Idiap Abstract Text Summarization System for German Text Summarization Task

Shantipriya Parida, and **Petr Motlicek**
Idiap Research Institute
Rue Marconi 19, 1920 Martigny, Switzerland
{shantipriya.parida,petr.motlicek}@idiap.ch

Abstract

Text summarization is considered as one of the challenging tasks in the NLP community. The availability of multilingual summarization dataset is rare and difficult to construct. In this work, we build an abstract text summarizer for the German text using the state-of-the-art "Transformer" model. We propose an iterative data augmentation approach of using synthetic data additionally along with real German summarization data. To generate synthetic data, the Common Crawl (German) data are exploited, covering different domains. The synthetic data were found effective under low resource condition and particularly helpful for our multilingual scenario where availability of summarizing data is still a challenging issue.

1 Introduction

Automatic text summarization is considered as one of the most challenging tasks because when we as humans summarize a piece of text, we read it entirely to develop our understanding and then write a summary highlighting its main points. Due to the lack of human knowledge and language capability in computers, it makes automatic text summarization one of the non-trivial tasks (Allahyari et al., 2017).

The extractive summarization technique produces summaries by choosing a subset of sentences in the original text. Abstract text summarization aims to shorten the long text into a human-readable form that contains the most important fact from the original text (Allahyari et al., 2017; Kryściński et al., 2018).

The deep learning based neural attention model when applying to abstract text summarization performed very well as compared to standard learning based approaches (Rush et al., 2015). Abstract text summarization using the attentional encoder-decoder recurrent neural network approach shown state of the art performance and set as a baseline model (Nallapati et al., 2016). Further improvements made to the baseline model by using pointer generator network and coverage mechanism using reinforcement learning based training procedure (See et al., 2017; Paulus et al., 2017). There is a disadvantage in natural language processing tasks such as text summarization for resource-poor, morphologically complex languages due to the shortage of quality linguistic data available in these languages (Kurniawan and Louvan, 2018).

The use of synthetic data along with the real data is one of the popular approaches followed in machine translation domain un-

der low resource condition to boost the translation quality (Bojar and Tamchyna, 2011; Hoang et al., 2018; Chinea-Rıos et al., 2017). Even iterative back-translation (e.g. training back-translation systems, multiple times) were found effective in machine translation (Hoang et al., 2018). We explored similar approaches in our experiment for the summarization task.

Section 1 describes related work on abstract text summarization. Section 2 explains the techniques which we followed in our work. Section 3 describes the dataset used in our experiment. Section 4 explains the experimental settings: models and their parameters. Section 5 provides evaluation results with analysis and discussion. The paper is concluded in Section 6.

2 Method Description

Across all experiments performed in this paper, we have used the Transformer model as implemented in OpenNMT-py[1] (Vaswani et al., 2018; See et al., 2017). The Transformer model is based on encoder/decoder architecture. In context to summarize, it takes as input a text and outputs its summary.

We use synthetic data as shown in Figure 1 to increase the size of the training data.

3 Dataset

We use German wiki data (spread across different domain) collected from the SwissText 2019[2] (real data) and Common Crawl[3] German data (synthetic data) in our experiment. The statistics of all the datasets are shown in Table 1.

[1] http://opennmt.net/OpenNMT-py/Summarization.html

[2] https://www.swisstext.org/

[3] http://commoncrawl.org/

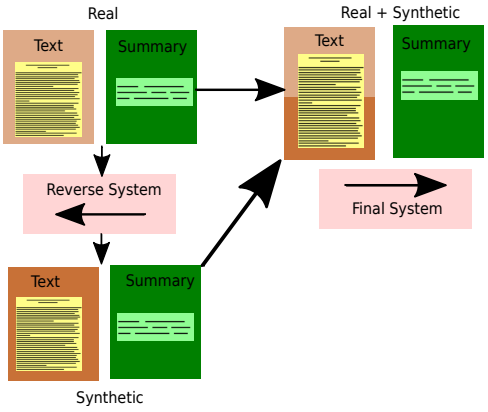

Figure 1: Generation of synthetic data using reverse system. To generate synthetic data, first, a system in the reverse direction (i.e. source as summary and target as text) is trained and then used to generate text for the given summary. Then both the real and synthetic data are input to the final system.

