The 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)
Taipei, Taiwan, November 21-22, 2022. The Association for Computational Linguistics and Chinese Language Processing

Automatic Generation of Abstracts for Research Papers

Dushan Kumarasinghe
Department of Computer Science
and Engineering
University of Moratuwa
dushan.21@cse.mrt.ac.lk

Nisansa de Silva
Department of Computer Science
and Engineering
University of Moratuwa
nisansadds@cse.mrt.ac.lk

Abstract

Summarizing has always been an important utility for reading long documents. Research papers are unique in this regard, as they have a compulsory summary in the form of the *abstract* in the beginning of the document which gives the gist of the entire study often within a set upper limit for the word count. Writing the abstract to be sufficiently succinct while being descriptive enough is a hard task even for native English speakers. This study is the first step in generating abstracts for research papers in the computational linguistics domain automatically using the domain-specific abstractive summarization power of the GPT-Neo model.

Keywords: NLP, Summarization, GPT-Neo

1 Introduction

The *abstract* of a research paper provides a quick summery of the entire paper: from the problem to the proposed solution to the result. Thus by definition, this section is expected to be concise and informative (de Silva et al., 2017). Text summarization is one of the main domains in Natural Language Processing (NLP) which has numerous use cases. There are two broad categories for this: *extraction* and *abstraction*. In *extractive* methods it uses existing words, phrases or sentences to form a summary. In contrast, *abstractive* methods follow a more complex mechanisms. First, a semanatic representation of the content is built. Then natural language generation mechanisms are used to create the summary using the aforementioned representation. This research proposes a hybrid mechanism of text summarization to generate the abstract scientific papers with evaluating several paths for the proposed solution.

The objective of this research is to reduce the burden on researchers by automatically generating the abstract section by using the sections of the paper that follows it. The researchers then may do minor adjustments to the generated section and publish.

Considering existing summarization techniques, abstractive solutions have domain specific limitations. On the other hand, domain specific implementations perform better in the perspective of precise representation of the subject matter. Abstractive solutions gain domain specificity from the process of models being built upon and information extracted from the training documents. Despite the loss of generalization, this improves the accuracy of the solution within the selected domain. Thus, we propose to build and test our solution for research paper abstract generation with the scope limited to the domain of *Computational Linguistics*. As future work, it may then be extended to other research domains.

2 Related Work

El-Kassas et al. (2021) emphasize the importance of developing abstractive automatic text summarization methods. The paper describes the different approaches, methods, building blocks, techniques, datasets, evaluation methods, and future research directions of summarization methods. Referring Dutta et al. (2019), El-Kassas et al. (2021) claim that different algorithms produce different summaries from the same input texts and it is very promising to combine outputs from multiple summarization algorithms to produce better summaries. Also the recommendation of Mahajani et al. (2019) to benefit from the advantages of both extractive and abstractive approaches by proposing hybrid automatic text summarization systems, has motivated the authors to create a comprehensive survey for researchers to enhance summary generation by combining different approaches and/or methods.

Extractive text summarization methods have

Technique	ROUGE-2
Ranking-based MMR (Yang et al., 2014)	0.1262
MCMR (B&B) (Alguliev et al., 2011)	0.1221
SpOpt-comp (Yao et al., 2015a,b)	0.1245
MCMR (PSO) (Alguliev et al., 2011)	0.1165
AdaSum (Zhang et al., 2008)	0.1172
Uni + Max (Ouyang et al., 2011)	0.1133
Sum_Sparse (Li et al., 2015a,b)	0.0920
PNR^2 (Li et al., 2008)	0.0895
MDS-Sparse-div (Liu et al., 2015)	0.0645

Table 1: ROUGE score of the text summarization methods on DUC 2007 dataset in Gambhir and Gupta (2017)

been developed more often since they are less complex than abstractive methods. Gambhir and Gupta (2017) presents a comprehensive survey of recent text summarization extractive approaches developed in the last decade. A few number of abstractive and multilingual text summarization approaches also have been discussed in the paper. Their needs, advantages and disadvantages are identified and states the useful future directions. Moreover the authors have compared the summarization techniques against DUC 2007[1] dataset and calculated the ROUGE-2 (Lin, 2004) scores extracted from Gambhir and Gupta (2017) are shown in Table 1.

Moratanch and Chitrakala (2016) have done a survey on abstractive text summarization techniques, their challenges and the state of the art datasets. They claim that abstractive summarization is an efficient form if summarization compared to extractive summarization and it generates a summary that will be in more coherent form, easily readable and grammatically correct.

