A Comparative Study on Abstractive and Extractive Approaches In Summarization of European Legislation Documents

Valentin Zmiycharov
FMI, Sofia University
"St. Kliment Ohridski"
Sofia, Bulgaria
valentin.zmiycharov
@gmail.com

Milen Chechev
FMI, Sofia University
"St. Kliment Ohridski"
Sofia, Bulgaria
milen.chechev@fmi.uni-sofia.bg

Gergana Lazarova
FMI, Sofia University
"St. Kliment Ohridski"
Sofia, Bulgaria
gerganal@fmi.uni-sofia.bg

Todor Tsonkov
FMI, Sofia University
"St. Kliment Ohridski"
Sofia, Bulgaria
ttsonkov@gmail.com

Ivan Koychev
FMI, Sofia University
"St. Kliment Ohridski"
Sofia, Bulgaria
koychev@fmi.uni-sofia.bg

Abstract

Extracting the most important part of legislation documents has great business value because the texts are usually very long and hard to understand. The aim of this article is to evaluate different algorithms for text summarization on EU legislation documents. The content contains domain-specific words. We collected a text summarization dataset of EU legal documents consisting of 1563 documents, in which the mean length of summaries is 424 words. Experiments were conducted with different algorithms using the new dataset. A simple extractive algorithm was selected as a baseline. Advanced extractive algorithms, which use encoders show better results than baseline. The best result measured by ROUGE scores was achieved by a fine-tuned abstractive T5 model, which was adapted to work with long texts.

1 Introduction

Automatic summarization of legislation documents is a rather challenging task, because they usually are very long and hard to understand. Therefore, any progress on this task might have great value for many businesses.

Most of the existing methods for text summarization are designed for a relatively short text (such as news, web pages, etc.). Most of the available datasets consist of short summaries of articles, news, etc. in which the expected summary is a few sentences long. However, in the case of legal documents both the text and the summary are longer. Dernoncourt et al. (2018) provide a comprehensive overview of the current datasets for summarization, including CNN/Daily Mail (Hermann et al., 2015), Gigaword (Graff and Cieri, 2003), LCSTS (Hu et al., 2015) and others. Noticeably, most of the larger scale summarization datasets consist of relatively short documents.

The paper's aim is to research the applicability of different algorithms for text summarization on a new dataset, called EU legislation documents. The collected documents were parsed, cleaned and prepared to be used for data summarization training by defining pairs of full text and corresponding summary.

There are two main approaches for text summarization: extractive and abstractive. Extractive summarization means identifying important parts of the text and concatenating them verbatim to produce a summary which is a subset of the sentences from the original text. Abstractive summarization aims to make algorithms that are able to "understand" the whole text and to generate a new shorter text that conveys the most important information from the original one (Sciforce, 2019).

The algorithm, which was selected as baseline, uses a classical approach for generating extractive summaries - sentence importance evaluation and combining the highly scored sentence to generate the summary (Malik, 2019). In this paper, we report results from different algorithms compared to the baseline algorithm. The compared algorithms include extractive and abstractive PreSumm (Liu and Lapata, 2019), which generates state-of-the-art results for CNN/DailyMail datasets (Hermann et al., 2015), a fine-tuned abstractive T5 model (Raffel et al., 2019), an extractive summarizer which uses BERT (Miller, 2019), an extractive summarizer which uses LEGAL-BERT (Chalkidis

Proceedings of Recent Advances in Natural Language Processing, pages 1645–1651
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_184

et al., 2020).

2 Related Work

This section presents existing approaches for text summarization. Some are especially related to our work because they focus on long texts and this is the case with the legal documents that we use.

Xiao and Carenini (2019) 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 taken into account 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 legislation documents.

Nakao (2000) 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., 2017). In the extraction phase, it creates a graph model to extract key sentences. In the abstraction phase, it uses a recurrent neural network based encoder-decoder, and devises pointer and attention mechanisms to generate summaries.

Vaswani et al. (2017) 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 mechanism. 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 pre-trained systems such as BERT (Bidirectional Encoder Representations from Transformers). They have been trained with huge general language datasets and can be fine-tuned to specific language tasks. The following algorithms we experimented with also rely on the Transformer architecture: PreSumm (Liu and Lapata, 2019), LEGAL-BERT (Chalkidis et al., 2020), T5 (Raffel et al., 2019).