3.1 SwissText datasets used as real data

We divide the 100K SwissText dataset (downloaded from SwissText 2019 website) into three subsets - train, dev and test - in 90:5:5 ratio (i.e. 90K for training, 5K for development and 5K for the test data). The experiments exploiting these datasets are described in the Section 4.3 (denoted as S1 experimental setup).

3.2 Common Crawl dataset used as synthetic data

The data crawled from the Internet (Common Crawl) used to prepare synthetic data to boost the training. The steps followed to create the synthetic dataset as follows.

Step 1: **Build vocab:** We create vocabulary using SwissText based on the occurrence of the most frequent (top N) German words.

Step 2: **Sentence selection:** The sentences from the Common Crawl data are se-

lected with respect to the vocab based on the threshold we provide (e.g. a sentence has 10 words and the threshold is 10% (0.1). For this sentence to select, at least 1 word out of the 10 words should be in the vocab).

Step 3: **Filtering:** Select random sentences (e.g. 100K) from the selected Common Crawl data in the previous step.

Step 4: **Generate summary:** The 100K data obtained from the previous step are used as a summary and required to generate corresponding text. We use the reverse trained model where we provide the source as summary and target as text. This results in the text as well as summary as an additional dataset to be utilized along with real data (SwissText).

Eventually, the 190K dataset is created (denote as Train_RealSynth) as a combination of 90K SwissText train data (real) and 100K synthetic data. This dataset is used in the experimental setup S2 (described in details in Section 4.3).

DataSet	#Sent	#Summ
Train_Real(SwissText)	90K	90K
Train_RealSynth(Swiss+CC)	190K	190K
Train_RealSynthRegen(Swiss+CC)	190K	190K
Dev (SwisText)	5K	5K
Test (SwissText)	5K	5K
Test (SwissText Evaluation)	2K	-

Table 1: Statistics of the experimental data.

4 Experimental Setup

This section describes our experiments conducted for the text summarization task.

4.1 Prepossessing

The preprocess step involves preprocessing the dataset such that source and target are aligned and use the same dictionary and additional we have truncate source length at 400 tokens and target length at 100 tokens to expedite training (See et al., 2017).

Setting	Dataset	R1_F1	R2_F1	RL_F1	BLEU
S1	Dev	43.9	28.5	46.3	12.6
	Test	39.7	22.9	42.2	9.0
S2	Dev	45.4	29.8	47.4	14.0
	Test	55.7	41.8	57.6	20.8
S3	Dev	44.3	28.5	46.4	13.1
	Test	40.0	23.0	42.3	9.4

Table 2: Evaluation results of our models.

4.2 Model Parameters

The Transformer model is implemented in OpenNMT-py. To train the model, we use a single GPU. To fit the model to the GPU cluster, the batch size equal to 4096 is selected for training. The validation batch size is set to 8. We use a starting learning rate of 2, drop out of 0.2 and 8000 warm-up steps. Decoding uses a beam size of 10 and we did not set any minimum length of output summary.

4.3 Model Setup

We use 3 settings: i) real data (we set this as the baseline in our experiment), ii) real data and synthetic data, and iii) real and regenerated synthetic data for the summarization task described as follows:

1. *S1: Transformer model using Train_Real data:* In this setup, we use the "Train_Real" data for training the Transformer model.

2. *S2: Transformer Model using Train_RealSynth data:* In this setup, we use the "Train_RealSynth" data for training the Transformer model. As a balance between real and synthetic data matters, we studied this by opting 1:1 ratio (e.g. 1 (real) :1 (synthetic)) in our experiment (Sennrich et al., 2015).

Ref Summary : "Das Feuerschiff Relandersgrund war ein finnisches Feuerschiff, das von 1888 bis 1914 im Schrenmeer bei Rauma positioniert war. Heute dient es als Restaurantschiff in Helsinki."
S1 Summary: :"Die "Rauma". ist ein 1886—1888 Feuerschiff der norwegischen Reederei "Libauskij", Das Schiff wurde in den 1930er Jahren gebaut und in den 2000er Jahren als Museumsschiff als"
S2 Summary: :"Das Feuerschiff Relandersgrund war ein Feuerschiff des das von 1888 bis 1914 im Einsatz war.Heute dient es als Restaurantschiff in Kotka,"
S3 Summary: :"Die Relandersgrund ist ein 18861888 Schiff der russischen Marine, das fur eine und Wracks gebaut worden ist."

