Impact of Text Extraction on Abstractive Scientific Paper Summarisation

Laura Celina Stahlhut[*], **Helen Schaller**[*], **Deborah Noemie Jakobi**[*]
Department of Computational Linguistics
University of Zurich
{lauracelina.stahlhut, helen.schaller, deborahnoemie.jakobi}@uzh.ch

Abstract

While machine learning has improved the results of NLP tasks across the board, researchers are often confronted with the issue that neural models cannot process lengthy input texts in their entirety. Having the constraint of a maximum input length of a model usually leads to the truncation of the input text. We propose an alternative method for shortening texts which is based on ROUGE scores. This paper shows that in the case of abstractive summarisation of scientific papers, our proposed method is not superior to truncation which is likely due to the structure of the data. However, we find that abstractive summarisation works better with individually extracted sentences rather than paragraphs which contradicts the truncation method and opens a space for further research into alternative extractive methods as a preprocessing step for neural models.

1 Introduction

Pretrained transformer-based models are commonly used for a wide range of generative NLP tasks such as text summarisation. The main aim of summarisation is to reduce a text to a shorter version while keeping most of its original meaning (Liu and Lapata, 2019). A distinction between extractive and abstractive summarisation can be made where the first refers to a way of shortening a document by copying its most characteristic sentences to form a summary whereas the second method results in summaries consisting of phrases that may not have been part of the original text (Collins et al., 2017). Abstractive summarisation is arguably the more complex task since being able to formulate abstract summaries entails both long-term reading comprehension and text generation ability while extractive models are based on simpler binary classification where they are tasked to predict whether a sentence is part of a summary. Others have used transformer-based models such as BERT to generate abstractive summaries of shorter texts with less than 1,000 words (e.g. Liu and Lapata (2019)). However, summarising longer and more complex texts such as scientific papers is not a straightforward task with conventional transformer-based models since they can only take an input text up to a certain length because of their full attention mechanism. This problem has been alleviated by the invention of BIGBIRD which employs a sparse attention mechanism that approximates the performance of BERT's full attention (Zaheer et al., 2021). As a result, the model can handle inputs with up to 4096 tokens[1]. However, texts such as scientific papers are still longer than this.

We will compare five different extractive methods that are applied prior to abstractive summarisation and analyse how these methods impact the quality of the final summaries.

2 Related Work

Different ways of leveraging both extractive and abstractive summarisation models at the same time have been tested on long documents. Chen and Bansal (2018) have proposed a model consisting of 1) an extractor agent which first applies neural encoding to the input sentences which may then be extracted by a pointer network, and 2) an abstractor network that rewrites the extracted sentences to a concise summary. An advantage of their extractive pointer network is that it extracts sentences that have different key points, so repetition will be avoided in the output summary. Wang et al. (2017) use a graph model for hierarchically extracting sentences from each paragraph and an RNN encoder-decoder that generates an abstractive summary. The combination of an extractive with an abstractive model outperforms abstractive models

[*] Equal contribution.

[1] Note that BERT applies subword tokenisation which means that #tokens $\geq$ #words if words are considered to be text units separated by white space.

CSPubSum dataset (Collins et al., 2017)	
# docs (train/val/test)	8104 / 2044 / 150
avg. doc length (words/sentences)	7394 / 272
avg. summary length (words/sentences)	56 / 4

Table 1: Information about the dataset used.

without a special extraction step. Liu et al. (2018) tackle the problem of multi-document summarisation with a hybrid extractive-abstractive model as well. They extract the most important paragraphs from the documents to reduce the size of the input to the abstractive transformer-based decoder. The aim of their work is similar to ours since they compare the impact of five different extraction methods on the generated summaries. They found that the extraction method "appears to have a significant effect on final performance" (Liu et al., 2018, 11). To our knowledge, combining an extractive and an abstractive step has not been used to summarise scientific papers. Collins et al. (2017) used neural sentence encoding and manually-crafted features for extractive summaristion of the CSPubSum dataset (cf. section 3). Cagliero and La Quatra (2020) who worked with the same dataset used a regression-based approach to add a ranking to the extracted sentences. The developers of the BIGBIRD (Zaheer et al., 2021) and the PEGASUS (Zhang et al., 2019a) models performed abstractive summarisation on the scientific_papers dataset (cf. section 3). Since many of the texts were too long to input to the models, they truncated the texts after 3072 tokens (Zaheer et al., 2021, 37).[2]