Pegasus (Zhang et al., 2019) is a state-of-the-art NLP deep-learning algorithm for abstractive text summarization. It can be used for both extractive and abstractive summarization but the abstractive is more challenging because when the text is long, it should be understood, processed and a new text should be generated.

3 Dataset Collection

In order to create a dataset on which to compare the algorithms, we collected legislation documents. The data was preprocessed and only the relevant information for the task was left.

The dataset consists of short, easy to understand explanations of the main legal acts passed by the EU, intended for a general audience. Most cover the main types of legislation passed by the EU: directives, regulations and decisions. But some cover other documents, such as international agreements. The summaries are grouped into 32 policy fields, and each links to the full, official version of the act. Summaries are not available for legal acts that are considered to be already sufficiently short/clear or aimed exclusively at a specialist audience (Publications Office of the European Union, 2020). The information is provided by the Publications Office of the European Union and is publicly available. It was retrieved on January 12, 2020.

After the data was collected, we analyzed it and cleaned it in order to focus on the problem of text summarization. For the summaries we extracted only the Key points section and for the full documents we removed the references to external documents. There are some summaries that combine more than one full legislation document: 169 summaries are a summary of two documents and 50 summaries are a summary of more than two documents. In these cases, the full documents are concatenated in the mentioned order.

In order to be able to evaluate a wider variety of algorithms and remove potentially incorrect data, the outliers were handled in the following ways:

- 49 summaries with more words than full text

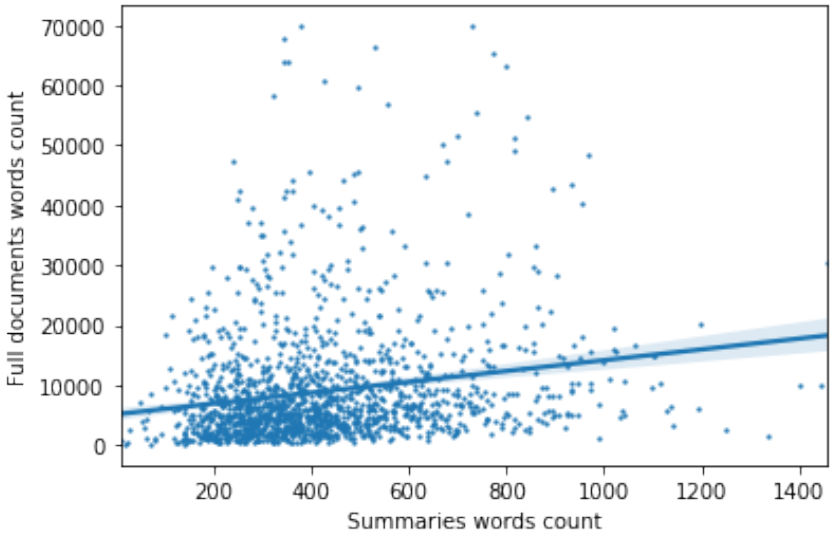

Figure 1: The dots represent the number of summaries and full texts words for each example. The big distance between the points and the regression line shows that the ratio between the number of words in the full documents and summaries may vary based on the content of the legislation documents.

were removed from the dataset.

- 86 summaries, which summarize more than one document and the same document exists for more than one summary were removed from the dataset.
- 18 summaries with word count ratio bigger than 200 were removed. Words ratio has a mean value of 28 and value of 29 at third quantile.
- Nine full documents with more than 75000 words were removed. Full document word count has a mean value of 9615 and value of 11141 at third quantile.
- Three summaries with more than 1500 words were removed. Summaries' word count has a mean value of 429 and value of 530 at third quantile.
- Two full documents with more than 2000 sentences were removed.
- Three full documents with sentence ratio more than 80 were removed.
- 29 summaries which summarize more than two documents were removed from the dataset.

During the initial collection the dataset contained 1750 records. After the cleaning there are 1563 summaries (10.7% of the dataset was removed). The mean length of the summaries and full texts is 424 and 8990 words respectively (see Fig. 1). The ratio between the number of sentences in summaries and full documents is 0.16.

