TL;DR : Meaning based Korean Extractive Summarization

Name

Lim Jun Hyeok
Department of Computer Science
dimoteo333@korea.ac.kr

Cho Jae Woo
Department of Computer Science
swjw14@korea.ac.kr

Abstract

Text Summarization has become useful since people do not prefer to read long articles. However, Korean text summarization has only focused on sentence-based summarization, even though the main idea of a text is often shown as smaller size, such as paragraph or morpheme. TL;DR suggests a meaning-based extractive summarization method that uses Korean pretrained BERT[1][2] model trained with 100,000 article datasets. We have trained the model with 2 kinds of different encoders, and the best model shows 33.27 ROGUE-L scores. The codes to reproduce our results are available at https://github.com/dimoteo333/TLDR.

1 Introduction

Text summarization is a task that distilling the most important information from a text to produce an abridged version for a particular task and user.[3] It has been enhanced by several great deep neural architectures such as BERT, BART[4]. This task is usually split into two fields, abstractive summarization and extractive summarization. Abstractive summarization gives summaries containing words and phrases that are not in the original text, whereas extractive summarization forms summaries by choosing the most important parts of the original article. We will focus on the extractive summarization in this project.

There exists several approaches that tried extractive summarization in Korean such as KoBertSum[5], KorBertSum[6]. They both used BertSum as the main architecture and used the different types of Korean pretrained BERT model for training the Korean dataset. Inspired by these works, we write the code based on BertSum and tried different types of pretrained BERT model and summarization layers. Also, we enlarged the dataset field to make the model flexible for various types of input articles.

2 Related Work

BertSum
BertSum[7] suggests a technique to use BERT to text summarization tasks. It changes the input token embedding of the BERT to give input of several sentences. Also, since we need to get an output of summarized sentences, several summarization layers are added at the output of BERT.

- To encode multiple sentences as one input, it inserts a [CLS] token before each sentence and a [SEP] token after each sentence. Also, sentences need to be distinguished from each other, therefore BertSum uses interval segment embeddings which makes odd sentence embedding 0, and even sentence embedding 1.
- The original paper suggests three types of extra summarization layers; simple classifier, transformer, and recurrent neural network layers and gives the output by a sigmoid classifier.

Extractive Summarization
Research work on extractive summarization tries various types of approaches. Several neural network architectures have been used to provide extractive summarization; such as transformer, recurrent neural networks. Also, there are attempts[8]that use document-level features to rerank the extractive summaries.

3 Approach

3.1 Sentence-based, Meaning-based Summarization

Most of the previous works on text summarization are based on sentence-based level summarization; which gets input as sentences. However, the main topic is often consists of smaller paragraphs. Also, by the characteristic of Korean language, Korean data is often transferred by a smaller unit of meaning such as phoneme. To compare the performance of sentence-based input and meaning-based input, we trained the model with a different type of BERT.

For Korean sentence-based model, we choose KoBert[9] for pretrained model, and for a meaning-based model we used KorBert[2] which gets input as a morpheme. To follow the instruction of KorBert, we preprocessed the data by using Korean morpheme analyzer khaiii[10]. We write the code based on BertSum[7], and followed the mainstream of prior researches[6][5].

3.2 Model

Our main architecture of the model follows BertSum[7] model. Since BERT is trained as a masked-language model, the main idea of BertSum is to encode multiple sentences and give embeddings differently. Also, after the sentence vectors are calculated at the output of BERT, several summarization layers are added to give us an output of the summarized sentence. For each sentence $sent_i$, sentences score $\hat{Y}_i$ is calculated and loss will be calculated as binary classification entropy between $\hat{Y}_i$ and Y_i. In this project, a simple sigmoid classifier, transformer is used for summarization layers.

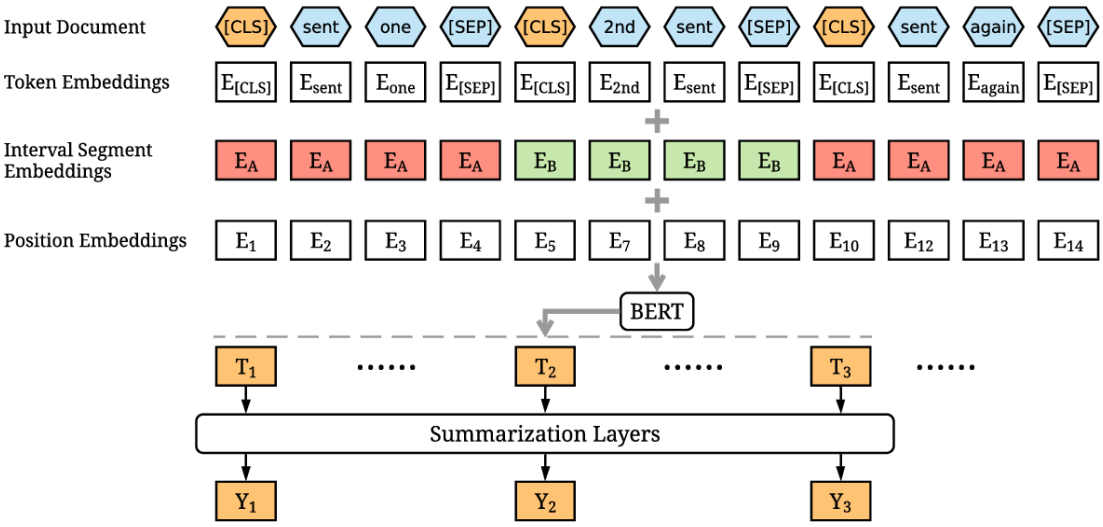

Figure 1: The overview architecture of BertSum model.

