Abstractive Text Summarization based on
Language Model Conditioning and Locality Modeling

Dmitrii Aksenov, Julián Moreno-Schneider, Peter Bourgonje,
Robert Schwarzenberg, Leonhard Hennig, Georg Rehm

DFKI GmbH, Alt-Moabit 91c, 10559 Berlin, Germany
{firstname.lastname}@dfki.de

Abstract

We explore to what extent knowledge about the pre-trained language model that is used is beneficial for the task of abstractive summarization. To this end, we experiment with conditioning the encoder and decoder of a Transformer-based neural model on the BERT language model. In addition, we propose a new method of BERT-windowing, which allows chunk-wise processing of texts longer than the BERT window size. We also explore how locality modeling, i.e., the explicit restriction of calculations to the local context, can affect the summarization ability of the Transformer. This is done by introducing 2-dimensional convolutional self-attention into the first layers of the encoder. The results of our models are compared to a baseline and the state-of-the-art models on the CNN/Daily Mail dataset. We additionally train our model on the SwissText dataset to demonstrate usability on German. Both models outperform the baseline in ROUGE scores on two datasets and show its superiority in a manual qualitative analysis.

Keywords: Summarisation, Language Modeling, Information Extraction, Information Retrieval, BERT, Locality Modeling

1. Introduction

Text summarization is an NLP task with many real-world applications. The ever-increasing amount of unstructured information in text form calls for methods to automatically extract the relevant information from documents and present it in condensed form. Within the field of summarization, different paradigms are recognised in two dimensions: extractive vs. abstractive, and single-document vs. multi-document. In extractive summarization, those sentences or words are extracted from a text which carry the most important information, directly presenting the result of this as the summary. Abstractive summarization methods paraphrase the text, and by changing the text aim to generate more flexible and consistent summaries. Furthermore, single-document summarization works on single documents, while multi-document summarization deals with multiple documents at once and produces a single summary. In this paper, we concentrate on single-document abstractive summarization. Most recent abstractive models utilize the neural network-based sequence-to-sequence approach. During training, such models calculate the conditional probability of a summary given the input sequence by maximizing the loss function (typically cross-entropy). Most approaches are based on the encoder-decoder framework where the encoder encodes the input sequence into a vector representation and the decoder produces a new summary given the draft summary (which is the part of the summary generated during previous iterations). The last layer of a decoder, the generator, maps hidden states to token probabilities. We use a state-of-the-art Transformer for sequence-to-sequence tasks which is built primarily on the attention mechanism (Vaswani et al., 2017).

We attempt to improve performance of abstractive text summarization by improving the language encoding capabilities of the model. Recent results have shown that the main contribution of the Transformer is its multi-layer architecture, allowing Self-Attention to be replaced with some other technique without a significant drop in performance (Domhan, 2018; Wu et al., 2019). Following this strategy, we develop a model that introduces convolution into the vanilla Self-Attention, allowing to better encode the local dependencies between tokens. To overcome the data sparsity problem, we use a pre-trained language model for the encoding part of the encoder-decoder setup, which creates a contextualized representation of the input sequence. Specifically, we use BERT due to its bi-directional context conditioning, multilingualism and state-of-the-art scores on many other tasks (Devlin et al., 2019). Furthermore, we propose a new method which allows applying BERT on longer texts. The main contributions of this paper are: (1) Designing two new abstractive text summarization models based on the ideas of conditioning on the pre-trained language model and application of convolutional self-attention at the bottom layers of the encoder. (2) Proposing a method of encoding the input sequence in windows which alleviates BERT's input limitations[1] and allows the processing of longer input texts. (3) Evaluating the performance of our models on the English and German language by conducting an ablation study on CNN/Dail Mail and SwissText datasets and comparing it with other state-of-the-art methods.

2. Related Work

2.1. Pre-trained Language Models

Traditionally, non-contextualized embedding vectors were used for pre-training neural-based NLP models (Mikolov et al., 2013; Pennington et al., 2014). Recently, pre-trained language models exploiting contextualized embeddings, such as ELMo, GPT-2, BERT and XLNet raised the bar in many NLP tasks (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019b). Recent attempts to use these models for text summarization demon-

---

[1] BERT can process sequences with a maximum of 512 tokens.

strated their suitability by achieving new state-of-the-art results (Zhang et al., 2019; Liu, 2019; Liu and Lapata, 2019).

2.2. Neural Abstractive Text Summarization

The neural approach toward abstractive summarization was largely adopted by state-of-the-art models (Shi et al., 2018). A significant contribution was the pointer Generator Network (See et al., 2017). It uses a special layer on top of the decoder network to be able to both generate tokens from the dictionary and extract them from the input text. It uses the coverage vector mechanism to pay less attention to tokens already covered by previous iterations. An example of earlier work adapting Reinforcement Learning (RL) is described by Paulus et al. (2018). The pure RL model achieved high ROUGE-1 and ROUGE-L scores but produced unreadable summaries. Its combination with typical cross-entropy optimization achieved high scores eliminating the unreliability problem. Liu et al. (2018), to the best of our knowledge, were the first to use the Transformer model for summarization. It was only used in the decoder on top of the extraction model with various attention compression techniques to increase the size of the input sequence. Zhang et al. (2019) incorporate BERT into the Transformer-based model. They use a two-stage procedure exploiting the mask learning strategy. Others attempt to improve their abstractive summarization models by incorporating an extractive model. For example, Li et al. (2018) use the Key information guide network to guide the summary generation process. In Bottom-up summarization (Gehrmann et al., 2018) the extractive model is used to increase the precision of the Pointer Generator mechanism. Another strand of research adapts existing models to cope with long text. Cohan et al. (2018) present the Discourse-Aware Attention model which introduces hierarchy in the attention mechanism via calculating an additional attention vector over the sections of the input text. Subramanian et al. (2019) showed that the language model trained on the combination of the original text, extractive summaries generated by the model and the golden summary can achieve results comparable to standard encoder-decoder based summarization models.