Table 3: Sample summaries on test set. The matching words of generating summaries with respect to references are shown in color blue.

3. *S3: Transformer Model using Train_RealSynthRegen data:* We propose an iterative approach to improve the quality of synthetic summaries. In this setup, after training a system with (real+synthetic) data, it is used to regenerate synthetic data for the final system. So, input data to the final system is a combination of real and regenerated synthetic data.

4.4 Training Procedure

The copying mechanism is applied during training which allows the summarizer to fall back and copy the source text when encounters $< unk >$ tokens by referencing to the softmax of the multiplication between attention scores of the output with the attention scores of the source (See et al., 2017). The systems are trained for 300K iterations.

5 Evaluation and Discussion

We evaluate the results for every 10,000 iterations on the dev and test set. The automatic evaluation results based on the dev and test set are shown in Table 2 with sample summaries in Table 3. To evaluate the proposed algorithms, we use ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score, which is the most popular metric for text summarization task and has various variants like ROUGE-N, ROUGE-L measures the overlap of n-grams between the system and reference summary (LIN, 2004). In addition, we also use the SacreBLEU[4] evaluation metric (Post, 2018).

Figure 2 presents the learning curves for the models (S1 and S2) on the development set.

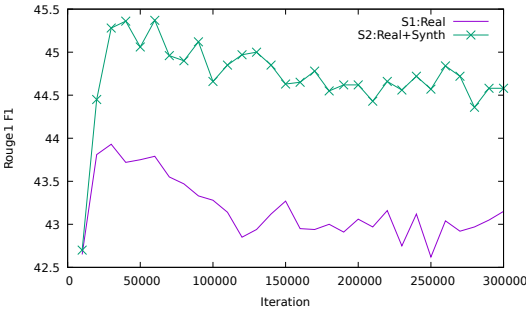

Figure 2: Learning curves in terms of Rouge1 F1 Score on dev set.

It can be seen that there is a variance (e.g. word selection, summary length) for S2 generated summary as compared with S1. During manual verification, we found that the summaries generated without a minimum length are better compared to summaries with minimum length. Although we do not explicitly specify a minimum length for the gener-

[4] https://github.com/mjpost/sacreBLEU

ated summaries for the models, the average length of words generated by model S2 (e.g. 41.42 words) is longer than the model S1 (e.g. 39.81 words). Some data (e.g. name, year) were not found correct during comparison of generating summary with respect to the reference. Although there is a variance in summaries generated by S3 nevertheless in terms of evaluation score, S3 outperforms S1 but perform worse than S2 (see Table 2).

6 Conclusion

In this paper, we highlighted the usage of synthetic data for the abstract text summarization task under low resource condition and observed an improvement in terms of automatic evaluation metrics. As the next step, we plan to investigate more on: i) synthetic summarization data, and ii) applying transfer learning on text summarization for the multilingual low resource data set with little or no ground truth summaries (Keneshloo et al., 2018).

Acknowledgments

The work is supported by a joint research project (under an InnoSuisse grant) oriented to further improve the automatic speech recognition and natural language understanding technologies for German. Title: **SM2: Extracting Semantic Meaning from Spoken Material**.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268* .

Ondrej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Sixth Workshop on Statistical Machine Translation*. page 330.

Mara Chinea-Rıos, Alvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. *WMT 2017* page 138.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. *ACL 2018* 23(32.5):18.

Yaser Keneshloo, Naren Ramakrishnan, and Chandan K Reddy. 2018. Deep transfer reinforcement learning for text summarization. *arXiv preprint arXiv:1810.06667* .

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 1808–1817.

Kemal Kurniawan and Samuel Louvan. 2018. Indosum: A new benchmark dataset for indonesian text summarization. In *2018 International Conference on Asian Language Processing (IALP)*. IEEE, pages 215–220.

C-Y LIN. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. of Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023* .

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* .

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, pages 186–191. http://aclweb.org/anthology/W18-6319.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 379–389.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* .

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709* .

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416* .