- We insert [CLS] token before each sentence and [SEP] token after each sentence. Also, we have used each BERT model dictionary to change our korean language inputs to BERT token embeddings.
- To distinguish multiple sentences, $[sent_1, sent_2, sent_3, sent_4, sent_5$ will be assigned embedding $[E_A, E_B, E_A, E_B, E_A]$.
- **Simple Classifier** Only one linear layer is added at BERT outputs and predicted score is:

$$\hat{Y}_i = \sigma(W_o T_i + b_o)$$

- **Transformer** Inter-sentence transformer extracts document-level features focusing on summarization tasks from the BERT outputs:

$$\tilde{h}^l = LN(h^{l-1} + MHAtt(h^{l-1}))$$
$$h^l = LN(\tilde{h}^l + FFN(\tilde{h}^l))$$

where $h^0 = PosEmb(T)$ and T are the sentence vectors output by BERT, PosEmb is the function of adding positional embeddings to T. LN[11] is the layer normalization operation, and MHAtt[12] is the multi-head attention operation, and l is the depth of the stacked layers. Final output layer is sigmoid classifier:

$$\hat{Y}_i = \sigma(W_o h_i^L + b_o)$$

where h^L is the vector for ith sentence from the Lth layer of transformer.

4 Experiments

4.1 Data

Our model used the text summarization data of AIHub(https://aihub.or.kr/aidata/8054). It contains the original article, human abstractive summarization, extractive summarization data of legal documents, magazine, and newspaper articles. The dataset is consists of 360,000 articles, but for shorter training and testing time, we randomly choose 100,000 articles for training and 10,000 articles for testing. We used original sentences, abstractive extraction to the input of BERT model and gets the output of ranking top-3 sentences as a summary.

4.2 Evaluation method

We have used ROUGE[13] score to evaluate our model. This score is usually used for automatic summarization of texts and machine translation evaluation. The score is calculated by comparing human summarization and model-produced summarization. ROGUE-N scores check how many N-grams are overlapped between human summarization and automated summarization, and ROGUE-L measures the longest matching sequence of words using LCS.

4.3 Experimental details

For model training, we used Google Colab GPU (mostly Tesla T4) for 15,000 steps. Each training took 2-3 hours, and testing took 40 minutes. We used Adam for optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$. Learning rate follows the formula[12] with warming-up on the first 6000 steps.

$$lr = 2e^{-3} \cdot min(step^{-0.5}, step \cdot warmup^{-1.5})$$

4.4 Results

Model	ROUGE-1	ROGUE-2	ROGUE-L
Vanilla BERT+Transformer	24.12	9.03	23.07
KoBert+Classifier	24.97	9.78	23.91
KoBert+Transformer	22.66	8.79	21.68
bert-kor-base[1] +Transformer	25.15	9.82	24.04
KorBert+Classifier	34.40	13.82	33.27

Table 1: Test set results on AIHub Text Summarization dataset using ROGUE F_1.[2]

The experiment results on AIHub datasets are shown in Table 1. As expected, the meaning-based model (KorBert) shows a much higher score than other sentence-based models. Also, contrary to expectation, Korean based BERT models scores are not much different from the vanilla BERT model.

The results scores are quite low compared to other researches[7]. The main reason for this is prior researches used only newspaper articles for a dataset, and ours used a dataset that contained legal documents and various types of articles. Also, our total train steps are set quite low due to the Colab usage limit, therefore the model is underfitted.

5 Analysis

The main purpose of this project is to verify whether the Korean meaning-based summarization method works well compared to the previous sentence-based summarization methods. As we see the results, using morpheme-based KorBert[2] shows much higher scores than other BERT-based models. We can hypothesize these characteristics of text summarization and Korean language make this difference; meaning-based summarization extracts much more useful features in the text.

To verify more differences between them, we tried several more BERT and summarization layer models for an experiment. But errors such as tensor size or token embedding mismatch occurred, and we cannot handle these errors in time. Also, contrary to our expectation, Korean BERT sentence-based models and vanilla BERT model show not much difference. This results from the inconsistency of our test environment and small training steps.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 2019. Association for Computational Linguistics.

[2] ETRI. Korbert. https://aiopen.etri.re.kr/service_dataset.php, 2020.

[3] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, December 1995.

[4] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[5] Hangil Kim. Kobertsum. https://github.com/uoneway/KoBertSum, 2020.

[6] raqoon886. Korbertsum. https://github.com/raqoon886/KorBertSum, 2021.

[7] Yang Liu. Fine-tune bert for extractive summarization. 2019.

[8] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online, July 2020. Association for Computational Linguistics.

[9] SKTBrain. Kobert. https://github.com/SKTBrain/KoBERT, 2019.

[10] Kakao. khaiii. https://github.com/kakao/khaiii, 2020.

[11] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[13] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[1] https://huggingface.co/kykim/bert-kor-base

[2] Since there are no prior researches that used the same AIHub dataset, we included only our experiment results.