3. Approach

Our text summarization model is based on the Transformer architecture. This architecture adopts the original model of Vaswani et al. (2017). On top of the decoder, we use a Pointer-Generator (Formula 1) to increase the extractive capabilities of the network (we later refer to this architecture as CopyTransformer).

$$p(w) = p_{gen}P_{copy}(w) + (1 - p_{gen})P_{softmax}(w), \quad (1)$$

where $P_{copy}(w)$ is the probability of copying a specific word w from the source document, $P_{softmax}(w)$ is the probability of generation a word calculated by the abstractive summarization model and p_{gen} is the probability of copying instead of generation.

3.1. Convolutional Self-Attention

The Transformer, like any other self-attention network, has a hierarchical multi-layer architecture. In many experi-
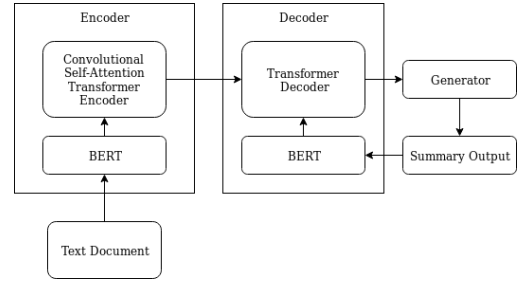

Figure 1: Model overview

ments it was shown that this architecture tends to learn lexical information in the first layers, sentence-level patterns in the middle and the semantics in the upper layers (Raganato and Tiedemann, 2018; Tenney et al., 2019). The disadvantage of this approach is that during the attention operation it considers all tokens as equally important, whereas syntactic information is mostly concentrated in certain local areas. This problem is usually specified as the problem of locality modeling. As syntactic information can help in identifying more important words or phrases, it could be beneficial to focus attention on these regions.

A successful approach to the locality modeling task are the so-called convolutions (local) self-attention networks (Yang et al., 2019a). Essentially, the problem is dealt with by the application of a 1-dimensional convolution to the self-attention operation at the network's lower layers. This strengthens dependencies among neighboring elements and makes the model distance-aware when it searches for low-level patterns in a sequence. In other words, it restricts the attention scope to the window of neighboring elements. The 1D convolution applied to attention is illustrated in Formulas 2, 3 and 4.

$$\widehat{\mathbf{K}}^h = \{\mathbf{k}^h_{i-\frac{M}{2}}, \ldots, \mathbf{k}^h_i, \ldots, \mathbf{k}^h_{i+\frac{M}{2}}\}, \quad (2)$$

$$\widehat{\mathbf{V}}^h = \{\mathbf{v}^h_{i-\frac{M}{2}}, \ldots, \mathbf{v}^h_i, \ldots, \mathbf{v}^h_{i+\frac{M}{2}}\}, \quad (3)$$

$$\mathbf{o}^h_i = \text{ATT}(\mathbf{q}^h_i, \widehat{\mathbf{K}}^h)\widehat{\mathbf{V}}^h, \quad (4)$$

where $\mathbf{q}^h_i$ is the query and $M + 1$ ($M \leq I$) is its attention region centered at the position i.

The convolution can be extended to the 2-dimensional area by taking interactions between features learned by the different attention heads of the Transformer into account. In the original Transformer each head independently models a distinct set of linguistic properties and dependencies among tokens (Raganato and Tiedemann, 2018). By applying 2-dimensional convolution, where the second dimension is the index of attention head, we explicitly allow each head to interact with learned features for their adjacent sub-spaces. The shortcoming of the original implementation is that the first and the last heads do not interact as they are assumed not to be adjacent. Thus, we assume that considering the heads' sub-spaces periodically, we can increase the model's effectiveness by applying circular convolution to the second dimension. In Section 5, we evaluate both the original version and our modification.

$$\widetilde{\mathbf{K}}^h = \bigcup [\widehat{\mathbf{K}}^{h-\frac{N}{2}}, \ldots, \widehat{\mathbf{K}}^h, \ldots, \widehat{\mathbf{K}}^{h+\frac{N}{2}}], \qquad (5)$$

$$\widetilde{\mathbf{V}}^h = \bigcup [\widehat{\mathbf{V}}^{h-\frac{N}{2}}, \ldots, \widehat{\mathbf{V}}^h, \ldots, \widehat{\mathbf{V}}^{h+\frac{N}{2}}], \qquad (6)$$

$$\mathbf{o}_i^h = \text{ATT}(\mathbf{q}_i^h, \widetilde{\mathbf{K}}^h)\widetilde{\mathbf{V}}^h, \qquad (7)$$

where $(M+1)$ $(N \leq H)$ is the window region over heads and $\bigcup$ stands for the union of keys $\widehat{\mathbf{K}}^h$ and values $\widehat{\mathbf{V}}^h$ from different subspaces.

