Query-Based Keyphrase Extraction from Long Documents

Martin Docekal, Pavel Smrz
Brno University of Technology
idocekal@fit.vutbr.cz, smrz@fit.vutbr.cz

Abstract

Transformer-based architectures in natural language processing force input size limits that can be problematic when long documents need to be processed. This paper overcomes this issue for keyphrase extraction by chunking the long documents while keeping a global context as a query defining the topic for which relevant keyphrases should be extracted. The developed system employs a pre-trained BERT model and adapts it to estimate the probability that a given text span forms a keyphrase. We experimented using various context sizes on two popular datasets, Inspec and SemEval, and a large novel dataset. The presented results show that a shorter context with a query overcomes a longer one without the query on long documents.[1]

Introduction

Keyphrase refers to a short language expression describing the content of a longer text. Due to their concise form, keyphrases can be used for a quick familiarization with a document. They also improve the findability of documents or passages within them. In the bibliographic records, keyphrase descriptors enable flexible indexing.

Whether a text span is a keyphrase depends on the context of that span because a keyphrase for a specific topic may not be a keyphrase for another topic. The presented work builds on the idea that the topic can be explicitly given as an input to the keyphrase extraction algorithm in the form of a query. We approximate such a query with a document's title in our experiments. We also investigate the influence of context size and document structure on the results.

Related Work

Traditional approaches to keyphrase extraction involve graph-based methods, such as TextRank (Mihalcea and Tarau 2004) and RAKE (Rose et al. 2010). Recently, many types of neural networks have been used for the task (Lin and Wang 2019; Sahrawat et al. 2020). Most of the deep learning work assumes the existence of a title and an abstract of the document and extracts keyphrases from them because they struggle with longer inputs such as whole scientific articles (Kontoulis, Papagiannopoulou, and Tsoumakas 2021). Some works try to overcome this limitation by first creating a document summary and then extracting keyphrases from it (Kontoulis, Papagiannopoulou, and Tsoumakas 2021; Liu and Iwaihara 2021). Our research follows an alternative path, compensating for the limited context by a query specifying a topic.

Model

First, a document is split into parts (contexts), which are further processed independently. Then, the devised model estimates the probability that a given continuous text span forms a keyphrase. It looks for boundaries t_s and t_e, corresponding to the text span's start and end, respectively. The inspiration for this approach comes from the task of reading comprehension, where a similar technique is used to search for potential answers to a question in an input text (Devlin et al. 2019). Formally, the model estimates probability:

$$P(t_s, t_e|x) = P(t_s|x)P(t_e|x)\,, \tag{1}$$

where x is the input sequence. It assumes that the start and the end of a text span $< t_s, t_e >$ are independent. The probabilities $P(t_s|x)$ and $P(t_e|x)$ are obtained in the following way:

$$P(t_*|x) = \text{sigmoid}(w_*^T \text{BERT}(x)[*] + b)\,, \tag{2}$$

where $*$ stands for the *start* and *end* positions. Weights w_s and w_e are learnable vectors, b is the scalar bias and $\text{BERT}(x)[i]$ is BERT (Devlin et al. 2019) vector representation of a token from sequence x on position i. See the model illustration in Figure 1.

The task of predicting whether a given token is the start token of a span or the end token could be seen as binary classification with two classes *start/end* and *not start/not end*, respectively. The binary cross-entropy (BCE) loss function is used for training in the following way:

$$\text{BCE}(v_s, g_s) + \text{BCE}(v_e, g_e), \tag{3}$$

where v_s is a vector of probabilities that a token is the start token of a span, for each token in the input, and v_e is vector computed analogously, but for the ends. The g_s and g_e are vectors of ground truth probabilities of starts or ends, respectively.

Copyright © 2022 by the authors. All rights reserved.

[1]The code is available at https://github.com/KNOT-FIT-BUT/QBEK.

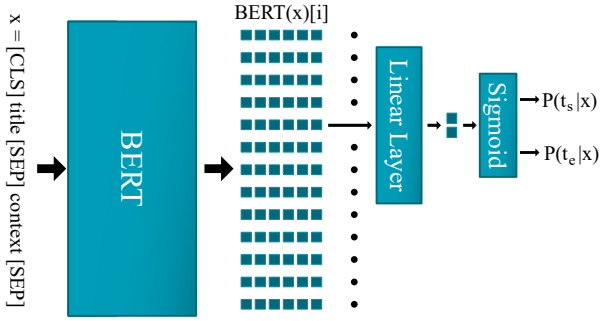

Figure 1: Illustration of our model with query at the input.

