StructExtSum – Bulgarian Legislation Text Extractive Summarization by Structure Understanding

Valentin Zmiycharov, Gergana Lazarova, Todor Tsonkov, Ivan Koychev

Sofia University "St. Kliment Ohridski", Faculty of Mathematics and Informatics, 5, James Baucher Blvd, 1164 Sofia, Bulgaria

Abstract

Determining the most important part of documents in the legal field has great value because the texts are very long and hard to understand. In this paper, a new approach is proposed for long EU legislative documents in Bulgarian. The approach uses the inner structure of the documents. It relies on the structure of the text and trains a model to predict the importance of each sentence of the text. A dataset has been collected consisting of texts in Bulgarian with initially 1739 documents and 1657 after cleansing. Multiple experiments have been conducted. The results show that the proposed approach improves the rouge score and performs better than the other algorithms.

Keywords

Bulgarian text summarization, long text summarization, legislation summarization

1 Introduction

Automatic summarization of legislative documents is a rather difficult task due to the length of the texts which makes them hard to understand and process. Using Bulgarian makes the task even more challenging due to the relative lack of pre-trained models for summarization compared to more popular languages like English.

The existing models for text summarization in general have been trained on shorter texts (such as posts from social networks or news feeds) than legislative documents. There are some text summarization approaches designed for longer documents, which we will overview in the next section. In the current paper, we have selected a dataset of EU legislative documents translated into Bulgarian.

There are two basic approaches to make summarization – extractive and abstractive. Existing abstractive models don't have good prediction accuracy against long texts in Bulgarian and we have decided to focus on extractive approaches.

The contributions of the paper are:

- Data collection (Bulgarian legislative documents have been collected from [15]) The data has been cleaned and preprocessed.
- A new approach for long texts is proposed StructExtSum. To process long texts, we use the inner structure of the legislative documents and teach a model to predict the importance of each sentence in the section. The features used in the learning process will be discussed in detail in the section Experiments.
- The proposed approach is compared to Baseline which extracts the first sentences of the full text, an extractive unsupervised summarizer based on TF-IDF and an extractive k-Means multilingual SlavicBERT summarizer (Bulgarian, Czech, Polish, and Russian).

EMAIL: valentin.zmiycharov@gmail.com (A. 1); gerganal@fmi.uni-sofia.bg (A. 2); ttsonkov@gmail.com (A. 3); koychev@fmi.uni-sofia.bg (A. 4)



Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Education and Research in the Information Society, October 13-14, 2022, Plovdiv, Bulgaria

2 Related work

The section presents an overview of the existing approaches for text summarization of long documents, which is the case with legal documents.

Nakao [9] presents an algorithm for text summarization using the thematic hierarchy of a text. The proposed algorithm is intended to generate a one-page summary. Based on the ratio of source text size to a given summary size, the algorithm generates a summary with some breaks to indicate thematic changes. This algorithm cannot easily be adapted to summaries with dynamic length. Another approach is to combine extractive and abstractive models (Wang et al.) [11]. In the extraction phase, it creates a graph model to extract key sentences. In the abstraction phase, it uses a recurrent neural network, based on an encoder-decoder, and devises pointer and attention mechanisms to generate summaries.

Xiao and Carenini [12] focus on extracting informative sentences from a given document (without dealing with redundancy), especially when the document is relatively long (e.g., scientific articles). They rely on section information to guide the generation of summaries. Global and local contexts are considered when deciding if a sentence should be included in the summary. This approach struggles when there is not a well-defined structure of sections which is the case with the legislative documents.

Vaswani et al [10] presented the Transformer architecture, which establishes a new single-model state-of-the-art BLEU score on two machine translation tasks. The architecture consisted of feed-forward networks and attention mechanisms. The basic architecture of a Transformer is based on the encoder-decoder model and is especially suitable for summarization because it can handle sequential data. Yet, the data does not need to be processed in order (for instance the beginning of the text does not have to be processed before the end). This is very useful for parallel training and reduces the time needed to train the transformers. The encoder takes all the input and encodes it into a vector containing the numerical representation of the text. Then the decoder decodes the vector and produces the summary. The datasets used for training can be big and thus exist in pre-trained models such as BERT (Bidirectional Encoder Representations from Transformers)[18]. They have been trained with huge general language datasets and can be fine-tuned to specific language tasks. We tried to use BERT but couldn't achieve better results than the other algorithms presented in the article.