The convolutional self-attention has been shown to be very effective in Machine Translation and several other NLP tasks. However, to our knowledge, it was never applied to the text summarization problem. For the experiments reported on in this paper, we created our implementation of the local attention and the convolutional self-attention network (Transformer). It supports both 1D and 2D modes having the size of the kernels as system parameters. As in Yang et al. (2019a) we incorporate convolutional self-attention in the Transformer encoder by positioning it in the place of the self-attention in the lower layers. In Section 5, we show that the low-level modeling capabilities of our encoder provides a strong boost to the model's prediction accuracy in the text summarization task.

3.2. BERT-Conditioned Encoder

The main task of the encoder is to remember all the semantic and syntactic information from the input text which should be used by the decoder to generate the output. Knowledge transfer from the language model should theoretically improve its ability to remember the important information due to the much larger corpus used in its pre-training phase compared to the corpus used in the text summarization training phase. We thus condition our encoder on the BERT language model.

For the encoder conditioning, we used the most straightforward strategy recommended for the BERT based model: placing the pre-trained language model in the encoder as an embeddings layer. This should make the embeddings of the system context-dependent. We decided not to fine-tune the encoder on BERT for the sake of memory and time economy. Instead, we follow the general recommendations by concatenating the hidden states of the last four layers of BERT into a 3072-dimensional embedding vector (Devlin et al., 2019). We use two variations of the BERT-based encoder. The first model uses only BERT to encode the input sequence and the second model feeds BERT's generated embeddings into the vanilla Transformer encoder.

3.3. BERT-Windowing

One of the key features of our approach is its ability to overcome the length limitations of BERT, allowing it to deal with longer documents. BERT's maximum supported sequence length is 512 tokens[2], which is smaller than the average size of texts used in most summarization datasets. Our method relies on the well-known method of windowing which to our knowledge was never used before neither

in the BERT-based models nor in abstractive text summarization research (Figure 2). We apply BERT to the windows of texts with strides and generate N matrices, every matrix embedding one window. Then we combine them by doing the reverse operation. The vectors at the overlapping positions are averaged (by summing and dividing by the number of overlapping vectors). As a result, we have the matrix of embeddings with the shape of the hidden size times the length of the text. The drawback of this approach is that we reduce the size of the context as each resulted vector is calculated based on maximum twice the window size number of tokens. Besides, the split of the text to equal size windows will aggravate the consistency of the input as some sentences will be split in an arbitrary manner between two adjacent windows. Despite this drawback, we assume that this procedure will nevertheless improve the accuracy of the encoder trained on the non-truncated texts. We set the window size to the maximum size of 512 tokens and the stride to 256. We consider this stride size optimal due to a trade-off between the average context size and computational requirements of the model (number of windows). By this trade we ensure every token to have a 768 tokens-context except for the 256 initial and final tokens, that only have 512 tokens-context.
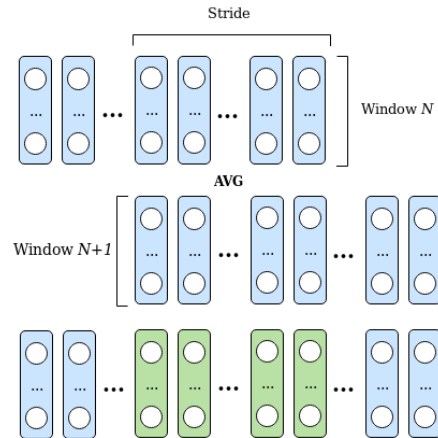

Figure 2: Integration of BERT-generated contextual representations from two windows

3.4. BERT-Conditioned Decoder

In the decoder, pre-training was applied in a similar way. The main difference is that instead of the final output of BERT we use only its word embedding matrix (without positions). The reason behind this is that in the decoder the generated probability distribution is conditioned on the incomplete text (previous summary draft output) while BERT implicitly assumes consistent and completed input (Zhang et al., 2019). As context-independent embeddings are not enough to represent the minimum set of features to make a meaningful prediction, the custom Transformer decoder is always stacked on top of BERT.

Our whole BERT-based model is similar to One-Stage BERT (Zhang et al., 2019) and BertSumAbs (Liu and Lapata, 2019) but differs in the usage of the four last hidden states of BERT to create contextualized representation, in presence of Pointer Generator and capabilities to process

---

[2] These are not tokens in the traditional sense, but so-called WordPiece tokens, see Devlin et al. (2019).
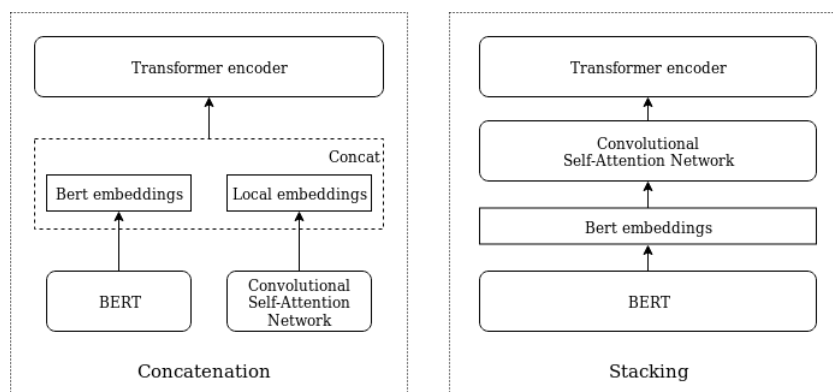
Figure 3: Two different approaches for the integration of the BERT-conditioning with Convolutional Self-Attention

Method	ROUGE-1	ROUGE-2	ROUGE-L
CopyTransformer	31.95	14.49	30.02
+ 1D conv.	32.62	14.99	30.74
+ 2D conv.	**32.72**	**15.12**	**30.85**
+ 2D Circular conv.	32.68	15.01	30.76

Table 1: Ablation study of model with Convolutional Self-Attention on the CNN/Daily Mail dataset (kernel sizes are 11 and 3)

long texts. In Figure 1 we show the schema of the basic model with the BERT-conditioned convolutional self-attention encoder and BERT-conditioned decoder.

