Project Final Report: Text Summarization Using NLP

Mark Trovinger
Purdue Fort Wayne
tromv01@pfw.edu

Atharva Atre
Purdue Fort Wayne
atreaa01@pfw.edu

Navyaprabha Rajappa
Purdue Fort Wayne
rajan02@pfw.edu

Dr. Rusert
Advisor - PFW NLP
jrusert@pfw.edu

Abstract

This technical proposal presents a project to develop a tool for text summarization using natural language processing (NLP) techniques. The motivation for this project stems from the need to summarize lengthy texts, such as meeting minutes and news articles, for which taking notes can be overwhelming. The proposed solution will explore both extractive and abstractive summarization methods, with a focus on algorithms such as LexRank, LSA, Luhn, KL Sum, and Transformer-based models. The CNN/Daily Mail dataset was chosen for training and evaluation of the models. Various performance evaluation metrics considered like BLEU , ROGUE score, F1-score etc. We later finalized on variations of the ROGUE score. The project also includes a hyper-parameter optimization section, where we fine-tune transformer based models like BERT to perform better on summariation tasks. The results of this project will help identify the most appropriate technique for specific use cases and contribute to the development of efficient and accurate NLP tools for text summarization.[1]

1 Introduction

Automatic text summarization, a sub-field of Natural Language Processing (NLP), has become increasingly vital in recent years, owing to the exponential growth of digital information. The ability to condense extensive documents into coherent and informative summaries is essential for information retrieval, knowledge management, and decision-making. Various NLP models have been proposed and implemented for text summarization tasks, each exhibiting unique strengths and weaknesses. In this research paper, we present a comprehensive comparative study of six prominent NLP models, namely Spacy, HeapQ, TextRank, BERT, BART, and seq-to-seq, with a focus on extractive and abstractive summarization techniques.

We delve into the architectural intricacies and methodologies employed by each model, and evaluate their performance on the widely-adopted CNN/DailyMail dataset[2]. Recognizing the potential of BERT and BART in generating high-quality summaries, we further investigate the impact of hyperparameter optimization on their performance. By conducting an extensive grid search and iterative fine-tuning, we identify optimal configurations that enhance the models' capabilities in generating accurate and coherent summaries.

The results of our study not only provide valuable insights into the comparative effectiveness of these NLP models, but also contribute to the ongoing quest for refining and improving text summarization techniques. Moreover, our findings on hyperparameter optimization for BERT and BART can serve as a foundation for future research, paving the way for the development of more advanced and efficient summarization algorithms.

1.1 Problem

The problem is the overwhelming amount of information that people encounter daily like News articles, Research papers, and other documents. With the rise of the internet, social media, and other digital technologies, we are inundated with vast amounts of text-based information that we simply don't have time to read and process.

1.2 Motivation

The main motivation of text summarization is to compress lengthy text into concise summaries that preserve the important information. This approach aims to provide individuals with a quick overview of a document's content, enabling them to glean insights without having to read the entire text.

[1] https://github.com/atharvapurdue/text_summarization

[2] https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail

2 Related Work

In this section, we provide an overview of the related work in the field of text summarization, focusing on the six NLP models discussed in our study. We also highlight previous research on hyperparameter optimization, which serves as the basis for our investigation of BERT and BART.

2.1 Spacy

Spacy is an open-source library designed for various NLP tasks, including tokenization, part-of-speech tagging, named entity recognition, and dependency parsing (Honnibal and Montani, 2017) Although not specifically tailored for text summarization, Spacy's functionalities have been leveraged in research to develop custom extractive summarization algorithms. (Rajpurkar et al., 2018)

2.2 HeapQ

HeapQ is an unsupervised extractive summarization model that employs a modified version of the Heap's Law to rank sentences based on their importance (Damodaran et al., 2021). The model has been shown to generate coherent and informative summaries, particularly for shorter texts.

2.3 TextRank

TextRank is a graph-based algorithm for extractive summarization inspired by Google's PageRank (Mihalcea and Tarau, 2004). The model computes sentence importance scores based on their similarity to other sentences within the document, and constructs summaries by selecting the highest-ranked sentences. Several studies have extended and improved TextRank for various applications, including multi-document summarization and keyphrase extraction(Lu et al., 2016; Barrios et al., 2016).

2.4 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained deep learning model designed for a wide range of NLP tasks, including text summarization (Devlin et al., 2018). By leveraging its bidirectional context representation and transfer learning capabilities, BERT has been successfully applied to both extractive and abstractive summarization tasks (Liu and Lapata, 2019; Zhang et al., 2019).
NOTE - Dr. Rusert asked us how exactly BERT was modified for text summarization task. Here is the explaination -
In this paper, they proposed a method that modifies the BERT architecture for text summarization tasks. They introduced interval segment embeddings and a custom positional embedding matrix to the original BERT model, enabling it to better capture document-level context and effectively rank sentences for extractive summarization. For abstractive summarization, they incorporated the BERT model as the encoder in an encoder-decoder framework, where the decoder is a Transformer-based architecture with a copy mechanism. This setup allows the model to generate abstractive summaries by leveraging both the source text and the contextualized representations provided by the BERT encoder. These modifications to the BERT architecture enabled BERTSUM to achieve state-of-the-art performance in both extractive and abstractive summarization tasks on the CNN/Daily Mail dataset.