Zmiycharov et al [1] presented results from experiments with summarization approaches on long legislative documents in English. The following algorithms are compared: PreSumm [3], LEGAL-BERT [4], T5 [5], and "K-Means and BERT" based Summarization [2]. A comparison of those algorithms on scientific papers is presented in [7]. The used language models are originally trained and developed for the English language, therefore the task is even more challenging for the Bulgarian language, because it is a low-resource language. The only work that explores text summarization for the Bulgarian language is presented by Taushanov et al [6]. The work presents and trains a LSTM model and compares extractive and abstractive approaches for text summarization on a dataset that consists of short news articles in Bulgarian. The results from this comparison show that in this case, the algorithms for abstractive text summarization perform better than the extractive ones.

The difference between the existing approaches and the one provided in the paper is that we focus on the inner structure of the text and on extracting the features of each sentence.

3 Dataset collection and preprocessing

The dataset contains legislative documents and their summaries in Bulgarian from European Union Law ([15]). Some summarize more than one document. In these cases, the content of the full documents is concatenated in the order they are mentioned.

The data was collected on April 7th, 2022. Initially, there were 1739 items in the dataset. We found some invalid data and cleaned it. It contains texts that are not translated into Bulgarian and summaries, which have more words than the full text. After the cleaning process, the dataset contains 1657 documents. Figure 1 shows the ratio between the number of words in the summaries and the full texts. The line plots the relationship between the summaries' word count and the full texts' word count. The confidence interval, which extends the regression line, corresponds with the standard error of the estimate.



Figure 1: Each point represents an item in the dataset. The X-axis shows the number of words in the summary and Y axis the number of words in the full text. As seen by the diagram, the ratio between the word count differs a lot and there are outliers in terms of word count.

Both summaries and full texts often represent text bullets and are not well-structured text. The texts are divided into paragraphs, some of which have headings but the headings of the paragraphs are various and depend on the current legal document. Some paragraphs are united into sections. Unfortunately, the sections vary a lot and again do not appear in all documents (408 types of sections, only a few are present in all documents and most of the sections appear only in one document). To focus on the relevant information, we did a preprocessing. The sections which appear most frequently in the summaries are: "Key points", "Date of entry into force", "Background", and "Main documents". We extracted only the Key points section and it is the target we are aiming at predicting. Example: [16]. In the full documents, we ignored the headers like "Official Journal of the European Union" and the references. Some full texts contain multiple documents in one URL. We made sure to concatenate them. Example: [17]





Figure 2: An example of a summary of two legal documents. The two documents are concatenated.

Full documents are represented by a hierarchical structure. They have 3 types of headers: title, chapter, and article. The presence of each of them is not consistent across documents, as some documents have only titles, some have only articles, etc. The title aims to start a new document. There are cases where the full texts contain multiple documents. Chapters group logical parts of the texts and the articles represent legislation articles. Each chapter and article has a number and name, for example, CHAPTER I GENERAL PROVISIONS, Article 1 Purpose.

4 StructExtSum

Our proposed approach (StructExtSum) extracts features for each sentence in the full text and calculates the rouge score of this sentence against the original summary. First, we calculate the rouge 1 F1 score of each sentence of the full text against the original summary. We normalize these values by dividing them by the maximum score. Then we extract features for each sentence, normalize them against respective maximum values, and train linear regression to predict the rouge score. We split all the texts into 10 folds and trained and got the rouge scores for each fold. After having the predicted scores for all sentences, we order them in descending order and concatenate them until we reach the word count in the original summary. The result is then calculated on all predicted summaries against the original ones.

We use the following features for the linear regression:

1. Order of a sentence – each sentence order is divided by the total number of sentences in the full text

2. Order of an article – Based on the structure of the documents, they are split into articles. Each sentence is part of one article. This feature shows the number of the article, the sentence is part of, divided by the total number of articles.

3. Order of a sentence in an article – the order of the sentence in the article divided by the total number of sentences in the article.