3.5. Integration of BERT and Convolutional Self-Attention

We evaluated two different ways of integrating the BERT-conditioning with the convolutional self-attention of the model's encoder (Figure 3).

Stacking This approach comprises feeding the BERT-generated embeddings to the convolutional self-attention Transformer encoder. A potential problem with this approach is that convolutional self-attention is assumed to be beneficial when applied in the lower layers as its locality modeling feature should help in modeling of local dependencies (e. g., syntax). At the same time, BERT is a hierarchical model where the last layers target high-level patterns in the sequences (e. g., semantics). We assume that the application of the network detecting the low-level patterns on BERT's output can undermine its generalization abilities.

Concatenation Because of the considerations raised above, we also develop a second approach which we call Concatenation. We split the convolutional self-attention Transformer encoder into two networks where the first one uses only convolutional self-attention and the second original self-attention (identical to the Transformer encoder). Then we feed the original sequences into BERT and into the convolutional self-attention network in parallel. The resulting embedding vectors are concatenated and fed into the Transformer encoder. In this way, we model the locality at the lower layers of the encoder at the cost of a smaller depth of the network (assuming the same number of layers).

4. Datasets

We aim to develop a system that works in a language-independent way. It assumes that either the upstream components are available in the respective language, or they are themselves language-independent, such as the multi-lingual version of BERT. Since most summarization datasets are in English however, we use English for the evaluation and additionally include German to check if of our model can be applied to another language.

4.1. CNN/Daily Mail

Our experiments are mainly conducted on the CNN/Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016). It contains a collection of news articles paired with multi-sentence summaries published on the CNN and Daily Mail websites. This dataset is the de facto standard for training summarization models. We use the non-anonymized data as was used for training of the most recent state-of-the-art models (e. g., See et al. (2017)). The raw dataset consists of separate text files each representing a single article or a summary. We use the data in its preprocessed version as provided by Gehrmann et al. (2018). It has 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs. To align the data with the vocabulary of BERT we tokenized it using the BPE-based WordPiece tokenizer (Devlin et al., 2019). As all samples in BERT's training data are prepended with the special token "[CLS]", we follow
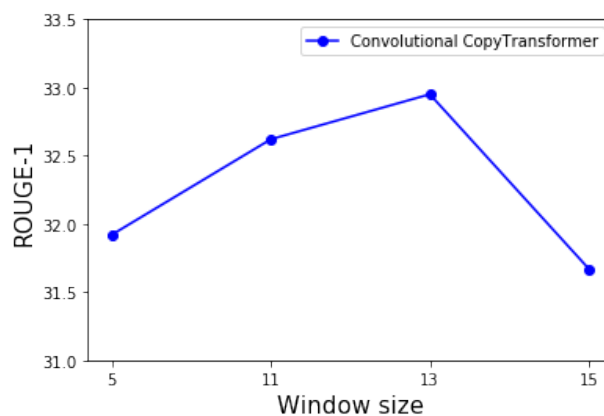

Figure 4: Effect of the window size on ROUGE-1

6683

Model	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	24.82	6.27	22.99
CopyTransformer	31.95	14.49	30.02
Bert Encoder + Transformer Decoder	31.3	13.37	29.46
Bert-transformer Encoder + Transformer Decoder	32.5	14.68	30.68
Bert-transformer Encoder + Bert-transformer Decoder	**33.23**	**14.99**	**31.26**
Transformer (full text)	23.18	5.15	21.48
Bert-transformer Encoder + Transformer Decoder (full text)	**31.51**	**14.1**	**29.77**

Table 2: Ablation study of the BERT-based model on truncated and original CNN/Daily Mail dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	36.40	20.69	34.14
CopyTransformer	39.44	25.11	37.16
Bert-transformer Encoder + Transformer Decoder	**44.01**	**29.60**	**41.65**
Bert-transformer Encoder + Bert-transformer Decoder	43.22	29.01	40.84
Transformer (full text)	34.76	18.65	32.61
Bert-transformer Encoder + Transformer Decoder (full text)	**45**	**30.49**	**42.64**

Table 3: Ablation study of the BERT-based model on the truncated and original SwissText dataset

this and add it to every source text in our dataset. In the resulting dataset, the average lengths of an article and a summary are 895 and 63 tokens, respectively. In most of our experiments, we use the clipped version of the training and validation datasets with each article truncated to 512 tokens. In the experiments on BERT windowing, we use the full-text version.

4.2. SwissText Dataset

To evaluate the efficiency of the model in a multi-lingual, multi-domain environment we conduct a series of experiments on the German SwissText dataset. This dataset was created for the 1st German Text Summarization Challenge at the 4th Swiss Text Analytics Conference – SwissText 2019 (ZHAW, 2019). It was designed to explore different ideas and solutions regarding abstractive summarization of German texts. To the best of our knowledge, it is the first long document summarization dataset in the German language that is publicly available. The data was extracted from the German Wikipedia and represents mostly biographical articles and definitions of various concepts.