4 Experiments

The aim of the experiments described in this section is to compare different approaches for text summarization of legislation documents. For this purpose we used the dataset mentioned in the previous section.

4.1 Experiments Design

Different algorithms were experimented on the same dataset. They contain both extractive and abstractive approaches. Some of them do not require training, while others are trained from scratch or fine-tuned on the dataset.

T5 and PreSumm (which is based on BERT) have restrictions for the number of tokens in the input and in the output. In order to be able to handle the data for these algorithms, the full texts and summaries were splitted into chunks: full texts containing 1024 tokens and summaries - 128 tokens. We used the same ROUGE metric to evaluate each summary chunk against each full text chunk. During training each full text chunk is paired with the most applicable summary chunk. During evaluation we generated summaries for all chunks from the full text. They were concatenated and the result

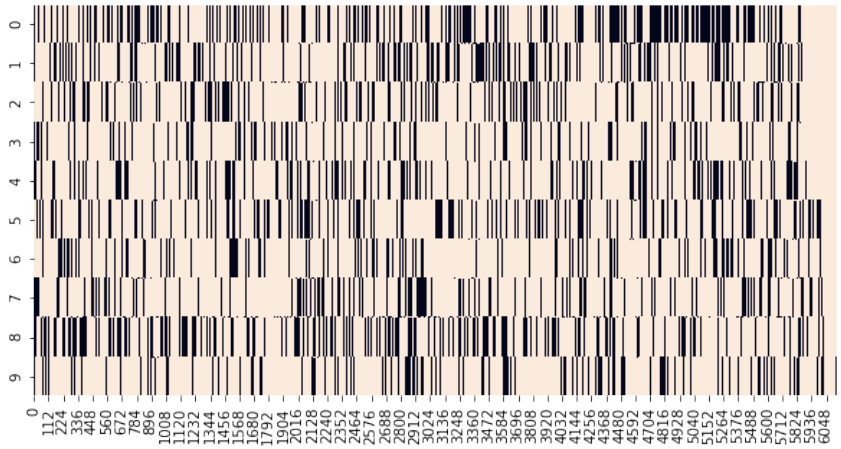

Figure 2: Heatmap of words being selected for extractive summary from example texts from the dataset. The extractive summarizer with BERT and K-Means chooses sentences from the whole document and does not focus on specific parts of it.

was evaluated against the original summary.

4.1.1 Extractive Summarization Based on Weighted Frequency Tokens of Sentences (Baseline Algorithm)

The first approach that was used is basic extractive summarization (Malik, 2019). 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. 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 in order to avoid the ones that do not contain enough information by themselves. Finally, the sentences are ordered in descending order by the sum of the weighted frequencies. The summary contains the sentences in 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.

4.1.2 Fine-tuned PreSumm Encoder

PreSumm is a pre-trained encoder based on BERT for the purpose of text summarization (Liu and Lapata, 2019). This method is entirely based on BERT and provides two implementations: BERTSUMEXT for extractive summarization and BERTSUMABS for abstractive summarization.

For both extractive and abstractive settings, the algorithm generates a summary consisting of the sentences which maximize the ROUGE-2 score against the gold summary during training. When generating summaries for a new document, the model is first used to obtain the score for each sentence. These sentences are ranked by their scores from highest to lowest. During sentence selection, Trigram Blocking is used to reduce redundancy (Paulus et al., 2017). Given a summary and candidate sentence, the sentence is skipped if there exists a trigram overlapping between it and the summary. The aim is to minimize the similarity between the sentence being considered and sentences which have been already selected as part of the summary.

4.1.3 Extractive Text Summarization with BERT and K-Means

We used the solution proposed by Miller (2019). It works 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. Fig. 2 displays a heatmap of the full text and the words that were selected to be part of the summary.

Model	Type	Metric	Precision	Recall	F1 score
Extractive summarization based on weighted frequency tokens of sentences (baseline)	extractive	Rouge 1	19.22	73.14	26.52
		Rouge 2	7.41	29.94	10.41
		Rouge 3	3.46	13.30	4.83
Summarization with BERT and K-Means	extractive	Rouge 1	35.85	54.16	36.82
		Rouge 2	11.57	17.14	11.65
		Rouge 3	4.64	6.23	4.48
Summarization with LEGAL-BERT and K-Means	extractive	Rouge 1	34.06	56.45	36.06
		Rouge 2	11.25	18.48	11.72
		Rouge 3	4.63	6.95	4.63
PreSumm	extractive	Rouge 1	22.64	71.80	29.25
		Rouge 2	8.19	28.20	10.85
		Rouge 3	3.47	11.52	4.57
PreSumm	abstractive	Rouge 1	33.30	25.09	28.46
		Rouge 2	5.41	4.08	4.63
		Rouge 3	1.29	0.99	1.11
T5	abstractive	Rouge 1	42.89	52.25	**39.27**
		Rouge 2	15.94	18.97	**14.17**
		Rouge 3	7.28	8.07	**6.28**

Table 1: Results for experimented models. Fine-tuned T5 model generated the best results. Baseline F1 score was improved by all other algorithms.

4.1.4 Data-specific Approach for Legal Documents

The Extractive Text Summarization with BERT and K-Means allows its encoding model to be replaced and we experimented by changing it to LEGAL-BERT. (Chalkidis et al., 2020). LEGAL-BERT is a model which is based on BERT and is trained on twelve GB of diverse English legal texts from several fields. This experiment was encouraged by the specific vocabulary which the legislation documents consist of.

4.1.5 T5 Model

T5 (Raffel et al., 2019) is an encoder-decoder model and converts all NLP problems into a text-to-text format. It is trained using teacher forcing. This means that for training it always needs an input sequence and a target sequence. It is pre-trained on an open-source pre-training dataset, called the Colossal Clean Crawled Corpus (C4). The T5 model, pre-trained on C4, achieves state-of-the-art results on many NLP tasks while being flexible enough to be fine-tuned to a variety of important downstream tasks.

4.2 Evaluation Metrics

The most widely used metric for 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 reference or a set of references (human produced) summary. ROUGE-N refers to the overlap of n-gram between the system and reference summaries. In particular ROUGE-1, ROUGE-2 and ROUGE-3 were used in the conducted experiments.

4.3 Results

Table 1 shows the results from the experiments. All extractive approaches outperform the baseline algorithm. PreSumm only improved Rouge 1 score from 26.52 (baseline) to 29.25. Extractive Text Summarization with BERT and K-Means yielded the best extractive results - 36.82 Rouge 1 score. When we tried to use the same algorithm but replaced BERT with pre-trained LEGAL-BERT which is fine-tuned on legal texts the results were slightly worse - 36.06 Rouge 1 score. Having in mind that the original summaries are generated in an abstractive way by experts the score of 36.82 can be considered a big success.

Both abstractive algorithms (PreSumm and T5) have limitations on input and output size. Both full texts and summaries were splitted to chunks of sizes 1024 and 128 respectively. During the

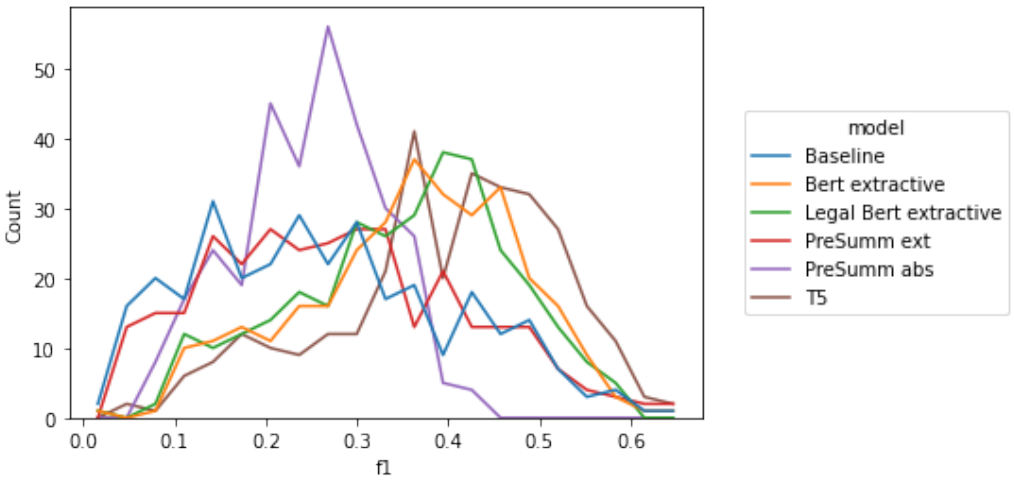

Figure 3: The lines represent Rouge 1 F1 scores distribution for the compared models. The horizontal axis represents the F1 score. The vertical axis represents the number of documents which have achieved this score.

