-finetuning**. *CoRR*, abs/2101.11038. ArXiv preprint arXiv:2101.11038.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. **SciBERT: A Pretrained Language Model for Scientific Text**. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3615–3620. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jeremy Howard and Sebastian Ruder. 2018. **Universal Language Model Fine-tuning for Text Classification**. ArXiv preprint arXiv:1801.06146.
- Diederik P. Kingma and Jimmy Ba. 2017. **Adam: A Method for Stochastic Optimization**. ArXiv preprint arXiv:1412.6980.
- Mateusz Kochanek, Przemysław Kazienko, Jan Kocon, Igor Cichecki, Oliwier Kaszyca, and Dominika Szydło. 2023. Can Innovative Prompt Engineering with ChatGPT Address Imbalances in Machine Learning Datasets? *Authorea Preprints*. Authorea preprint.
- Anders Giovanni Møller, Arianna Pera, Jacob Dalsgaard, and Luca Aiello. 2024. The Parrot Dilemma: Human-Labeled vs. LLM-augmented Data in Classification Tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 179–192.
- OSDG, UNDP IICPSD SDG AI Lab, and PPMI. 2023. OSDG community dataset (OSDG-CD). <https://zenodo.org/records/7540165>. Zenodo.

Jessica Saemi Roady. 2023. Automatic classification of academic papers according to the UN sustainable development goals – an interdisciplinary perspective. Master’s thesis, University of Zurich, Zurich, 12. Supervisor: Dr. Simon Clematide, Department of Computational Linguistics.

A. M. Sadick. 2023. SDG classification with BERT. <https://huggingface.co/sadickam/sdg-classification-bert>.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Attributes: Abstract Start

1. Posing a question to frame the abstract in an engaging manner.
2. Mentioning the methodology used in the study.
3. Highlighting the significance or novelty about the research.
4. Using a narrative hook to grab attention.
5. Outlining the purpose or objective of the study.

A.2 Attributes: Length

1. 40
2. 100
3. 160

A.3 Attributes: Style

1. Tests hypotheses by manipulating variables to establish cause-and-effect relationships, using controlled experiments.
2. Constructs and articulates abstract concepts to develop theoretical frameworks for real-world application.
3. Compiles and evaluates existing research to summarize findings and highlight research gaps and patterns.
4. Provides an in-depth analysis of a specific event or individual to understand underlying principles.

5. Observes and describes phenomena as they occur naturally, detailing the observed features without manipulation.
6. Investigates relationships between variables to assess the strength and direction of associations.
7. Observes the same subjects over time to document changes and trends.
8. Gathers data from a population at a single time point to provide a snapshot of various characteristics.
9. Collaboratively addresses real-world problems, combining research with practical action for iterative improvements.
10. Integrates qualitative and quantitative methods to leverage their strengths for comprehensive insights.

A.4 Example Attributes: Main Topic SDG 1

1. Development Studies
2. Economics
3. Social Sciences - Interdisciplinary
4. Sociology
5. Environmental Science
6. Public, Environmental & Occupational Health
7. Anthropology
8. Political Science
9. Geography
10. Urban Studies
11. Education & Educational Research
12. Business & Economics
13. Agricultural Economics & Policy
14. Psychology - Applied
15. Law
16. Social Work
17. Demography
18. Health Care Sciences & Services

19. International Relations
20. Energy & Fuels
21. Human Geography
22. Behavioral Economics
23. Public Health

A.5 Example Attributes: Sub-topics SDG 1, Development Studies

1. Impact of microfinance programs on rural poverty
2. Effectiveness of conditional cash transfers
3. Role of technology in poverty alleviation
4. Sustainable livelihood frameworks
5. Community-based development projects
6. Urban vs. rural poverty dynamics
7. International development aid effectiveness
8. Gender and poverty
9. Education's impact on poverty reduction
10. Poverty and climate change resilience

Chapter 5

Workshops

Workshop

AI Support Systems for Academic Research

Susie Rao, Noah Mamié, Yilei Tu and Prakhar Bhandari

ETH Zurich

Description

With the surging of AI technologies, academicians have been adapting the ways research works are produced, transmitted and evaluated. AI support systems are widely used to retrieve information, such as journal recommenders [1] [2], SemOpenAlex [3], (LLM and KG powered) academic chatbot [4], SciSpace [5], Consensus [6], Research GPT [7]. In this hands-on workshop, we will go through three presentations on topics to understand (1) academic networks through the lens of OpenAlex, (2) journal recommendation techniques, (3) LLM and KG powered academic chatbot. Then the participants will have the chance to interact with a Telegram academic chatbot we design in-house at ETH Zurich. The academic chatbot supports retrieving papers, authors and affiliations that are relevant for user queries (e.g., Recommend the top-5 papers related to the paper “Attention is All You Need” by Ashish Vaswani.).