We work with two types of inputs. One consists of a text fragment such as a sentence, and the other uses a query (document title) and a text segment, separated by a special token. Various context sizes are explored in our experiments. The context size determines how big is the document part the model sees at once. Every context part of a document is processed independently. The final list of keyphrases is created by collecting keyphrase spans with their scores and selecting the top ones.

Datasets

Besides two standard datasets for keyphrase extraction, we created and used a novel dataset of long documents, referred to a Library, and we also prepared an unstructured version of the SemEval-2010 dataset. A comparison of the datasets is given in Table 1.

dataset	train	val.	test	sentences (train)
SemEval-2010	130	14	100	66 428 (72%)
Unstructured-SemEval-2010	130	14	100	45 346 (67%)
Inspec	1 000	500	500	5 894 (25%)
Library	48 879	499	499	298 217 589 (94%)

Table 1: The number of documents in each split along with the total number of sentences in a train set. The percentage in the sentences column is the proportion of sentences without keyphrases.

We had to annotate the spans that represent given keyphrases in the text as the discussed datasets provide just a list of associated keyphrases with no information about their actual positions. The search was case insensitive and the Porter stemmer was utilized for the SemEval and Hulth2003 (Inspec) datasets. For the Library dataset, as it is in Czech, the *MorphoDiTa* lemmatizer[2] was used.

SemEval-2010 (Kim et al. 2010) consists of whole plain texts from scientific articles. The dataset provides keyphrases provided by authors and readers. As it is common practice (Kim et al. 2010; Kontoulis, Papagiannopoulou, and Tsoumakas 2021), we use a combination of both in our experiments. Our validation dataset was created by randomly choosing a subset of the train set. As the original data source does not explicitly provide the titles, which we need to formulate a query, we have manually extracted the title of each document from the plain text.

Documents in this dataset have a well-formed structure. They contain a title and abstract and are divided into sections introduced with a headline. As we want to investigate the influence of such structure on results, we have made an unstructured version of this dataset. We downloaded the original PDFs and used the GROBID[3] to get a structured XML version of them. We kept only the text from the document's main body while the parts such as title, abstract, authors, or section headlines were removed. Nevertheless, document keyphrase annotations remain the same. We call this dataset Unstructured-SemEval-2010. The name SemEval is used to name these two collectively.

Inspec (Hulth 2003) contains a set of title-abstract pairs collected from scientific articles. For each abstract, there are two sets of keyphrases — *controlled*, which are restricted to the Inspec thesaurus, and *uncontrolled* that can contain any suitable keyphrase. To be comparable with previous works (Hulth 2003; Liu and Iwaihara 2021), we used only the *uncontrolled* set in our experiments.

Library is a newly created dataset that takes advantage of a large body of scanned documents, provided by Czech libraries, that were converted to text by OCR software. This way of getting the text is unreliable, so the dataset contains many errors on the word and character level. The dataset builds on the documents where the language was recognized as 'Czech' by the OCR software.

All used documents in the original data source are represented by their significant content (the average number of characters per document is 529 276) and metadata. The metadata contains (not for all) keyphrases and document language annotations. We did not ask annotators to annotate each document. Instead, we selected metadata fields used by librarians as keyphrase annotations. So, our data and metadata come from the real-world environment of Czech libraries. We have filtered out all documents with less than five keyphrases.

Documents come from more than 290 categories. Various topics such as mathematics, psychology, belles lettres, music, and computer science are covered. Not all annotated keyphrases can be extracted from the text. Considering the lemmatization, the test set annotations contain about 53% of extractive keyphrases. Bibliographic field Title (MARC 245[4]) is used as the query. Note that the field may contain additional information to the title, such as authors.

Experimental Setup

The implemented system builds on PyTorch[5] and PyTorch Lightning[6] tools. The BERT part of the model uses the