Abstractive summarization can be categorized into two main types as Structure based approach and semantic based approach. Moratanch and Chitrakala (2016) note that major issue of abstractive summarization is there is no generalized framework, parsing and alignment of parse trees is difficult. Extracting important sentences, sentence ordering as in original source and information diffusion are open issues according to Moratanch and Chitrakala (2016)

Bidirectional Encoder Representations from Transformers (BERT), proposed by Devlin et al. (2018), has become a mainstay in various NLP applications and have proved to produce state of the art results for numerous tasks (Ratnayaka et al., 2022). Liu and Lapata (2019) show how BERT can be applied in text summarization and propose a general framework for both extractive and abstractive summarization models. They propose a novel document level encoder based on BERT that can encode a document into representations for its sentences. Their extractive model is built in top if this encoder by stacking several intersentense transformer layers to capture document level features for extracting sentences. Their abstractive model uses an encoder-decoder architecture, combining the same pretrained BERT encoder with a randomly-initialized transformer decoder Vaswani et al. (2017).

Abstractive text summarization can be naturally cast as mapping and input sequence if words in a source document to a target of words called summary according to Nallapati et al. (2016). These deep learning based models are called sequence to sequence models. Nallapati et al. (2016) model abstractive text summariation using attentional encoder-decoder RNN and show that they achieve state of the art performance on Gigaword corpus (decribed in Rush et al. (2015)) and DUC corpus [2]. These sequence to sequence modes have been successful is many problems such as machine translation Bahdanau et al. (2014), speech recognition Bahdanau et al. (2016) and video captioning Venugopalan et al. (2015). Comparing machine translation authors highlight the challenges in summarization is unlike in translation, summarization needs to compress the original document in a lossy manner such that key concepts in the original document are preserved. But in machine translation it is expected to be loss-less and almost one-to-one word level alignment.

Nallapati et al. (2016) use an attentional encoder-decoder RNN model similar to Bahdanau et al. (2014) and show that it perform well for the metioned two corpus. They have presented a new corpus by modifying Hermann et al. (2015), named CNN/Daily Mail corpus (See, 2021) which has become a standard benchmark dataset used for evaluating the performance of different summarization models .

Cohan et al. (2018) proposed a discourse aware model for abstractive summarizing of single longer form documents such as research papers. In their encoder, they first encode each discourse section and with them then encode the document. Most of the other approaches (Liu and Lapata, 2019) and

[1] https://www-nlpir.nist.gov/projects/duc/data/2007_data.html

[2] https://duc.nist.gov/data.html/

data sets in literature such as CNN, Daily Mail (See, 2021) and New York Times (Sandhaus, 2008) articles are news paper articles which are smaller in size compared to research papers. One advantage in attempting to summarize scientific papers is that they follow a standard discourse structure and come with ground truth summaries. Thus, Cohan et al. (2018) have made two datasets collected from scientific repositories: arXiv.org[3] and PubMed.com[4].

3 Methodology

In this section, we discuss the data set generation as well as the methods used for comparative analysis.

3.1 Dataset Generation

Since we are focusing on *computational linguistics* as our domain for the abstract generation, a specific dataset was generated by collecting publicly available research papers in this domain from arXiv.org. More than 7000 research papers were downloaded in the form of LaTeX sources.

3.2 Data Preparation

Papers downloaded as LaTeX sources were then processed to *json* files by separating the sections in the paper so that abstracts can be separated in the training and testing steps. Regular expression based implementations were mainly used for the section separation task. Cleaning the LaTeX text was also done to remove unwanted latex command that won't contribute to the meaning of the text. But citations were kept remained in the cleaned text.

One constraint we had to satisfy in the model training was the max chunk size. 2048 is the maximum size we can use. Limiting number of words to this max chunk size was another problem we had to solve since research papers are comparatively long documents. This limited 2048 token size is divided into abstract, text and tags as shown in Fig 1

This size portion calculation requires a decision on the number of tokens N, to be declared as the token size of the abstract section. Instead of defining it in an arbitrary manner, we generated the Fig 2 which shows the token size distribution of the abstract sections in our data set. Thus, by looking at the 3rd quartile boundary, we selected 185 as the number of desired tokens in abstracts, N, for the