The dataset was tokenized by the multilingual WordPiece tokenizer (Devlin et al., 2019) and preprocessed in the same way as the CNN/Daily Mail dataset. It was split into the training, validation and testing sets containing 90,000, 5,000 and 5,000 samples, respectively. The average length of a source sequence is 918 tokens, which makes this dataset suitable for our experiments on windowing.

5. Experiments

Our system is built on the OpenNMT library. For training, we use cross-entropy loss and the Adam optimizer with the Noam decay method (Kingma and Ba, 2014). Regularization is made via dropout and label smoothing. For evaluation, we calculate the F1-scores for ROUGE using the files2rouge library. The ROUGE evaluation is made on the sequences of WordPiece tokens.

5.1. Locality Modeling

To evaluate the effect of convolution on self-attention we introduce it in the first layer of the encoder. We use the same kernel sizes as in Yang et al. (2019a). In these experiments, to accelerate the training process, we use a small model with a hidden size of 256, four self-attention heads and three layers in the encoder and decoder. All models are trained for 90,000 training steps with the Coverage Penalty. As a baseline, we use our implementation of CopyTransformer. In contrast to See et al. (2017), we do not re-use the attention layer for the decoder but train a new Pointer-Generator layer from scratch.

The results are presented in Table 1. We see that both convolutions over tokens and over attention heads improve the ROUGE scores. Standard convolution outperformed circular convolution on ROUGE-1, ROUGE-2 and ROUGE-L by 0.06, 0.13 and 0.09 percent, respectively.

We also investigated the effect of the window size of the 1-dimensional convolution on ROUGE scores (Figure 4). In contrast to findings in Machine Translation, we found that size 13 returns the best result for the summarization task.

5.2. BERT Conditioning

To find the optimal architecture of the BERT-based abstractive summarizer we conducted an ablation study (Table 2). All hyperparameters were set equal to the ones in experiments in convolutional self-attention. On CNN/Daily Main dataset we test three different models: BERT encoder+Transformer Decoder, BERT-Transformer encoder+Transformer decoder and BERT-Transformer encoder+BERT-Transformer decoder. The version of BERT used in the experiments is BERT-Base. As the baseline, we use the Transformer without Pointer Generator. From the results, we observe that BERT improves the efficiency of the model when it is used in both encoder and decoder. Besides, BERT in the encoder is more effective when it is used to produce embeddings to be used by the

standard Transformer encoder than when it is used solely as an encoder. Even without a Pointer Generator, our model outperformed the CopyTransformer baseline by 1.28, 0.5 and 1.24 on ROUGE-1, ROUGE-2 and ROUGE-L.

To evaluate our BERT-windowing method we conducted the experiments on the full text. Our approach outperforms the baseline, which proves that the method can be successfully applied to texts longer than 512 tokens. The final performance of this model is still lower than that of the model trained on the truncated text, but as the same pattern can be observed for the baselines we assumed this relates to the specifics of the dataset that is prone to having important information in the first sentence of a text.

On SwissText data we use the multilingual version of BERT-Base. We evaluated two models with Bert-transformer encoder and Transformer and BERT-Transformer decoders (Table 3). The introduction of BERT into the transformer increased the ROUGE-1, ROUGE-2 and ROUGE-L scores by 7.21, 8.91 and 7.51 percent, respectively. At the same time, the usage of BERT in the decoder decreased the overall score. We assume that the reason behind this is that in multilingual BERT, due to its language-independence, the embedding matrix outputs less precise contextualized representations which undermines their benefits for the summarization task.

On the non-truncated texts, usage of the Bert-transformer encoder increased the ROUGE scores by 10.23, 11.84 and 10.03 percent. Furthermore, it gives us higher scores compared to the same model on truncated texts. This demonstrates the usability of BERT-windowing for this particular dataset. We assume that the difference in performance on the CNN/Daily Mail datasets reflects the difference in distribution of the useful information within the text. Particularly, that in the SwissText dataset, it is spread more uniformly than in the CNN/Daily Mail dataset. We conducted a small experiment comparing the average ROUGE score between a golden summary and the head and the tail of a document (taking the first or last n sentences, where n correlates to the length of the gold summary) on both datasets. The difference between taking the head and a tail on the SwissText dataset (ROUGE-L of 34.79 vs. 20.15, respectively) was much smaller than on CNN / Daily Mail (ROUGE-L of 16.95 vs. 12.27, respectively) which confirms our hypothesis.

5.3. Integration Strategies

To evaluate the integration strategies, we trained two models with the respective BERT-based baselines. Both models have in their encoder two Transformer layers and one Convolutional Transformer layer placed on top of BERT or in parallel, respectively (Table 4).

The method of stacking does not provide any significant improvement. With the introduction of convolutional self-attention only ROUGE-1 increased by 0.12 percent, while ROUGE-2 dropped by 0.3 and ROUGE-L remained the same. Considering that in many domains ROUGE-2 maximally correlates with human assessment (see Section 7), we dismiss this method. The concatenation strategy convolution is shown to be much more efficient, increasing ROUGE scores by 0.44,0.33 and 0.43 percent. This confirms our hypothesis that locality modeling is the most efficient when applied at the bottom on the non-contextualized word representations. Unfortunately, this model failed to outperform the stacking baseline. We conclude that the concatenating architecture undermines the performance of the Transformer model, and the convolutional self-attention is not beneficial when used together with pre-trained language models. Hence, we decided to train our two final models separately.