[2] http://hdl.handle.net/11858/00-097C-0000-0023-43CD-0

[3] https://github.com/kermitt2/grobid

[4] https://www.loc.gov/marc/bibliographic/bd245.html

[5] https://pytorch.org/

[6] https://www.pytorchlightning.ai/

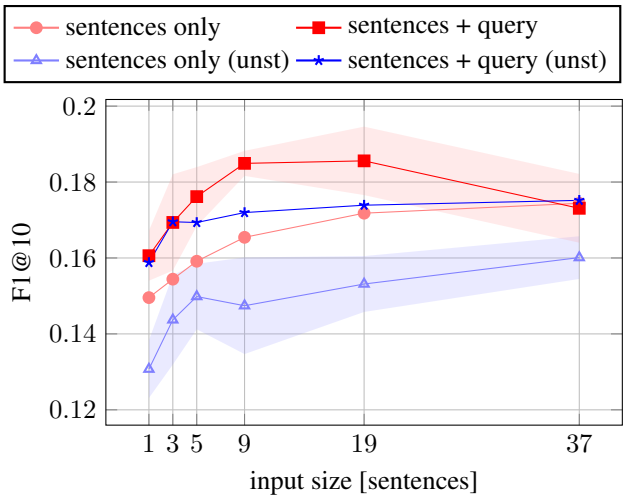

Figure 2: Results for SemEval-2010 and Unstructured-SemEval-2010 test set. The light red and blue areas are confidence intervals with a confidence level of 0.95. Each point corresponds to an average of five runs.

implementation of BERT by *Hugging Face*[7] and it is initialized with pre-trained weights of *bert-base-multilingual-cased*. These weights are also optimized during fine-tuning.

The Adam optimiser with weight decay (Loshchilov and Hutter 2017) is used in all the experiments. The evaluation during training on the validation set is done every 4 000 optimization steps for the Library dataset and every 50 steps for Inspec (25 for whole abstracts with titles). For SemEval datasets, the number of steps differs among experiments. Early stopping with patience 3 is applied, so the training ends when the model stops improving. Batch size 128 is used for experiments with the Library dataset, and batch size 32 is used for Inspec and SemEval datasets. The learning rate 1E-06 is used for the experiments with SemEval datasets, while it is set to the value of 1E-05 for all other datasets. Inputs longer than a maximum input size are split into sequences of roughly the same size in a way that forbids splitting of keyphrase spans. In edge cases (split is not possible), the input is truncated. No predicted span is longer than 6 tokens.

The official script for SemEval-2010 is used for evaluation. However, the normalization of keyphrases is different for the Library dataset as we have used the mentioned Czech lemmatizer instead of the original stemmer. We use the F1 over the top five (F1@5) candidates for the Library dataset and over the top ten (F1@10) for the rest.

Experiments

The performed experiments investigate the influence of queries on four different datasets, the output quality with various context sizes, and the impact of the document structure.

The first set of experiments is performed on long documents with a well-formed structure from the SemEval-2010 dataset and compares them with SemEval's unstructured version. Figure 2 shows that inputs with a query are better than those without a query, but the last point. For the structured input, it can be seen that from the point with 19 sentences, the performance of input with query stops with the fast growth. It correlates with Figure 3 showing the saturation of the model input. Notice that from 19 sentences, the input becomes more saturated, and the splitting strategy starts shrinking contexts.

[7]https://huggingface.co/

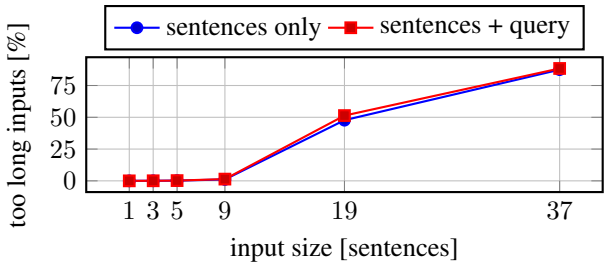

Figure 3: The proportion of inputs longer than the maximum input size for SemEval-2010 train set.

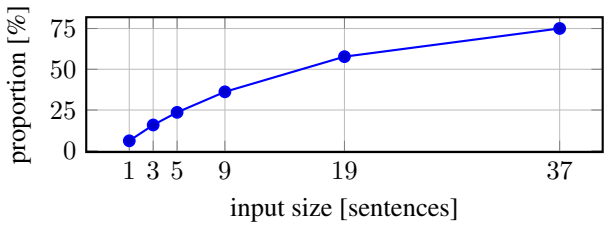

Figure 4: The proportion of contexts containing at least one section headline as a substring for SemEval-2010 test set.

It is not surprising that the nominal values are lower for unstructured inputs. On the other hand, it is clear that the query has a bigger influence on the unstructured version, especially for short context sizes, because the average absolute difference among results (with- and without a query) for each context size is 2.2% compared to 1.29% for the structured one.

Looking at the curve of results with a query on an unstructured version, we assume that the model can exploit a document structure without explicitly tagging it with special tokens because additional context size above the three sentences is not much beneficial compared to the case with the document structure. This hypothesis is supported by the fact that the proportion of the context containing structured information grows with context size, as is demonstrated in Figure 4 showing the proportion of contexts containing a section headline.

The second set of experiments was performed on our Library dataset. The results can be seen in Figure 5. We have chosen F1@5 because only approximately half of the documents have ten and more keyphrases. Again, the results show that queries are beneficial. Also, it can be seen that the shape of the query curve is similar to Unstructured-SemEval-2010. The average absolute difference between the version with and without query is now 3.1%. For F1@10, it

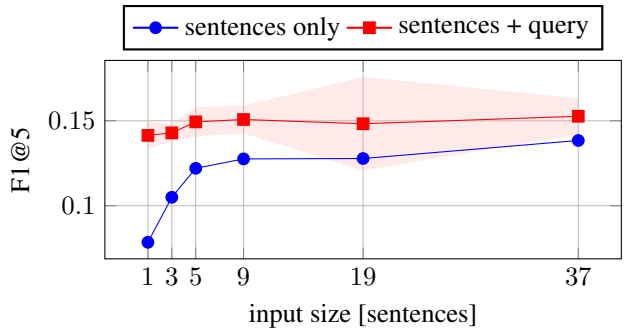

Figure 5: Results for Library test set for various context sizes. The light red area symbolizes a confidence interval with a confidence level of 0.95. Each point is average from three runs.

		Inspec F1@10	**SemEval 2010 F1@10**
TextRank		15.28	6.55
KFBS + BK-Rank		**46.62**	15.59
DistilRoBERTa + TF-IDF		-	16.2
context [sentences]	**query**		
whole document	✗	39.67	-
1	✗	40.26	14.96
	✓	39.95	16.06
19	✗	-	17.18
	✓	-	**18.56**

Table 2: Comparison of achieved results with other work. KFBS + BK-Rank and TextRank is from (Liu and Iwaihara 2021). The DistilRoBERTa + TF-IDF is from (Kontoulis, Papagiannopoulou, and Tsoumakas 2021). Our results are averages from five runs.

is 2.3, which is close to the value for the unstructured version of SemEval.

The last set of experiments is done on the Inspec dataset, which has only titles and abstracts. The purpose is to investigate the influence of a query on short inputs containing mainly salient sentences. Results are summarized in Table 2, which also compares our results with other works. It shows that the results for a single sentence, a single sentence with a title, and whole abstract with a title are similar. The explanation can be that the abstract contains mainly salient sentences containing keyphrases, and also, the abstract itself defines the topic of the article. A similar observation is presented in (Liu and Iwaihara 2021), where the version without summarization gives similar results as the extraction performed on a summary.

Conclusions

We have conducted experiments that show that query-based keyphrase extraction is promising for processing long documents. Our experiments show the relationship between the context size and the performance of the BERT-based keyphrase extractor. The developed model was evaluated on four datasets; one of them is non-English. The datasets allowed us to find when the query-based approach is beneficial and when not. It was shown that a query gives no benefit when extracting keyphrases from abstracts. On the other hand, it is beneficial for long documents, particularly those without a well-formed document structure on short context sizes.

Acknowledgment

This work was supported by the Technology Agency of the Czech Republic, Grant FW03010656 – MASAPI: Multilingual assistant for searching, analysing and processing information and decision support. The computation used the infrastructure supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

References

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 NAACL Conference: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Hulth, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 EMNLP Conference*, 216–223.

Kim, S. N.; Medelyan, O.; Kan, M.-Y.; and Baldwin, T. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th SemEval*, 21–26. Uppsala, Sweden: Association for Computational Linguistics.

Kontoulis, C. G.; Papagiannopoulou, E.; and Tsoumakas, G. 2021. Keyphrase extraction from scientific articles via extractive summarization. In *Proceedings of the Second SDP Workshop*, 49–55. Online: Association for Computational Linguistics.

Lin, Z.-L., and Wang, C.-J. 2019. Keyword extraction with character-level convolutional neural tensor networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 400–413. Springer.

Liu, T., and Iwaihara, M. 2021. Supervised learning of keyphrase extraction utilizing prior summarization. In Ke, H.-R.; Lee, C. S.; and Sugiyama, K., eds., *Towards Open and Trustworthy Digital Societies*, 157–166. Cham: Springer International Publishing.

Loshchilov, I., and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on EMNLP*, 404–411.

Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory* 1:1–20.

Sahrawat, D.; Mahata, D.; Zhang, H.; Kulkarni, M.; Sharma, A.; Gosangi, R.; Stent, A.; Kumar, Y.; Shah, R. R.; and Zimmermann, R. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *ECIR*, 328–335. Springer.