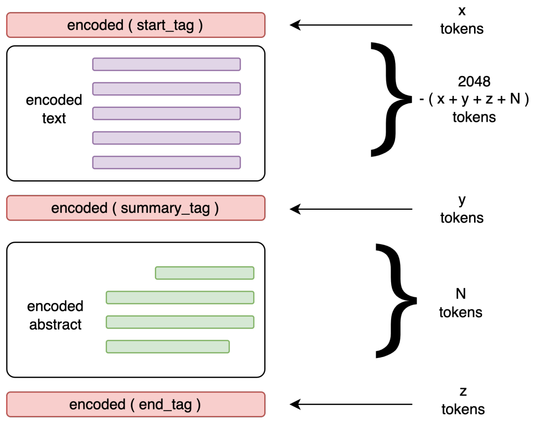

Figure 1: Token size portions for GPT-Neo model feeding

process of generating formatted text for feeding the model for training and prediction.

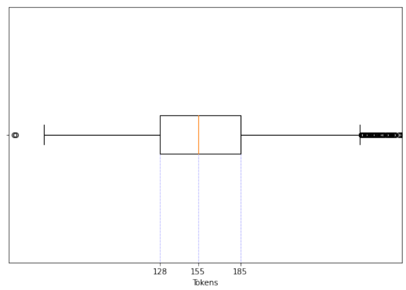

Figure 2: Token size distribution of abstract sections

After determining this N value we calculated the text body size within the constraint of 2048 total tokens. This constraint is imposed by model trained chunk size of GPT-Neo. Thus, the first N tokens are reserved for the abstract. Then. x,y and z number of tokens are put aside to carry the *start*, *summary* and *end* tags. Thus, the body text size is calculated to be $2048 - (x + y + z + N)$ number of tokens. However, as we discussed above, research papers are long documents and thus, the above calculated **Body Size** let alone even the full length of 2048 is not enough to cover the entirety of a research paper.

For this we used the pre-summarization to limit the body text into the window of **Body Size**.

3.3 Pre-Summarization

For this pre-summarization, two main mechanisms were tested.

1. Vector average method

[3] https://arxiv.org/
[4] https://pubmed.ncbi.nlm.nih.gov/

2. Extractive method

These two approach of converting long text into a trainable or predictable vector is shown in Fig 3. After the text is decreased, it will be encoded and formatted with predefined tags.

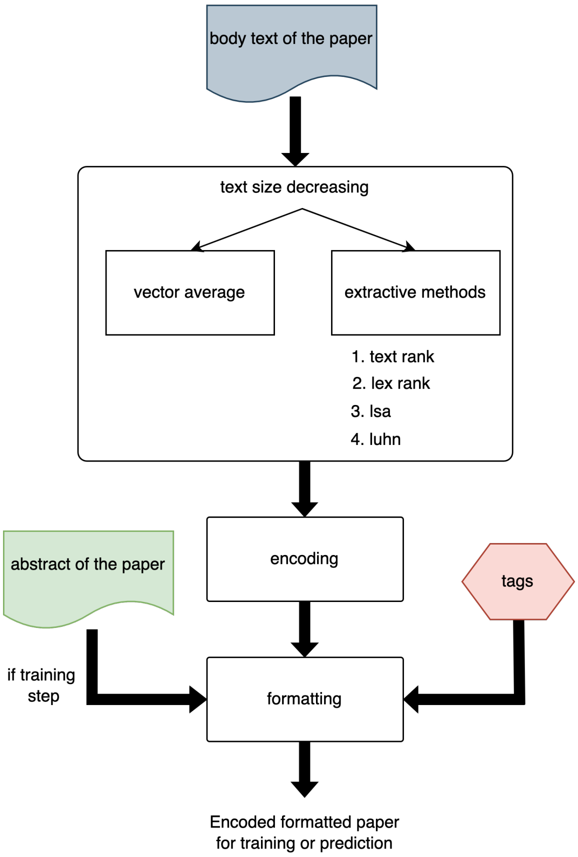

Figure 3: Data preparation overview

In **Vector average method** we divided the research paper text sans the abstract into chunks of **Body Size** and converted them using GPT-2 Tokenizer (Radford et al., 2019), which were then sent through an average pooling operation. With this, we obtain a vector of token size 2048 where the first N tokens represent the abstract with no information loss, the three flag tokens, and finally the average pooled context of the rest of the research paper like shown in the Fig 4

Extractive method simply chooses max number of sentences that can be fit inside the given token limit and it is shown in the Fig 5. But the algorithm has to select those limited sentences with preserving the original meaning of the full text. For that we have used 4 algorithms separately and evaluated the results for each method.