4. Order of a chapter – Based on the structure of the documents, they are split into chapters. Each sentence is part of exactly 1 chapter. This feature shows the number of the chapter, the sentence is part of, divided by the total number of chapters.

5. Order of a sentence in a chapter – the order of the sentence in the chapter divided by the total number of sentences in the chapter.

6. Stop words ratio – The number of stop words in the sentence divided by the total number of words in the sentence.

7. TF-IDF – The extractive approach, explained in section 2 relies on this feature. It shows the TF-IDF score for each sentence regarding the other sentences in the full text.

8. Words count – The number of words in the sentence divided by the number of words in the longest sentence in the text.

All features have values between zero and one.

5 Experiments

The experiments of this section aim to compare the presented document-structure aware approach with other approaches for text summarization.

5.1 Experiments Design

We tried abstractive and extractive approaches and decided to focus on the extractive due to the bad performance of the abstractive ones.

The most widely used metric for the evaluation of text summarization is rouge (Recall-Oriented Understudy for Gisting Evaluation). Rouge is a set of metrics used for evaluating automatic summarization and machine translation software. The metrics compare an automatically produced summary to a human-produced summary. Rouge-N refers to the overlap of n-gram between the system and reference summaries. Rouge-L refers to Longest Common Subsequence (LCS) based statistics. The longest common subsequence problem considers sentence-level structure similarity naturally and identifies the longest co-occurring in sequence n-grams automatically. In particular rouge-1, rouge-2 and rouge-L F1 scores were used in the conducted experiments.

We started generating summaries that have the same length as the original summary. This way the precision and recall are the same and the problem can be defined as finding the best sentences, which summarize the full text with a fixed amount of words. This way we avoid the problem of changing the F-score due to generating larger or smaller summaries and focus entirely on the relevance of the

sentences and not on the length of the summary. For all experimented approaches we selected only sentences with at least 3 words.

The proposed approach is compared to three algorithms described in the following subsections.

5.1.1 Baseline

We assumed that the most important information is at the beginning of the document/section. Therefore, our baseline approach collects the first k consecutive sentences from the full texts. The number k of sentences is decided based on the number of words in the original summary. We continue adding sentences until we reach the number of words. Abstractive approaches like Pegasus [13] also use the first part of the text. The results from the experiment reported below provide evidence that this assumption is good enough for a baseline.

5.1.2 TF-IDF Summarizer

We have also tested basic extractive summarization (Malik, 2019) [14]. The first step of the algorithm is to split the full text into a list of sentences. After that all special characters and stop words are removed. Then all sentences are tokenized. Next, the weighted frequency of occurrences of all words must be calculated. The weighted frequency of each word can be found by dividing its frequency by the frequency of the most occurring word. After that, the words in the original sentences are replaced by their respective weighted frequency. The weighted frequency for the words removed during preprocessing is zero. For each sentence, the sum of weighted frequencies is calculated. Only sentences with more than three words are evaluated to avoid the ones that do not contain enough information. Finally, the sentences are sorted in descending order by the sum of the weighted frequencies. The summary contains the sentences at the beginning of the ordered list. The number of sentences to be selected is based on the ratio between the number of sentences in the training dataset. The algorithm does not require training and is entirely based on the content of the full document.

5.1.3 Extractive summarization with SlavicBERT and kMeans

The proposed approach is also compared to the extractive summarization algorithm proposed by Miller [2]. The algorithm works in the following way: the document is tokenized into clean sentences. The tokenized sentences are passed to the BERT model for inference to output embeddings. The embeddings are then clustered with K-Means. The embedded sentences that were closest to the centroid are selected as the candidate summary sentences. The algorithm uses the core BERT implementation. For Bulgarian document summarization, the core BERT model was replaced with the multilingual SlavicBERT. The average ratio of the original texts and their summaries is calculated. The generated summary length is the average ratio, multiplied by the current text length.

5.2 Analysis of results

Table 1 shows the results of the different approaches. Linear regression approach based on structure features (StructExtSum) improves the Extractive based on TF-IDF by 2.5% and baseline by 6.5%. While baseline and Extractive do not require training and are tested on all documents, the last approach was tested using 10-fold cross-validation on all documents.