3 Dataset

We chose the dataset CSPubSum (cf. table 1) from Collins et al. (2017). It consists of more than 10,000 scientific papers from computer science obtained from ScienceDirect, a database for scientific publications. The key characteristic of this dataset is that all papers are guaranteed to have author-written highlight statements which summarise the main findings of each contribution and serve as gold standard for each paper's summary.

The data was preprocessed[3] in the same way as the data used to pretrain the BIGBIRDPEGASUS model, i.e. the scientific_papers dataset produced by Cohan et al. (2018). This dataset consists of long scientific papers acquired from arXiv. The difference between CSPubSum and scientific_papers is that the latter uses the abstract as ground truth summary while we use the author-written highlight statements.

4 Experiments

We will apply two different extractive steps and use their outputs as the input for an abstractive model. Comparing the output summaries of the abstractive model will enable us to judge which extractive models may be more suitable in this setup.

4.1 Extractive Model

For each extractive method, we create a version that extracts sentences from the input article and one that extracts paragraphs both up to the input length of 3072 tokens. We delimit paragraphs as text sections located between section headers.[4] In addition, we will test how truncating the documents prior to abstractive summarisation performs.

We expect a difference in summarisation quality when using either extracted paragraphs or sentences. We assume that the summaries will be worse if the abstractive model's input consists of incoherent sentences instead of paragraphs that preserve long-term dependencies.

PEGASUS-inspired extractive step (PEGex): Our first extractive model is inspired by the gap sentence generation (GSG) pretraining objective of the PEGASUS transformer encoder-decoder where important sentences are masked from the input and the model is trained to generate these masked sentences from the rest of the input (Zhang et al., 2019a).[5] The assumedly most important sentences were chosen to be masked. The authors found that the ROUGE-1-F1 between a sentence and the rest of the text was a good proxy for a sentence's importance. We selected a) the most important sentences from the whole paper and b) the most important paragraphs from the whole paper. For selecting paragraphs, we calculated the ROUGE-1-F1 score

[2] Although the maximum input length is 4096 tokens.

[3] We removed figures and tables, normalised formulas and citations with special tokens and removed reference sections.

[4] This definition of a paragraph entails texts of various length since some authors are accustomed to making longer sections than others.

[5] The authors argue that this pretraining task resembles the downstream task of abstractive summarisation more closely than other pretraining tasks which supposedly makes fine-tuning this model better and faster (Zhang et al., 2019a).

between each sentence of a paragraph and the rest of the paper (without the paragraph) and considered the average across all sentences of a section to get a final value for a paragraph's importance.

Random extractive step (RANex): We shorten the input by picking random sentences and paragraphs from the document. The rationale behind this is that important information is not just located at the beginning of a paper but might also occur rather towards its end. We expect this version to perform worse than extraction with PEGex.

Truncating extractive step: We truncate the texts after 3072 tokens, as was originally done in the paper describing the BIGBIRD model (Zaheer et al., 2021). For the truncated version there exists no difference in paragraph or sentence extraction.

4.2 Abstractive Model

The abstractive model we use is the transformer model BIGBIRDPEGASUS[6] which is trained on abstractively summarising texts from the dataset `scientific_papers` (cf. section 3) (Zaheer et al., 2021). It employs BIGBIRD's (Zaheer et al., 2021) sparse attention mechanism and was pretrained with the gap sentence generation objective typical for PEGASUS (Zhang et al., 2019a). For each extractive method, a separate abstractive model is finetuned. This ensures that the abstractive model will not be biased towards one particular extraction method and that all measured differences in summarisation quality will be due to the extraction method. To generate the output summaries we used a sampling strategy instead of greedy decoding. We limited the number of words to consider for sampling and added a repetition penalty (Keskar et al., 2019) to encourage more diverse output. The hyperparameters we used are listed in the appendix.