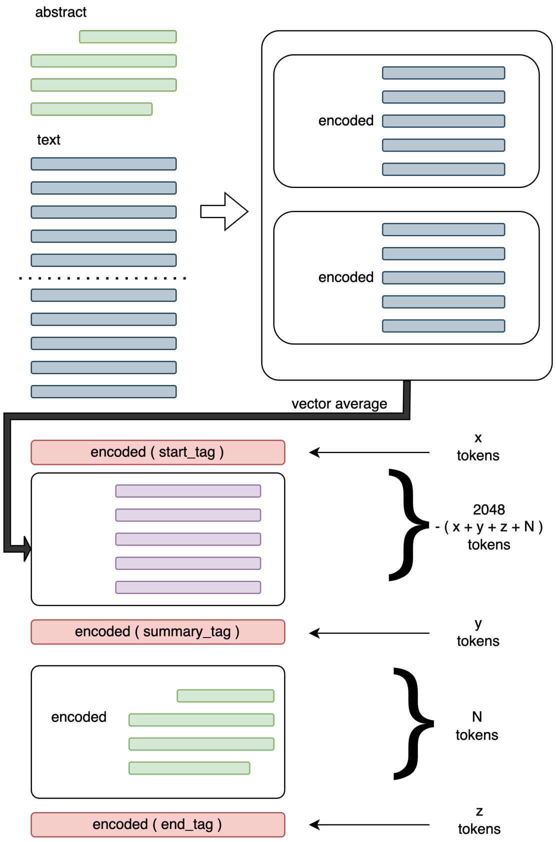

Figure 4: Tokenization strategy of vector average method

1. Lex Rank Erkan and Radev (2004) which is a stochastic graph-based method
2. Text Rank Mihalcea and Tarau (2004) which is a graph based ranking model
3. Latent Semantic Analysis(LSA) Landauer et al. (1998) which is a semantic based algorthm
4. Luhn (Luhn, 1958) which is a significance based algorithm

After these text is limited to to the given **Body Size** by any of the method describe above, they were then converted to *tfrecords* which supports distributed datasets and leverages parallel I/O. Generation of these *tfrecords* were done by encoding the LaTeX source of each paper. A predefined *start tag*, *summary tag*, and *end tag* were applied in this encoded vector so that the model can be guided on what type of text to predict in the respective subsections of the predicted text.

The 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)
Taipei, Taiwan, November 21-22, 2022. The Association for Computational Linguistics and Chinese Language Processing