Algorithm	rouge-1	rouge-2	rouge-L
Baseline (first N sentences)	28.22	9.97	22.87
TF-IDF	29.29	10.91	23.00
StructExtSum	30.04	11.15	23.51
Summarization with	19.69	7.33	17.53
SlavicBERT and K-Means			

Table 1 shows the performance of StructExtSum compared to the other algorithms.

StructExtSum can easily be applied to other languages if the features for each sentence can be extracted. To further validate the impact of the algorithm we created a legislation dataset in English. We used the same approach to download, preprocess, extract features and generate summaries. Table 2 shows that StructExtSum again manages to outperform baseline and TF-IDF approaches.

Algorithm	rouge-1	rouge-2	rouge-L
Baseline (first N sentences)	36.23	11.10	29.05
TF-IDF	37.31	11.77	29.32
StructExtSum	38.99	12.20	30.16
Summarization with BERT	36.82	11.65	28.53
and K-Means			

Table 2 shows the performance of the StructExtSum algorithm compared to the other algorithms forEnglish texts.

6 Conclusion

Popular NLP models do not produce good enough results for text summarization in less common languages like Bulgarian. When texts are long and contain information about specific fields like law things get even more complicated because there are fewer resources. Better understanding the structure of the document and extracting the simple features of each sentence enables the invention of a promising extractive summarizer, which can be easily transferred to different languages and fields. Future work includes extending proposed features with embeddings and other useful information. Different algorithms to calculate the score of each sentence will be considered. We will also aim at experimenting with more datasets in a wider variety of languages, fields, and lengths of texts.

7 Acknowledgements

This research is partially supported by Project UNITe BG05M2OP001-1.001-0004 funded by the OP "Science and Education for Smart Growth" and also co-funded by the EU through the ESI Funds

8 References

- Valentin Zmiycharov, Milen Chechev, Gergana Lazarova, Todor Tsonkov, and Ivan Koychev. 2021. A Comparative Study on Abstractive and Extractive Approaches in Summarization of European Legislation Documents. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pages 1645–1651
- [2] Derek Miller. 2019. Leveraging BERT for Extractive Text Summarization on Lectures. arXiv e-prints, page arXiv:1906.04165.
- [3] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. arXiv e-prints, page arXiv:1908.08345.
- [4] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. arXiv e-prints, page arXiv:2010.02559.
- [5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Textto-Text Transformer. arXiv e-prints, page arXiv:1910.10683.
- [6] Nikola Taushanov, Ivan Koychev and Preslav Nakov. 2018. Abstractive Text Summarization with Application to Bulgarian News Articles. Third International Conference, Computational linguistics in Bulgaria.

- [7] Todor Tsonkov, Gergana Lazarova, Valentin Zmiycharov, Ivan Koychev, A Comparative Study of Extractive and Abstractive Approaches for Automatic Text Summarization on Scientific Texts, Education and Research in the Information Society, 2021, ISSN (online):1613-0073
- [8] Franck Dernoncourt, Mohammad Ghassemi, and Walter Chang. 2018. A repository of corpora for summarization. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).
- [9] Yoshio Nakao. 2000. An algorithm for one-page summarization of a long text based on thematic hierarchy detection. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pages 302–309, Hong Kong. Association for Computational Linguistics.
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv e-prints, page arXiv:1706.03762.
- [11] S. Wang, X. Zhao, B. Li, B. Ge, and D. Tang. 2017. Integrating extractive and abstractive models for long text summarization. In 2017 IEEE International Congress on Big Data (BigData Congress), pages 305–312.
- [12] Wen Xiao and Giuseppe Carenini. 2019. Extractive Summarization of Long Documents by Combining Global and Local Context. arXiv e-prints, page arXiv:1909.08089.
- [13] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. arXiv e-prints, page arXiv:1912.08777.
- [14] Usman Malik. 2019. Text summarization with nltk in python. https://stackabuse.com/text-summarization-with-nltk-in-python/
- [15] https://eur-lex.europa.eu/homepage.html?locale=bg
- [16] https://eur-lex.europa.eu/legal-content/BG/TXT/HTML/?uri=LEGISSUM:4428638&from=BG
- [17] https://eur-lex.europa.eu/legal-content/BG/TXT/HTML/?uri=CELEX:32006D0871&from=BG
- [18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding