A Comparative Study of Extractive and Abstractive Approaches for Automatic Text Summarization on Scientific Texts

Todor Tsonkov, Gergana Lazarova, Valentin Zmiycharov, Ivan Koychev

Faculty of Mathematics and Informatics, Sofia University "St Kliment Ohridski", Sofia, Bulgaria

Abstract
Automatic summarization of long documents is a challenging task and it is not well studied. The existing text summarization approaches are developed and tested mainly on relatively short documents such as news, web pages etc. In this paper, we aim to study the performance of some of the existing state of the art text summarization algorithms on scientific papers, which are relatively long documents. For the conducted experiments we used the Yale Scientific Article Summarization Dataset. Summarizing scientific texts is a challenge itself due to the many different topics that have uncommon words, long and complex sentences and are hard to understand even by humans. The dataset consists of 1000 scientific papers with both human-generated summaries and the original abstracts. We have used both abstractive and extractive text-summarization algorithms. We propose a chunk-based approach for the abstractive algorithms (Google Pegasus and T5). The ROUGE score is used to evaluate and compare the results.

Keywords
natural language processing, deep learning, text summarization

1 Introduction

Automatic summarization of a long document is a useful, but rather challenging problem. In this paper, we aimed to study this problem using a dataset of scientific papers. Summarization of scientific texts is an even more challenging task because each scientific topic has its domain knowledge, many specific words and phrases, which makes general language models not directly applicable.

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. where the expected summary is a few sentences long. However, there are cases in which both the text and the summary are sufficiently long. For example, a good summary of a book might be several pages. Dernoncourt et al. 2018 [6] provide a comprehensive overview of the current datasets for summarization. Noticeably, most of the larger scale summarization datasets consist of relatively short documents.

There are two main approaches for text summarization: extractive and abstractive. The extractive summarization methods aim to identify important sections of the text and use them verbatim to produce a subset of the sentences from the original text. On another side, the abstractive summarization methods aim to create algorithms that are capable of "understanding" the whole text and generating a new shorter text that conveys the most important information from the original one.

This study aims to evaluate the applicability of different algorithms for text summarization on scientific documents. We report results from experiments using state of the art algorithms from both approaches evaluating them on a dataset of scientific papers. An important step of the study is the text preprocessing of the source papers to make them suitable for training the summarization models.

Education and Research in the Information Society, September 27–28, 2021, Plovdiv, Bulgaria
EMAIL: ttsonkov@gmail.com, gerganal@fmi.uni-sofia.bg, vzmijcharo@uni-sofia.bg, koychev@fmi.uni-sofia.bg

The main contributions in this paper are:

- We have investigated existing approaches in text summarization and have selected the most suitable approaches for scientific texts.
- We have run experiments with different algorithms against a dataset containing scientific documents.

The paper is organized as follows – we provide an introduction part that shows the problem we are trying to solve, then we investigate the existing approaches and describe in detail the selected algorithms that we use to evaluate the scientific dataset. We have provided the results and made a conclusion based on them.

2 Related Works

Different methods for generating summaries of long texts have been proposed. The approach of Xiao and Carenini (2019) [1] proposes a novel neural single document extractive summarization model for long documents, incorporating both the global context of the whole document and the local context within the current topic. They evaluate the model on two datasets of scientific papers on extracting sentences from a given document (without dealing with redundancy) that contain information, 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 legislation documents.

Nakao [11] presents an algorithm for text summarization using the thematic hierarchy of a text. The proposed algorithm is intended to generate a one-page summary. The algorithm consists of 3 stages. 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. A possible improvement of this algorithm can be to be adapted to summaries with dynamic length.

Another approach that was tried was to combine extractive and abstractive models (Wang et al., 2017) [3]. It consists of two phases: extractive and abstractive. 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 encoder-decoder architecture and devises pointer and attention mechanisms to generate summaries.

Vaswani [4] presented an architecture called “Transformer”, which is compared to the RNNs and Convolutional Neural Networks (CNN). 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 doesn’t need to be processed in order (for instance the beginning of the text doesn’t 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 will decode the vector and produce the summary. The datasets used for training can be bigger and thus exist pre-trained models 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. (Vaswani, et al., 2017).

Zhao et al [5] have evaluated their best PEGASUS model on 12 downstream summarization tasks spanning news, science, stories, instructions, emails, patents, and legislative bills. Experiments demonstrate it achieves state-of-the-art performance on all 12 downstream datasets measured by ROUGE scores. Their model also shows surprising performance on low-resource summarization, surpassing previous state-of-the-art results on 6 datasets with only 1000 examples. Also, they have validated their results using human evaluation and show that their model summaries achieve human performance.