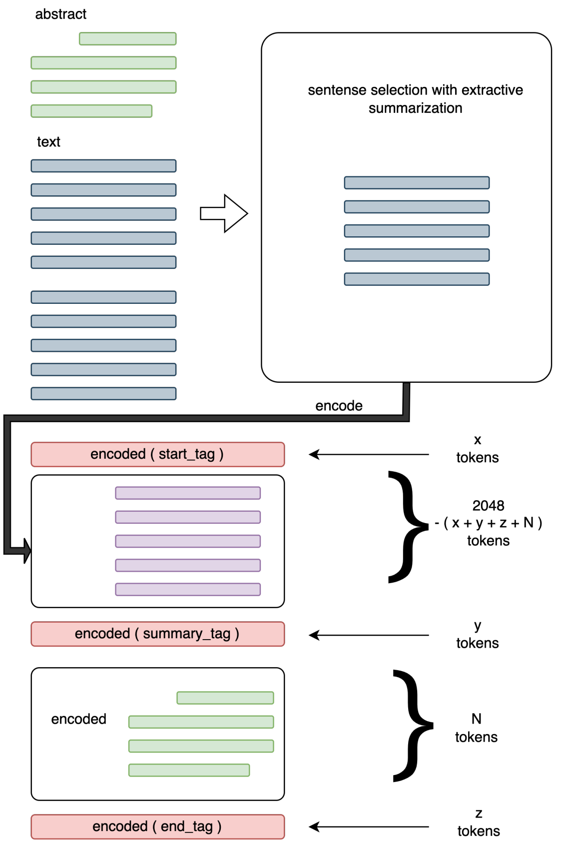

Figure 5: Tokenization strategy of extractive methods

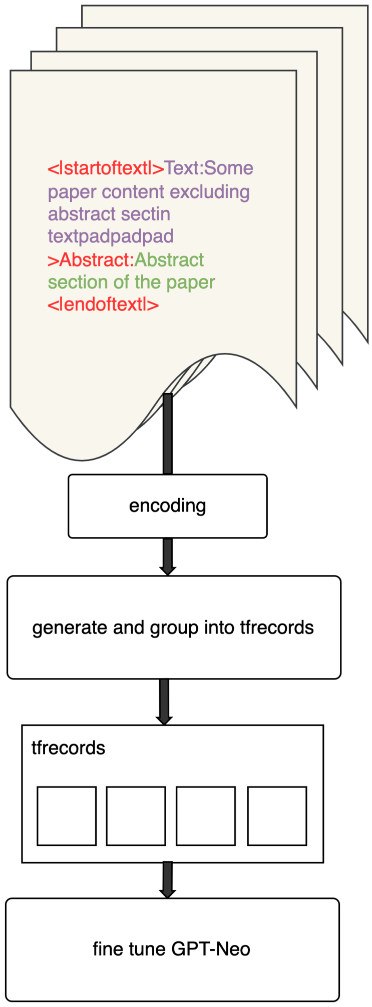

Figure 6: Fine tuning GPT-Neo

3.4 Model Tuning

GPT-Neo (Black et al., 2021) model was fine tuned with the dataset after text size reduction as described in Fig 3.2 and tokenized with GPT-2 tokenizer. Fine tuining was done using Google Colab[5] with the TPUs. Since using TPUs dataset and pretrained model were stored in the google cloud[6] and then processed with colab with the power of TPUs[7]. Fine tuning text format is shown in the Fig 6 GPT-Neo model was fine-tuned with batch size of 8, mesh shape of x:4,y:2, train steps of 1000 and steps per checkpoint of 500.

3.5 Prediction

Fine tuned GPT-Neo (Black et al., 2021) models were used with encoded text of the papers by related pre-summarization methods. Prediting was also done using Google Colab with the power of TPUs. Prediction text format is shown in Fig 7. As shown in Fig 7, abstract tag is provided so that GPT-Neo can predict the text from that point until it predict the end of text tag.

For the prediction, GPT-Neo model was utilized with batch size of 1, mesh shape of x:4,y:2, train steps of 1000 and steps per checkpoint of 500. This effectively mirrors our training configuration discussed in Section 3.4.

4 Results

Separately fine tuned GPT-Neo models were evaluated for each pre-summerizer as shown in Table 4; where it can be observed that Latent Semantic Analysis and Luhn based pre-summarizations have obtained the best results for the tested *ROUGE* scores.

It was then decided to analyse the configurations given in Table even further by considering the Precision and Recall measures as there are different research domains that give priority to one over the other. For example, de Silva (2020) discussed how in the case of medical domain NLP, recall takes precedence over precision. Same is discussed for

[5] https://colab.research.google.com/
[6] https://cloud.google.com/storage
[7] https://cloud.google.com/tpu

Pre-Summarization Method	ROUGE-1	ROUGE-2	ROUGE-L
Vector Average	0.1843	0.0204	0.1698
Lex Rank	0.2612	0.0478	0.2359
Text Rank	0.2548	0.0441	0.2304
LSA	**0.2629**	0.0472	**0.2382**
Luhn	0.2602	**0.0483**	0.2343

Table 2: ROUGE Scores comparison of the models based on the pre-summarization method