Schedule

10:50 -11:10 Organizer and workshop introduction, background probe of participants

11:00 – 12:20 Part 1: Talks

11:10 – 11:30 Deep diving academic networks through the lens of OpenAlex (by Noah Mamié)
Abstract: The talk includes introducing OpenAlex as a resource for academic research in various projects. (1) We compare two academic databases (OpenAlex and Microsoft Academic Graph) by benchmarking graph-based algorithms in tasks like node classification. (2) We introduce a graph reasoning framework that is capable of reasoning around the complex topic of deciding on the most worthy individuals to receive the annual Nobel Prize by leveraging graph neural networks.

11:30 – 11:50 Recommendation System for Journals based on ELMo and Deep Learning (by Mahmoud Hemila)

Abstract: The work evaluates how adequate recommender systems are for the selection of journals that fit to scientific publications. Specifically, several word embedding (word2vec, tf-idf, ELMo) and classification (LR, CNN, RNN, MLP) methods were tested and evaluated against each other in terms of their recommendation accuracy.

11:50 – 12:10 Large language models and knowledge graph powered academic chatbot (by Susie Xi Rao)

Abstract: We introduce an academic chatbot designed to help identify relevant publications, authors, and affiliations in academia and applied innovation. Leveraging similarity thresholds and query transformations, the chatbot delivers answers that are informed by an academic knowledge base. We address the challenges of efficiency, reproducibility, and interpretability through a combination of rule-based solutions and large language models backed by knowledge-graph embeddings. We have also critically analyzed the output of our chatbot and discussed various future directions of improvement.

12:10 – 12:20 Preparation for the hands-on session

14:35 –15:35 Part 2: Hands-on

14:30 -15:20 Hands-on session on the academic bot

15:20 – 15:30 Feedback and exchange

Workshop

RAG: Unveiling the Power of Retrieval-Augmented Generation

Célien Donzé¹, Jonathan Guerne¹, Henrique Marques Reis¹ and Pedro Costa²

HE-Arc Neuchâtel¹, Lumind²

Description

The Retrieval-Augmented Generation (RAG) framework represents a groundbreaking approach that seamlessly integrates two fundamental techniques, retrieval and generation, within a large language model (LLM). The result is the generation of more context-aware and informative responses, making RAG a valuable tool for companies with extensive documentation but lacking an efficient means to access specific information. This workshop endeavors to provide a comprehensive understanding of the RAG technology, emphasizing its applications and advantages. Through a technical introduction accompanied by concrete examples, participants will gain insights into how RAG can be effectively employed to address challenges related to information retrieval and contextual generation. The workshop will also facilitate discussions on the practical implementation of RAG in real-world scenarios, exploring its potential in enhancing knowledge management systems. Furthermore, the workshop will delve into the realm of self-hosted Large Language Models (LLMs), shedding light on the importance of data privacy and security in the deployment of generative AI technologies. Participants will be equipped with knowledge about the intricacies of hosting LLM models independently. By the conclusion of the workshop, participants will possess the skills to proficiently interact with a LLM, querying it about the contents of its associated documents. The overarching goal is to empower individuals with the expertise needed to harness the full potential of RAG and self-hosted LLMs, fostering a practical and informed approach towards the integration of these technologies in diverse real-world applications.

Schedule

Technical background, duration: 30 min

 RAG overview

Hands on, duration: 1h

 In Google collab (only a laptop is needed)

Applied examples, duration: 30min

 Showcase real world implementation of RAG applications

Time for discussion, duration: 30min

References

Ollama Available: <https://github.com/ollama/ollama>

LangChain Available: <https://github.com/langchain-ai/langchain>

Workshop

Grounding Generative AI Models

Holger Keibel, Johannes Porzelt

Karakun AG

Description

One fundamental shortcoming of generative AI systems such as ChatGPT is that they tend to produce hallucinatory outputs which the human user might easily take to be facts, with potentially disastrous consequences. In principle, it is the responsibility of the user to fact-check any generative AI output before relying on it, but this is not common practice and often impossible for users when the respective topic lies outside their field of expertise.

This calls for (semi-)automated approaches to grounding generative AI models in relevant facts – or at least to quantify the confidence that a given generative AI output is correct. Existing approaches range from integrating domain-specific knowledge directly into the models (typically by means of retrieval-augmented generation, RAG) to applying post-generation filtering techniques (automated fact-checking) to making the generation of the AI output transparent to the user (explainable AI).

This workshop aims at sharing practical experience applying such approaches. The three presentations apply LLMs generatively on domains ranging from ticket routing in customer support to teaching assistants to the ethically challenging domain of euthanasia decisions. Hallucinations of the generative AI approaches are problematic in all three domains, and the presentations describe different approaches to mitigating them: RAG against a fixed set of outputs in conjunction with supervised learning approaches, RAG against multimodal Knowledge Graphs (KGs), and an explainable AI approach by querying the LLM itself in a series of yes/no questions. In a final panel discussion, lessons learned will be shared and discussed.