4.3 Overview of Experimental Setups

All experiments consist of an extractive followed by an abstractive summarisation step. The extractive model varies from setup to setup and serves the purpose to shorten the scientific papers to maximally 3072 tokens. The abstractive model is always BIGBIRDPEGASUS finetuned on the extracted text (individually for each extraction method). These are the extractive models we compare: sentence extraction (PEGex or RANex) and paragraph extraction (PEGex or RANex) as well as truncating the texts after 3072 tokens.

5 Evaluation

Following current practice, we report the results of our experiments using ROUGE scores. Rouge-N measures the N-gram recall between a generated summary and a reference summary while ROUGE-L measures the longest common subsequence of candidate and reference (Lin, 2004).

BERTScore[7] includes contextual embeddings (Zhang et al., 2019b) as created by BERT (Devlin et al., 2019) or ELMo (Peters et al., 2018) to measure similarity between reference and prediction instead of n-gram matches. The advantage is that paraphrases can be matched and that the score correlates more with human evaluation. The score is calculated by weighting the aggregation of cosine similarities between reference and prediction tokens. As evaluation of text summation is notoriously difficult, it would have been beneficial to let humans evaluate the summaries, but this is out of scope for this work.

6 Results and Discussion

Table 2 reports the ROUGE-L score for the best system of Collins et al. (2017) which is notably higher than the ROUGE-L scores achieved by our systems. This was to be expected since ROUGE scores, which measure lexical overlap, tend to favour extractive summarisation methods (Pilault et al., 2020). Also, our primary aim is not to compare our systems to other setups but to compare the different extraction methods used within our setup.

Our experimental results in Table 2 show that PEGex generally outperformed RANex. However, the abstractive model finetuned on truncated text still outperformed PEGex in all metrics.

The performance of the abstractive model was generally better when it was finetuned on individual sentences in the case of PEGex, although usually only by a small margin. For RANex, it seems to be the other way around, i.e. that paragraph selection is preferable. However, the differences are minuscule. What we have not considered so far is that the paragraphs might be unreasonably heterogeneous w.r.t. their lengths, meaning that some authors prefer to write much longer paragraphs than others. As a result, it is possible that the extraction methods including paragraph extraction

[6] https://huggingface.co/google/bigbird-pegasus-large-arxiv

[7] https://github.com/Tiiiger/bert_score

might have resulted in texts that are much shorter than 3072 tokens. Working with a different kind of coherent, but shorter text units consisting of several sentences could improve text extraction.

Collins et al. (2017, 200) found that some of the gold summary sentences were copy-pasted from the documents: Many more were copy-pasted from the abstract and the introduction compared to the other paragraphs, and the abstract had the highest ROUGE-L score compared to the reference summary. For both PEGex-sent and -par, text was mostly extracted from the abstract and introduction.[8] This correspondence with the findings of Collins et al. (2017) makes us believe that PEGex does a sensible job of extracting sentences that are relevant for generating summaries. However, this might also be an explanation for the strength of simple truncation since truncated papers likely contain all of the abstract and the introduction while PEGex might leave some parts out.

We think that the main weakness of our extraction method is that it extracts sentences / paragraphs covering the same information multiple times. It is a common trait of scientific papers that some of the information is repeated in several paragraphs. With PEGex, which selects sentences or paragraphs that are most similar to the rest of the text, it is likely that the extracted text units have overlapping contents which would not be beneficial to the summary quality. Improving on this point could be crucial for beating the baseline of truncation since truncated text is also likely to contain the same information multiple times.

Manual experiments on the validation set showed a decrease in repetitiveness and a general improvement of the text quality in the generated summaries when using sampling instead of greedy decoding, but the ROUGE scores decreased. This phenomenon has been observed before (Schluter, 2017) and raises concerns about the adequacy of using ROUGE to evaluate summarisation quality, also since ROUGE's capability of measuring the information overlap between two texts has shown to be limited (Deutsch and Roth, 2021). Due to the lack of a reliable automatic evaluation metric that correlates perfectly with human judgements, human evaluation currently seems to be the best way to score summaries. Such a study using our systems is still open for future work.