5.4. Model Comparison

For the final comparison of our model to other state-of-the-art methods we conducted experiments on the CNN/Daily Mail dataset. We set the hidden state to 512, the number of Transformer layers in the encoder and layers to six and the number of self-attention heads to eight. Hence, our baseline is smaller than the original CopyTransformer (Gehrmann et al., 2018), which may be the reason why it performs slightly worse (Table 5). BERT-conditioning was used in both the encoder and decoder. The sizes of convolution kernels are set to 13 and three. The networks were trained for 200,000 training steps on a single NVIDIA GeForce GTX 1080 Ti. The generation of the summary was made via the Beam search algorithm with the Beam size set to four. Finally, the generated summaries were detokenized back to the sequences of words separated by spaces.

For the BERT-based model, we set the minimum length of a generated summary to 55 as we found that without such restriction the model was prone to generate shorter sequences than in the test dataset. The model outperformed the baseline by 1.27 on ROUGE-1, 1.14 on ROUGE-2 and 1.3 on ROUGE-L. This is better than the scores of One-Stage BERT but still worse than the two-stage and Bert-SumAbs models.

For the convolutional CopyTransformer we use convolutional self-attention in the first three layers of the encoder. It increased ROUGE-1, ROUGE-2 and ROUGE-L by 0.25, 0.41 and 0.12.

Furthermore, we present the first publicly available benchmark for the SwissData dataset (Table 6).[3] All param-

---

[3]For comparability with our other model we include results

Method of Integration	Model	ROUGE-1	ROUGE-2	ROUGE-L
Stacking	BERT+CopyTransformer	35.28	**17.12**	33.31
	BERT+Convolutional CopyTransformer	**35.4**	16.82	**33.31**
Concatenation	BERT+CopyTransformer	34.82	16.46	32.79
	BERT+Convolutional CopyTransformer	**35.26**	**16.79**	**33.22**

Table 4: Different strategies for integrating language models with convolutional Self-Attention (CNN/Daily Mail dataset)

Method	ROUGE-1	ROUGE-2	ROUGE-L
BiLSTM + Pointer-Generator + Coverage (See et al., 2017)	39.53	17.28	36.38
ML + Intra-Attention (Paulus et al., 2018)	38.30	14.81	35.49
CopyTransformer (Gehrmann et al., 2018)	39.25	17.54	36.45
Bottom-Up Summarization (Gehrmann et al., 2018)	41.22	18.68	38.34
One-Stage BERT (Zhang et al., 2019)	39.50	17.87	36.65
Two-Stage BERT (Zhang et al., 2019)	41.38	19.34	38.37
ML + Intra-Attention + RL (Paulus et al., 2018)	39.87	15.82	36.90
Key information guide network (Li et al., 2018)	38.95	17.12	35.68
Sentence Rewriting (Chen and Bansal, 2018)	40.88	17.80	38.54
BertSumAbs (Liu and Lapata, 2019)	**41.72**	**19.39**	**38.76**
CopyTransformer (our implementation)	38.73	17.28	35.85
Convolutional CopyTransformer	38.98	17.69	35.97
BERT+CopyTransformer (enc., dec.)	**40**	**18.42**	**37.15**

Table 5: ROUGE scores for various models on the CNN/Daily Mail test set. The first section shows different state-of-the-art models, the second section presents our models and baseline.

Method	ROUGE-1	ROUGE-2	ROUGE-L
CopyTransformer (our implementation)	39.5	22.36	36.97
Convolutional CopyTransformer	40.54	23.62	38.06
BERT+CopyTransformer (enc.)	**42.61**	**25.25**	**39.85**

Table 6: ROUGE scores for our models on the SwissText test set

eters are equal to the CNN/Daily Mail baseline. BERT-conditioning was used only in the encoder. The networks were trained on the truncated texts in 90,000 training steps. From the results we see that the convolutional CopyTransformer showed much more efficiency than on CNN/Daily Mail dataset, outperforming the baseline by 1.04 percent on ROUGE-1, 1.26 on ROUGE-2 and 1.09 on ROUGE-L. The BERT-based model achieved the highest scores.

6. Qualitative Analysis

As ROUGE evaluation is not always a valid method for quality assessment we perceive the need for an additional, manual evaluation. The best solution would be to conduct a fine-grained study of the models' outputs by manually ranking them in terms of semantic coherence, grammaticality, etc. However, due to the time-consuming nature of such an evaluation, we reverted to a qualitative analysis comparing several summaries generated by different models. Figure 5 includes the reference summary and those generated by the different models. Comparing the first sentence we see that the vanilla Transformer model performed worse by copying only part of the original sentence omitting some characters in the word "meteorological". The model with convolution has copied the whole sentence but still made a spelling error. Finally, only the BERT-based model succeeded to generate the right token, "meteorological". Also, we see that while the BERT-based model's summary conveys the same meaning as the gold summary, the convolutional Transformer generates one and Transformer two sentences that are not present in the gold summary. Overall, on the given

for the bigger BERT+CopyTransformer model. At the same time, we found that the smaller model without the copy mechanism achieved higher scores with 45.12 ROUGE-1, 28.38 ROUGE-2 and 42.99 ROUGE-L. This needs to be explored in future work.

example all models provided a summary of extractive nature and only the BERT-based model shows some level of abstractiveness merging parts of the two sentences into the single one (in the second summary's sentence). This is far from the gold summary where every sentence in some way paraphrases the original text. Hence, given this particular example, our models demonstrate some explicit improvements. Still, abstractive summarization remains challenging. The paraphrasing capabilities of all state-of-the-art systems are low and the models are not guaranteed to produce summaries which follow the initial order of the sequence of events.