Although a lot of research has been done in the field, most of the proposed models have limitations concerning the length of the documents. When the size of the document is large, the number of the parameters of the model grows significantly and the computations take a lot of time. [14] use a chunk-based model, dividing the documents into parts and comparing them to state-of-the-art approaches.

3 Experiments Design

The experiments presented in this paper aim to compare different text summarization algorithms on scientific papers. The main challenge of this experiment appears to be that some of the used text summarization algorithms have a limited length of the input text.

3.1 Dataset

For the experiment, we are using the Yale Scientific Article Summarization Dataset[1]. The dataset consists of 1000 scientific papers from the area of Computational Linguistics, each of which has both the original abstracts and authors' generated summaries, which has a big overlap with the paper abstract as we observed.

3.2 Prepossessing

Each paper is preprocessed by removing the ABSTRACT part and a text-summarization algorithm is applied. For evaluation, the ROUGE metric is used. Both the paper abstracts and the human-created summaries are used as gold summaries.

As a preprocessing, we have parsed the texts by chunks sized 1024 to train the machine learning algorithms. This is the maximum length that can be used with the tools we work. In further experiments, we can try different heuristics like splitting the texts by paragraphs or by chunks of a few sentences and checking the predictive accuracy of each method.

3.3 Evaluation Metrics

For the evaluation metric, we used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics. The ROUGE-n measures the overlap of n-grams between the automatically generated summaries and the reference summaries (human-generated). For example, ROUGE-1 refers to the overlap of unigram (single word) between the system and reference summaries.

In our case, both the paper abstracts and the human-created summaries are used as gold summaries.

3.4 Compared Summarization Algorithms

3.4.1 Google Pegasus [5]

One of the trendiest state-of-the-art algorithms that try to solve the problem of abstractive text summarization is an NLP deep learning model called PEGASUS. It can be used for abstractive summarization and the abstractive approach is more challenging because the text is long. The model needs to be able to "understand" the text and it should be able to generate new text that briefly represents the content.

3.4.2 Extractive Text Summarization based on Bert and K-means

After sentence tokenization, the Bert model outputs the embeddings. Then, clusterization is applied. The authors experimented with both K-means and Gaussian Mixture Models and due to the very similar performance, K-means was selected as a clustering algorithm (Bert-K-means) [12]. The embedded sentences, which are closest to the centroids, are selected to take part in the predicted summary. There is no restriction on the length of the texts.

3.4.3 T5

T5 is an encoder-decoder model [13]. It converts all NLP problems into a text-to-text format and is trained using teacher forcing. For training, it always needs an input sequence and a target sequence. It is pre-trained on an open-source dataset Colossal Clean Crawled Corpus (C4). The T5 model, pre-

[1] https://cs.stanford.edu/~myasu/projects/scisumm_net/

trained on C4, achieves state-of-the-art results on many NLP tasks and can also be fine-tuned to a variety of important downstream tasks.

The main difference between T5 and Bert is that T5 always takes strings as an input/output while BERT models only take as an input the class models.

4 Results from the experiments

4.1 Extractive versus Abstractive Algorithms

Table 1: Comparison of the extractive and abstractive algorithms.
The abstracts of the papers are used as gold summaries

Summarizer	Metric	F1	Precision	Recall
Bert-K-means	rouge-1	0.**2852**	0.2411	0.4764
Bert-K-means	rouge-2	0.0746	0.0713	0.1266
Bert-K-means	rouge-3	0.0338	0.0377	0.0580
Pegasus	rouge-1	0.2255	0.2546	0.2567
Pegasus	rouge-2	0.1463	0.1926	0.2089
Pegasus	rouge-3	0.**0660**	0.0734	0.0812
T5	rouge-1	0.2649	0.4951	0.1853
T5	rouge-2	0.**1020**	0.1945	0.0709
T5	rouge-3	0.0622	0.1200	0.0431

Table 2: Rouge scores of the best performing Extractive algorithm **Bert-K-means** on the authors' summaries

Metric	F1	Precision	Recall
rouge-1	0.3443	0.2954	0.4625
rouge-2	0.0843	0.0719	0.1167
rouge-3	0.0354	0.0303	0.0507

In **Table 1** the three algorithms: the extractive Bert-K-means, the abstractive Pegasus and T5 are compared based on the rouge-1, rouge-2 and rouge-3 scores. The extractive algorithm outperforms the abstractive ones because the implementations of the abstractive still cannot cope with long texts and need text partition into chunks. The number of the parameters of the algorithms get high and there are limitations in the very implementations. T5's maximum input text length is 512 and Google Pegasus's is 1024.

For the best performing extractive Bert-K-means algorithm, further experiments were held based on the Authors' summaries. The results can be seen in **Table 2.**

The extractive approach (Bert-K-means) achieves very good results – having an F1 rouge-1 score of 0.3443 on the author summaries and 0.2852 on the abstracts. Being an extractive approach, it selects a subset of the original sentences in the document. 5% of the sentences are selected.