training phase each full chunk was paired to the best chunk from the summary for which it had the highest rouge-1 f1 score. During evaluation the generated summaries from all chunks of the full text were concatenated and compared to the original summary. Both approaches yielded better results than the baseline.

Fine-tuned T5-base abstractive model with overridden implementation for handling the long texts by splitting them to chunks showed the best overall results. Here is first paragraph the summary with the highest score (0.65 Rouge-1 F1 score):

Original:

The European order for payment (EOP) procedure applies to all civil and commercial matters in cases where at least one of the parties lives in an EU country different from the one where the application for an order is made. The procedure does not apply to certain issues:revenue, customs or administrative matters,state liability for acts and omissions in the exercise of state authority,matrimonial property regimes,bankruptcy, proceedings relating to the winding-up of insolvent companies or other legal persons, and judicial arrangements,social security,claims arising from non-contractual obligations, unless there was an agreement between the parties or an admission of debt or they relate to liquidated debts arising from joint ownership of property.

Generated:

a European order for payment procedure is established in the EU country where the claimant lives. Its purpose is to ensure that creditors and debtors have equal access to justice throughout the EU. The regulation also establishes an electronic system for determining which courts have jurisdiction to issue an order for payment, as well as a mechanism for the recovery of uncontested pecuniary claims.

Fig. 3 shows the distribution of Rouge 1 F1 scores for all experimented models. The PreSumm abstractive model curve is most similar to normal distribution. We can also observe similar behaviour between baseline extractive curve and PreSumm extractive. The rest have similar shapes and the fine-tuned T5 model achieves the best overall results.

5 Conclusion

We introduced a new dataset about summarization of European legislation documents. We also presented a comparative study of various algorithms for automatic text summarization on this dataset. In our experiments on these tasks, we obtained promising results including huge improvements over baseline. We believe that the new dataset, adopting existing algorithms to domain specific data and the results described in this paper will accelerate research directions on text summarization to expand the variety of domains with domain specific information and different sizes.

Acknowledgments

This research is partially supported by Project UNITe BG05M2OP001-1.001-0004 funded by the OP ”Science and Education for Smart Growth” and

co-funded by the EU through the ESI Funds.

References

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.

Franck Dernoncourt, Mohammad Ghassemi, and Walter Chang. 2018. A repository of corpora for summarization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

David Graff and Christopher Cieri. 2003. English Gigaword.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. *arXiv e-prints*, page arXiv:1506.03340.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A Large Scale Chinese Short Text Summarization Dataset. *arXiv e-prints*, page arXiv:1506.05865.

Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. *arXiv e-prints*, page arXiv:1908.08345.

Usman Malik. 2019. Text summarization with nltk in python. https://stackabuse.com/text-summarization-with-nltk-in-python/. Accessed: 2020-05-02.

Derek Miller. 2019. Leveraging BERT for Extractive Text Summarization on Lectures. *arXiv e-prints*, page arXiv:1906.04165.

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.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *arXiv e-prints*, page arXiv:1705.04304.

Publications Office of the European Union. 2020. Summaries of eu legislation. Data retrieved from EUR-Lex, https://eur-lex.europa.eu/browse/summaries.html?locale=en.

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 Text-to-Text Transformer. *arXiv e-prints*, page arXiv:1910.10683.

Sciforce. 2019. Towards automatic text summarization: Extractive methods. [Online; posted 23-January-2019].

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.

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.

Wen Xiao and Giuseppe Carenini. 2019. Extractive Summarization of Long Documents by Combining Global and Local Context. *arXiv e-prints*, page arXiv:1909.08089.

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.