The BERTScore is difficult to interpret as none of the previous papers working with `CSPubSum` have worked with this metric. In other evaluations of abstractive scientific paper (Gabriel et al., 2019) or news text summarisation (Li et al., 2019), the BERTscore was lower (i.e. not higher than 68.00). This should be interpreted with care as the studies are not directly comparable to ours. Still, the large gap suggests that there is some amount of semantic overlap between our models' predictions and the reference summaries.

Extractive model		**ROUGE**			**BERT Score**
		R1	R2	RL	F1
Collins et al. (2017)		-	-	~32.25	-
PEGex	par	29.77	6.94	19.68	84.85
	sent	*30.36*	*7.57*	*20.29*	*85.10*
RANex	par	27.44	5.34	18.30	84.62
	sent	27.44	5.20	18.31	84.56
Truncation		**31.87**	**8.55**	**20.98**	**85.38**

Table 2: Results for all models. The best of our results is in bold, the second best in italics.

7 Conclusion

We evaluated a range of extractive steps to shorten scientific papers prior to feeding them to an abstractive summarisation model. Our ROUGE-based extractive step did not improve the results compared to truncating the texts. However, we observed that finetuning the abstractive model on individual sentences may yield better results in some scenarios than using coherent text in the form of paragraphs. This is an interesting observation that calls for further experiments. For future work, expanding our proposed method of shortening texts to different datasets such as news articles would be interesting since their structure is different from the structure of scientific papers. Studying the influence of different sampling parameters on the output summary and how this is reflected in different metrics might also result in useful insights. And finally, since automatic evaluation using metrics such as ROUGE has a limited expressive power, doing a human evaluation study to compare the generated summaries of the different systems would be insightful as well.

[8] Paragraphs from which sentences are selected most frequently (in this order): *Introduction, Abstract, Discussion, Conclusion, Results*. Paragraphs selected most frequently for paragraph selection (in this order): *Abstract, Introduction, Conclusion, Discussion, Results.*

References

Luca Cagliero and Moreno La Quatra. 2020. Extracting highlights of scientific articles: A supervised summarization approach. *Expert Systems with Applications*, 160:113659.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *CoRR*, abs/1805.11080.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada. Association for Computational Linguistics.

Daniel Deutsch and Dan Roth. 2021. Understanding the extent to which content quality metrics measure the information quality of summaries. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 300–309, Online. Association for Computational Linguistics.

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

Saadia Gabriel, Antoine Bosselut, Jeff Da, Ari Holtzman, Jan Buys, Kyle Lo, Asli Celikyilmaz, and Yejin Choi. 2019. Discourse understanding and factual consistency in abstractive summarization.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization.

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

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *CoRR*, abs/1801.10198.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. 2017. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 305–312.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert.

A Appendix

- We would include a graph of the number of epochs we chose to train the models. We plotted the losses etc. to find a good training paradigm.
- Summary of manual experiments with Hugging Face's `generate` function, more on what each hyperparameter does.
- Add a few example predictions to show that the output created with the `generate` function and the special hyperparameters improves the texts.
- Cheng and Lapata (2016) made a human evaluation study for six different summarisation models where they asked study participants to rank the outputs of the models. To do so, the participants had access to the original document and the different summaries. We think this would have given much more insight (compared to ROUGE scores) on whether the results are really better when using one extractive method compared to another.
- Hyperparameters: `do_sample=True` in the `generate` method. This hyperparameter allows for sampling from the vocabulary instead of greedy decoding. Other hyperparameters we set are: `repetition_penalty=1.3`, `top_k=100`, `top_p=0.95`, `temperature=0.95`, `learning_rate=2e-7`.
- We decided not to report all scores in the paper (just the F1 scores, not recall and precision). For the sake of completeness, we would just report them here to allow the scores to be compared to other models.