7. Discussion: Summarization Evaluation

ROUGE (Lin, 2004) is the most widely adopted metric used for evaluating automatic text summarization approaches. The evaluation is made though comparison of a set of system-generated candidate summaries with a gold standard summary. The availability of the corresponding software and its performance contributed to its popularity (Cohan and Goharian, 2016). Despite its adoption in many studies, the metric faced some key criticisms.

The main criticism of ROUGE is that it does not take into account the meaning expressed in the sequences. The metric was developed based on the assumption that a high quality generated candidate summary should share many words with a single human-made gold standard summary. This assumption may be very relevant to extractive, but not to abstractive summarization, where different terminology and paraphrasing can be used to express the same meaning (Cohan and Goharian, 2016). This results in the metric assigning low scores to any summary not matching the gold standard on the surface level. This also allows cheating the metric by generating ungrammatical and nonsensical sum-

Gold summary: researchers are developing a computer that can write weather forecasts . it takes meteorological data and writes a report designed to mimic a human . this process is known as ' natural language generation ' - lrb - nlg - rrb - . a prototype system will be tested on the bbc website later this year .
Transformer: researchers from london and edinburgh have developed a computer that can <mark>collateological</mark> information . these computer - generated weather updates are being tested by scientists at heriot - watt university and university college london . if the project is successful , a prototype system will be tested by generating local weather reports on the bbc ' s website . currently , the bbc website features 10 reports written by meteorologists .
Convolutional Transformer: researchers from london and edinburgh have developed a computer that can collate <mark>meterological</mark> information and then produce forecasts as if they were written by a human . it uses a process known as ' natural language generation ' - lrb - nlg - rrb - . these computer - generated weather updates are being tested by scientists at heriot - watt university and university college london . if the project is successful , a prototype system will be tested by generating local weather reports on the bbc ' s website .
BERT-Transformer: researchers from london and edinburgh have developed a computer that can collate meteorological information and produce forecasts as if they were written by a human . <mark>using met office data , it uses a process</mark> known as ' natural language generation ' - lrb - nlg - rrb - . if the project is successful , a prototype system will be tested by generating local weather reports on the bbc ' s website .

Figure 5: Comparison of the output of models on an example form CNN/Daily Mail testset. Surface realisation mistakes are highlighted in green and a typical abstractive feature, illustrating re-arranging of the sentence is highlighted in blue.

maries having very high ROUGE scores. Sjöbergh (2007) show how this can be achieved by choosing the most frequent bigrams from the input document.

ROUGE adoption relies on its correlation with human assessment. In the first research on the DUC and TDT-3 datasets containing news articles, ROUGE indeed showed a high correlation with the human judgments (Lin, 2004; Dorr et al., 2005). However, more recent research questions the suitability of ROUGE for various settings. Conroy and Dang (2008) show that on DUC data the linguistic and responsiveness scores of some systems do not correspond to the high ROUGE scores. Cohan and Goharian (2016) demonstrate that for summarization of scientific texts, ROUGE-1 and ROUGE-L have very low correlations with the gold summaries. ROUGE-N correlates better but is still far from the ideal case. This follows the result of Murray et al. (2005), showing that the unigram match between the candidate summary and gold summary is not an accurate metric to assess quality.

Another problem is that the credibility of ROUGE was demonstrated for the systems which operated in the low-scoring range. Peyrard (2019b) show that different summarization evaluation metrics correlate differently with human judgements for the higher-scoring range in which state-of-the-art systems now operate. Furthermore, improvements measured with one metric do not necessarily lead to improvements when using others.

This concern led to the development of new evaluation metrics. Peyrard (2019a) define metrics for important concepts with regard to summariazion: Redundancy, Relevance, and Informativeness in line with Shannon's entropy. From these definitions they formulate a metric of Importance which better correlates to human judgments. Clark et al. (2019) propose the metric of Sentence Mover's Similarity which operates on the semantic level and also better correlates with human evaluation. A summarization model trained via Reinforcement Learning with this metric as reward achieved higher scores in both human and ROUGE-based evaluation.

Despite these drawbacks, the broad adoption of ROUGE makes it the only way to compare the efficiency of our model with other state-of-the-art models. The evaluation of our system on the SwissData dataset confirms that its efficiency (in terms of ROUGE) is not restricted to CNN/Daily Mail data only.

8. Conclusion

We present a new abstractive text summarization model which incorporates convolutional self-attention in BERT. We compare the performance of our system to a baseline and to competing systems on the CNN/Daily Mail data set for English and report an improvement over state-of-the-art results using ROUGE scores. To establish suitability of our model to languages other than English and domains other than that of the CNN/Daily Mail data set, we apply our model to the German SwissText data set and present scores on this setup. A key contribution of our model is the ability to deal with texts longer than BERT's window size which is limited to 512 WordPiece tokens. We present a cascading approach and evaluate this on texts longer than this window size, and demonstrate its performance when dealing with longer input texts.

The source code of our system is publicly available.[4] A functional service based on the model is currently being integrated, as a summarization service, in the platforms Lynx (Moreno-Schneider et al., 2020), QURATOR (Rehm et al., 2020b) and European Language Grid (Rehm et al., 2020a).