Schedule

10:30 Holger Keibel & Johannes Porzelt (Karakun AG, Basel) Introduction

10:45 Flurin Gishamer (Open Systems, Zurich) Practical Strategies for Enhancing Reliability of GenAI Systems in Customer Operations: An Overview

11:10 Xiaokun Zhang (Athabasca University, Edmonton/Canada) Knowledge Graphs Enhanced Retrieval-Augmented Generation for Eliciting Higher-Order Thinking

11:35 Gerold Schneider (UZH, Zurich) Evaluating Transformers on the Ethical Question of Euthanasia

12:00 All speakers Panel Discussion: Lessons Learned, Q & A

12:30 End of workshop, lunch break

Workshop

Next-Gen Cleantech Solutions: Mining Insights from Media and Patent Data with Natural Language Processing (NLP) and Large Language Models (LLMs)

Holger Keibel, Johannes Porzelt

Karakun AG

Description

At a time when tackling environmental challenges is of paramount importance, the cleantech industry plays a central role in promoting sustainable solutions. However, technological innovation in the cleantech sector requires a deep understanding not only of the technologies, but also of the market requirements. This information is usually embedded in a large amount of patent and media data, which is difficult to analyze manually to effectively capture the development trend. Using Natural Language Processing (NLP) and the latest advancements in Large Language Models (LLMs) is a natural choice to accelerate innovation. In this workshop, we will share our insights gained in solving this task. Several presentations on various relevant topics will be offered, followed by a hands-on session where participants can try out our LLMs-powered cleantech question-answering and recommendation system.

Schedule

10:30 – 10:35 Welcome

10:35 – 11:35 Part 1: Decoding Cleantech

- 10:35 – 10:50 Dr. Janna Lipenkova (Anacode): Disentangling the Global Cleantech Innovation Landscape
- 10:50 – 11:05 Dr. Susie Xi Rao (ETH Zurich): Large Language Models in Cleantech
- 11:05 – 11:20 Prof. Dr. Daniel Perruchoud (FHNW): RAG and Its Application for Cleantech Innovation
- 11:20 – 11:35 Dr. Guang Lu (HSLU): Leveraging Large Language Models for Next-Gen Cleantech Recommender Systems

11:35 – 11:40 Break

11:40 – 12:30 Part 2: Emerging Visions in Cleantech

- 11:40 – 12:05 Dr. Guang Lu (HSLU): Student Work Highlights – Accelerating Cleantech Advancement through NLP-powered Text Mining and Knowledge Extraction
- 12:05 – 12:30 Prof. Dr. Daniel Perruchoud (FHNW): Student Work Highlights – RAG in Cleantech

12:30 – 13:45 Lunch Break

13:45 - 15:30 Part 3: NLP in Action (Group Work in Parallel)

Group 1 by Dr. Guang Lu (HSLU): Cleantech Text Analysis and NLP

Group 2 by Prof. Dr. Daniel Perruchoud & George Rowlands (FHNW): Cleantech RAG

15:30 - 15:45 Summary and Feedback

Dr. Janna Lipenkova (Anacode): NLP for Innovation Intelligence

Workshop

Battle of NLP Ideas

Manuela Hürlimann¹, Katsiaryna Mlynchyk¹ and Philipp Kuntschik²

ZHAW Centre for Artificial Intelligence¹, adesso Schweiz AG²

Description

In order to advance NLP in Switzerland and foster collaboration, it's crucial to bring together practitioners from different backgrounds and offer a setting in which they can come up with innovative ideas. The NLP Expert Group of the data+innovation alliance along with adesso, SwissNLP and the Centre for Artificial Intelligence at ZHAW are proudly hosting the Battle of NLP Ideas, which has been sparking lively discussions and coming up with exciting new ideas in the past.

The main goal of this workshop is to get people with different skills and perspectives to brainstorm about innovative ideas that advance NLP. These ideas should ideally then lead to real projects, whether in research, industry, or open-source work.

The past three editions have shown that participants enjoy the Battle of NLP Ideas and come up with great ideas. Many ideas have been pursued further to ideation workshops and joint project proposals.

Schedule

Introduction, 20 min: All participants welcomed, and the structure and the flow of the Battle of NLP Ideas workshop outlined.

Round 1, 20 min: A brainstorming session in the groups of 2-4 people to generate NLP-related ideas. The best idea(s) are voted for to be presented in Round 2.

Round 2, 30 min: The groups are merged within a room and the best ideas from the groups are presented. A discussion and new voting for the 2 best ideas are followed.

Final Round, 30 min: Best ideas from every room are presented. All participants vote for the best ideas. The winner is awarded the victory.

Follow-Up planning, 10 min: Participants get the opportunity to communicate on following the ideas up. Next steps and follow-up meetings scheduled.