4.2 Naive Approach

A simple but very successful approach is also proposed and evaluated. Due to the structure of the document, intuitively approaching the problem, the original idea was that most of the time, paper authors summarize their scientific work also in the conclusion part of the publication. We decided to evaluate the similarity between the abstracts and the conclusions. Of the 1000 documents, 651 were selected containing one of the following words in the tags: "Conclusion", "Concluding Remarks", "Summary", "Final remarks". Based on the ROUGE metric:

- The original abstracts are compared to the Conclusions
- The human-generated summaries are compared to the Conclusions.

Table 3: Naive Approach, the conclusion part of the paper is compared to the abstract of the document /authors' summary

Gold Summaries	Metric	F1	Precision	Recall
Authors' Summaries	rouge-1	**0.4182**	0.4265	0.4917
Authors' Summaries	rouge-2	**0.1403**	0.1473	0.1613
Authors' Summaries	rouge-3	**0.0713**	0.0772	0.0790
Abstract	rouge-1	0.3784	0.3590	0.5389
Abstract	rouge-2	0.1386	0.1476	0.1888
Abstract	rouge-3	0.0764	0.0924	0.0971

Generally, the author summaries, which are manually generated, summarize better the papers than the abstracts. The Conclusions are also more similar to the author summaries than to the very abstracts. It can be seen in Table 3 that the naïve approach – simply taking the conclusions as summaries, brings the best results (F1 rouge-1 score: 0.4185).

Comparing the abstractive approaches, it can be seen that T5 (0.2649) outperforms Google Pegasus (0.2255) but still is inferior to the extractive approach – considering the rouge-1 score. Still, the abstractive algorithms have higher rouge-2 and rouge-3 scores compared to extractive Bert-K-means rouge-2 and rouge-3 scores. The naive approach outperforms all the algorithms based on all rouge scores.

In the three tables, detailed information about the Precision and Recall scores is provided. The naive approach has higher precision (rouge-1:0.4265 on the Authors' Summaries and rouge-1:0.3784 on the abstracts of the scientific papers) and recall (rouge-1:0.3590 on the Authors' Summaries and rouge-1:0.5389 on the abstracts of the scientific papers) than the machine learning algorithms. Pegasus's Precision is very close to its Recall, whereas the extractive Bert-K-means Recall (rouge-1 Recall: 0.48) surpasses its Precision (rouge-1 Precision: 0.24) which means that most of the words in the system/gold summaries are present in the generated summaries whereas the opposite is not true.

5 Conclusion

This paper focuses on scientific paper summarization. An already published dataset with papers and hand-made summaries is used for multiple algorithm evaluation. It has experimented with both modern extractive and abstractive approaches. A naïve approach is also proposed, based on the intuitive idea that the conclusion part of the paper is very similar to the abstract.

The results show that the naïve approach leads to best rouge-1, rouge-2 and rouge-3 scores. The extractive approach outperforms the abstractive ones based on the rouge-1 score but the rouge-2 and rouge-3 scores are not so good.

For future work, it should be explored which parts of the papers should be summarized. The conclusion should take part in the selected paragraphs. A subset of the paragraphs will lead to shorter texts and will solve the chunk division problem of the texts.

For future work, human-expert summary evaluation can also be performed but it requires lots of people involved in the process.

Acknowledgements

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

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

[2] G.Izacard and E. Grave. Leveraging passage retrieval with generative models for open-domain question answering. arXiv preprint arXiv:2007.01282, 2020.

[3] S. Wang, X. Zhao, B. Li, B. Ge, and D. Tang. 2017. Integrating extractive and abstractive models for long text summarization. In IEEE International Congress on Big Data (BigData Congress), 2017, pages 305–312.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems. 2017, pages 6000–6010.

[5] Jingqing Zhang, Yao Zhao, Mohammad Saleh, Peter J. Liu. "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization." Thirty-seventh International Conference on Machine Learning (ICML). 2020.

[6] 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). 2018.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv e-prints, page arXiv:1810.04805, 2018.

[8] Lei Li, Wei Liu, Marina Litvak, Natalia Vanetik, and Zuying Huang. 2019. In Conclusion Not Repetition: Comprehensive Abstractive Summarization with Diversified Attention Based on Determinantal Point Processes. arXiv e-prints, page arXiv:1909.10852. 2019.

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

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

[11] 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. 2000.

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

[13] 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. 2019.

[14] Valentin Zmiycharov, Milen Chechev, Gergana Lazarova, Todor Tsonkov and Ivan Koychev. A Comparative Study on Abstractive and Extractive Approaches in Summarization of European Legislation Documents. Proceedings of the International Conference on Recent Advances in Natural Language Processing. 2021, pp. 1649-1655