Acknowledgements

---

[4]https://github.com/axenov/BERT-Summ-OpenNMT

9. Bibliographical References

Chen, Y.-C. and Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia, July. Association for Computational Linguistics.

Clark, E., Celikyilmaz, A., and Smith, N. A. (2019). Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy, July. Association for Computational Linguistics.

Cohan, A. and Goharian, N. (2016). Revisiting summarization evaluation for scientific articles. Available online (arXiv).

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In *NAACL-HLT*.

Conroy, J. M. and Dang, H. T. (2008). Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 145–152, Manchester, UK, August. Coling 2008 Organizing Committee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, June. Association for Computational Linguistics.

Domhan, T. (2018). How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808, Melbourne, Australia, July. Association for Computational Linguistics.

Dorr, B., Monz, C., President, S., Schwartz, R., and Zajic, D. (2005). A methodology for extrinsic evaluation of text summarization: Does ROUGE correlate? In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 1–8, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Gehrmann, S., Deng, Y., and Rush, A. (2018). Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, MA, USA. MIT Press.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12.

Li, C., Xu, W., Li, S., and Gao, S. (2018). Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, New Orleans, Louisiana, June. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Liu, Y. (2019). Fine-tune bert for extractive summarization. Available online (arXiv).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Moreno-Schneider, J., Rehm, G., Montiel-Ponsoda, E., Rodriguez-Doncel, V., Revenko, A., Karampatakis, S., Khvalchik, M., Sageder, C., Gracia, J., and Maganza, F. (2020). Orchestrating NLP Services for the Legal Domain. In Nicoletta Calzolari, et al., editors, *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France, 5. European Language Resources Association (ELRA). Accepted for publication. Submitted version available as preprint.

Murray, G., Renals, S., and Carletta, J. (2005). Extractive summarization of meeting recordings. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 593–596.

Nallapati, R., Zhou, B., dos Santos, C., Gülçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August. Association for Computational Linguistics.

Paulus, R., Xiong, C., and Socher, R. (2018). A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Peyrard, M. (2019a). A simple theoretical model of importance for summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy, July. Association for Computational Linguistics.

Peyrard, M. (2019b). Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy, July. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2018). Language models are unsupervised multitask learners. Available online.

Raganato, A. and Tiedemann, J. (2018). An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium, November. Association for Computational Linguistics.

Rehm, G., Berger, M., Elsholz, E., Hegele, S., Kintzel, F., Marheinecke, K., Piperidis, S., Deligiannis, M., Galanis, D., Gkirtzou, K., Labropoulou, P., Bontcheva, K., Jones, D., Roberts, I., Hajic, J., Hamrlová, J., Kačena, L., Choukri, K., Arranz, V., Vasiļjevs, A., Anvari, O., Lagzdiņš, A., Meļņika, J., Backfried, G., Dikici, E., Janosik, M., Prinz, K., Prinz, C., Stampler, S., Thomas-Aniola, D., Pérez, J. M. G., Silva, A. G., Berrío, C., Germann, U., Renals, S., and Klejch, O. (2020a). European Language Grid: An Overview. In Nicoletta Calzolari, et al., editors, *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France, 5. European Language Resources Association (ELRA). Accepted for publication.

Rehm, G., Bourgonje, P., Hegele, S., Kintzel, F., Schneider, J. M., Ostendorff, M., Zaczynska, K., Berger, A., Grill, S., Räuchle, S., Rauenbusch, J., Rutenburg, L., Schmidt, A., Wild, M., Hoffmann, H., Fink, J., Schulz, S., Seva, J., Quantz, J., Böttger, J., Matthey, J., Fricke, R., Thomsen, J., Paschke, A., Qundus, J. A., Hoppe, T., Karam, N., Weichhardt, F., Fillies, C., Neudecker, C., Gerber, M., Labusch, K., Rezanezhad, V., Schaefer, R., Zellhöfer, D., Siewert, D., Bunk, P., Pintscher, L., Aleynikova, E., and Heine, F. (2020b). QURATOR: Innovative Technologies for Content and Data Curation. In Adrian Paschke, et al., editors, *Proceedings of QURATOR 2020 – The conference for intelligent content solutions*, Berin, Germany, 02. CEUR Workshop Proceedings, Volume 2535. 20/21 January 2020.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Shi, T., Keneshloo, Y., Ramakrishnan, N., and Reddy, C. K. (2018). Neural abstractive text summarization with sequence-to-sequence models. Available online (arXiv).

Sjöbergh, J. (2007). Older versions of the rougeeval summarization evaluation system were easier to fool. *Information Processing & Management*, 43(6):1500 – 1505. Text Summarization.

Subramanian, S., Li, R., Pilault, J., and Pal, C. (2019). On extractive and abstractive neural document summarization with transformer language models. Available online (arXiv).

Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA. Curran Associates Inc.

Wu, F., Fan, A., Baevski, A., Dauphin, Y., and Auli, M. (2019). Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Yang, B., Wang, L., Wong, D. F., Chao, L. S., and Tu, Z. (2019a). Convolutional self-attention networks. *Proceedings of the 2019 Conference of the North*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, et al., editors, *Advances in Neural Information Processing Systems 32*, pages 5754–5764. Curran Associates, Inc.

Zhang, H., Cai, J., Xu, J., and Wang, J. (2019). Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China, November. Association for Computational Linguistics.

ZHAW. (2019). German text summarization challenge. Swiss Text Analytics Conference. Available online.