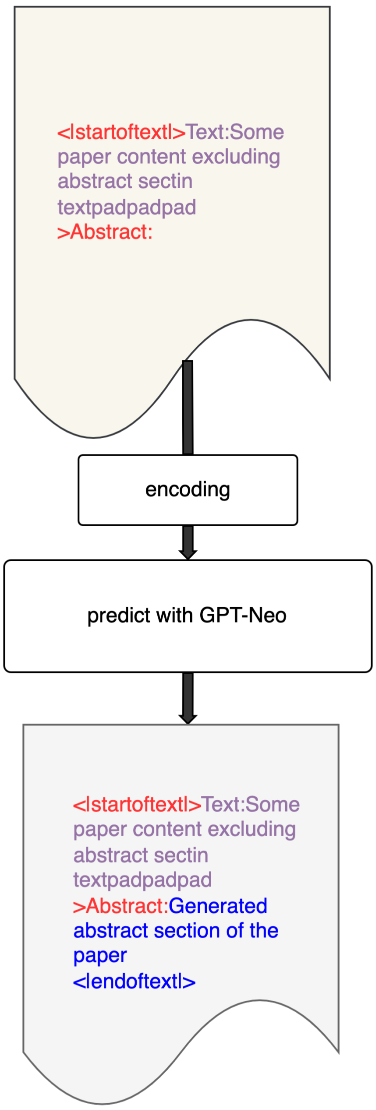

Figure 7: Predicting summary with GPT-Neo

the legal domain by Samarawickrama et al. (2020). Even though there is no definitive meta-study on the content in the research papers of the computational linguistics domain to conclude such a bias towards precision or recall, it was deemed prudent to report these values. For the ease of reading and comparison, the F1 values of Table are also brought forward.

Average vector method takes the average of encoded vectors of the chunks divided from the text of the paper before passing it into GTP-Neo for training or predicting. While average vector model seems to be too trivial for this task at a glance, recent prior work in the NLP domain have proved its usefulness at establishing a baseline for even complex tasks such as sentiment analysis (Jayawickrama et al., 2021). Results of this method are shown in Table 3.

ROUGE	F	P	R
1	0.1843	0.2157	0.1684
2	0.0204	0.0242	0.0187
L	0.1698	0.1987	0.1551

Table 3: ROUGE Scores of average vector based pre-summarizing.

Lex rank (Erkan and Radev, 2004) is a stochastic graph-based method for computing relative importance of textual units. It is based on the concept of eigenvector centrality in a graph representation of sentences. Similar, but mathematically simpler methods have shown promise in NLP applications in the Legal domain (Jayawardana et al., 2017). Model we trained with Lex rank has given the results shown in Table 4.

ROUGE	F	P	R
1	0.2612	0.3032	0.2384
2	0.0478	0.0568	0.0435
L	0.2359	0.2742	0.2152

Table 4: ROUGE Scores of Lex rank based pre-summarizing.

Since the advent of *PageRank* algorithm (Page, 1997; Page et al., 1999), using graph-based methods to rank text documents has been a popular solution for document level analysis (Karannagoda et al., 2013). *TextRank* (Mihalcea and Tarau, 2004) is also a graph based sentence extraction method which creates a graph for each sentence and rank them based on the similarity. In their legal document retrieval system, Sugathadasa et al. (2018) showed how *TextRank* can be utilized in representing documents in a semantically consistent manner. Pre-summarization based on this method has scored as shown in the Table 5.

ROUGE	F	P	R
1	0.2548	0.2916	0.2342
2	0.0441	0.0514	0.0403
L	0.2304	0.2637	0.2117

Table 5: ROUGE Scores of Text rank based pre-summarizing.

LSA (Latent Semantic Analysis) (Landauer et al., 1998) method is extracting and representing the contextual-usage meaning of words by statistical computations applied to the text. We have calculated the ROUGE scores of this method as a pre-summarizer with GPT-Neo and the results are shown in Table 6.

ROUGE	F	P	R
1	0.2629	0.302	0.2421
2	0.0472	0.0547	0.0435
L	0.2382	0.2737	0.2194

Table 6: ROUGE Scores of LSA based pre-summarizing.

Luhn algorithm (Luhn, 1958) calculates the significance of a sentence by considering frequency of word occurrence in the text and the relative position within a sentence. GPT-Neo Model trained Luhn algorithm as a pre-summarizer gave the results shown in Table 7.

ROUGE	F	P	R
1	0.2602	0.2954	0.2406
2	0.0483	0.0551	0.0448
L	0.2343	0.2663	0.2164

Table 7: ROUGE Scores of Luhn based pre-summarizing.

LSA based pre-summarization method has been scored the highest on ROUGE-1 and ROUGE-L while Luhn based pre-summarization method is scoring higher on ROUGE-2. All extractive pre-summarizations has been scored more than the twice of the score of the baseline, vector average method, in ROUGE-2.

5 Conclusion

We have used transfer learning with GPT-Neo for generating abstracts of research papers automatically. GPT-Neo model provides a language model that can be utilized for many tasks but we have to face the token limitation. We managed this limited token size with two main approaches which are, an average-pooling of the body context vectors and an extractive summarization. Observations have shown that extractive pre-summarization with GPT-Neo has better results compared to average pooling. We intend to extend the findings to generate the introduction as well.

References

Rasim M Alguliev, Ramiz M Aliguliyev, Makrufa S Hajirahimova, and Chingiz A Mehdiyev. 2011. Mcmr: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications*, 38(12):14514–14522.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

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. *arXiv preprint arXiv:1804.05685*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Soumi Dutta, Vibhash Chandra, Kanav Mehra, Sujata Ghatak, Asit Das, and Saptarshi Ghosh. 2019. *Summarizing Microblogs During Emergency Events: A Comparison of Extractive Summarization Algorithms: Proceedings of IEMIS 2018, Volume 2*, pages 859–872.

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

Vindula Jayawardana, Dimuthu Lakmal, Nisansa de Silva, Amal Shehan Perera, Keet Sugathadasa, and Buddhi Ayesha. 2017. Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings. In *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*, pages 79–84. IEEE.

Vihanga Jayawickrama, Gihan Weeraprameshwara, Nisansa de Silva, and Yudhanjaya Wijeratne. 2021. Seeking sinhala sentiment: Predicting facebook reactions of sinhala posts. *arXiv preprint arXiv:2112.00468*.

E. L. Karannagoda, H. M. T. C. Herath, K. N. J. Fernando, M. W. I. D. Karunarathne, N. H. N. D. de Silva, and A. S. Perera. 2013. Document Analysis Based Automatic Concept Map Generation for Enterprises. In *Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on*, pages 154–159. IEEE.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Chen Li, Yang Liu, and Lin Zhao. 2015a. Using external resources and joint learning for bigram weighting in ilp-based multi-document summarization. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 778–787.

Piji Li, Lidong Bing, Wai Lam, Hang Li, and Yi Liao. 2015b. Reader-aware multi-document summarization via sparse coding. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Wenjie Li, Furu Wei, Qin Lu, and Yanxiang He. 2008. Pnr2: Ranking sentences with positive and negative reinforcement for query-oriented update summarization. In *Proceedings of the 22nd international conference on computational linguistics (Coling 2008)*, pages 489–496.

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.

He Liu, Hongliang Yu, and Zhi-Hong Deng. 2015. Multi-document summarization based on two-level sparse representation model. In *Twenty-ninth AAAI conference on artificial intelligence*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Abhishek Mahajani, Vinay Pandya, Isaac Maria, and Deepak Sharma. 2019. A comprehensive survey on extractive and abstractive techniques for text summarization. *Ambient Communications and Computer Systems*, pages 339–351.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

N Moratanch and S Chitrakala. 2016. A survey on abstractive text summarization. In *2016 International Conference on Circuit, power and computing technologies (ICCPCT)*, pages 1–7. IEEE.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. *Information Processing & Management*, 47(2):227–237.

Lawrence Page. 1997. Method for node ranking in a linked database. *USA Patent*, 6.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Gathika Ratnayaka, Nisansa de Silva, Amal Shehan Perera, Gayan Kavirathne, Thirasara Ariyarathna, and Anjana Wijesinghe. 2022. Context sensitive verb similarity dataset for legal information extraction. *Data*, 7(7).

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685.

Chamodi Samarawickrama, Melonie de Almeida, Nisansa de Silva, Gathika Ratnayaka, and Amal Shehan Perera. 2020. Party identification of legal documents using co-reference resolution and named entity recognition. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 494–499. IEEE.

Evan Sandhaus. 2008. The new york times annotated corpus.

Abigail See. 2021. Github - abisee/cnn-dailymail: Code to obtain the cnn / daily mail dataset (non-anonymized) for summarization.

Naida Hewa Nisansa Dilushan de Silva. 2020. *Semantic Oppositeness for Inconsistency and Disagreement Detection in Natural Language*. Ph.D. thesis, University of Oregon.

Nisansa de Silva, Dejing Dou, and Jingshan Huang. 2017. Discovering Inconsistencies in PubMed Abstracts Through Ontology-Based Information Extraction. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ACM-BCB '17, pages 362–371, New York, NY, USA. ACM.

Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. 2018. Legal Document Retrieval using Document Vector Embeddings and Deep Learning. In *Science and information conference*, pages 160–175. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.

Libin Yang, Xiaoyan Cai, Yang Zhang, and Peng Shi. 2014. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information sciences*, 260:37–50.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2015a. Compressive document summarization via sparse optimization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2015b. Phrase-based compressive cross-language summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 118–127.

Jin Zhang, Xueqi Cheng, Gaowei Wu, and Hongbo Xu. 2008. Adasum: an adaptive model for summarization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 901–910.