2.5 BART

BART (Bidirectional and Auto-Regressive Transformers) is a denoising autoencoder based on the Transformer architecture, specifically designed for text generation tasks, including abstractive summarization (Lewis et al., 2020). The model has been demonstrated to outperform other state-of-the-art models on various summarization benchmarks, thanks to its ability to reconstruct corrupted input text (Raffel et al., 2020).

2.6 Seq-to-Seq

Sequence-to-sequence (seq-to-seq) models are a family of neural network architectures that transform an input sequence into an output sequence, making them suitable for abstractive summarization tasks (Sutskever et al., 2014). Notable seq-to-seq models include the LSTM-based encoder-decoder framework (Bahdanau et al., 2015) and the attention-based Pointer-Generator Network (See et al., 2017).

Implemented a sequence-to-sequence model using LSTM for text summarization. The model architecture is defined using the encoder-decoder framework with bidirectional LSTM layers in the encoder, and a unidirectional LSTM layer followed by a dense layer in the decoder. The model is then compiled with the RMSprop optimizer and sparse categorical cross-entropy loss. The model is trained on the training data and validated on a subset of the data using early stopping. The trained model is then evaluated on the test data, and the predictions are generated in the text (summary) format

2.7 Hyperparameter Optimization

Hyperparameter optimization has been extensively studied in the context of deep learning, with applications in various NLP tasks, including text summarization. Techniques such as grid search, random search, Bayesian optimization, and genetic algorithms have been employed to identify optimal configurations for improved model performance (Bergstra et al., 2011; Snoek et al., 2012). Previous research has also explored hyperparameter optimization for BERT and BART in tasks such as sentiment analysis, question-answering, and text classification (Sun et al., 2020; Garg et al., 2021). However, to the best of our knowledge, the impact of hyper-parameter optimization on these models in the context of text summarization remains under-explored, due to frequent updates in the training dataset and model architectures. We performed Hyperparameter optimization on BERT model, by using grid search method and the results were different from the one which the paper reported. The results were off, by a factor of 100, so it is likely that the results we got were in percentage, as we used a different version of the eval() metric than the author. Here are the results (taking the above assumption into account).

Model	ROUGE 1	ROUGE 2	ROUGE-L
BERT*	35.43	15.58	24.17
BERT**	41.28	18.68	28.19
BERT***	44.68	21.38	30.70

*Baseline model trained by us.
*Huggingface Hyperparameter optimization model (not by us).
**Hyperparameter optimized model trained by us (results on github)

3 Methods

In order to establish a baseline for later, more complex transformer-based models, we decided to test the models mentioned in related work, namely spACy, HeapQ, and TextRank. As mentioned later in the experiments section, we initially used only a small subset of the data for these initial experiments. This would be revised later, when it became clear that these results were essentially meaningless, and that we needed to have a better understanding of the ROUGE metric before continuing.

It must also be mentioned that as extractive methods for text summarization, there are no ”models” to be trained, in the sense of deep learning models having weights adjusted through utilization of an optimizer and loss function. This is due to the fact that TextRank and HeapQ only select sentences that are statistically relevant and the top sentences are then selected to be the summarization.

After exploring the smaller models, we turned our attention to the larger, transformer-based models mentioned above.

4 Experiments

Our initial experiments using the less complex models involved training them on a very small subset of the CNN/Daily Mail dataset. This was done in part due to compute resource concerns, as training on the full dataset was considered too computationally expensive for baseline results.

5 Results

5.1 Extractive Methods

The first set of results were using the extractive methods on a small subset of the dataset, with the results presented below.

Algorithm	f1 Score
spaCy	0.114012
HeapQ	0.13333
TextRank	0.2565866

F1 score was chosen as the metric for this portion due primarily to our familiarity with it, but we would change to ROUGE scores later. NOTE - In the final presentation, Dr. Rusert asked us how the F1 Score is computed for text summarization. F1 score is computed by comparing the system-generated summary with a human-written summary, or a gold standard summary. The system-generated summary is evaluated based on the overlap of its content with the content of the gold standard summary. The F1 score is a weighted harmonic mean of the precision and recall values.

5.2 Abstractive Methods

While the CNN/Daily Mail dataset is primarily an extractive dataset, it is useful to see how well the more complex, abstractive models, such as BERT, BART pretrained models and Seq2Seq enc-dec LSTM model perform on a dataset which is extractive in nature. The results of each models is below.

Model	ROUGE 1	ROUGE 2	ROUGE-L
BART	39.8079	18.036	24.8074
BERT	35.4348	15.5857	24.1659
Seq2Seq	31.1983	12.3501	21.8015

As seen above, BART and BERT perform better than Seq2Seq LSTM model. BART is giving slightly better results in ROUGE 1, ROUGE 2 and ROUGE-L scores than BERT.

6 Future Work

The findings of our research offer several avenues for future exploration in the domain of text summarization using NLP models. In this section, we outline potential directions that can build upon the current study and contribute to the advancement of automatic summarization techniques.

6.1 Exploration of Other NLP Models

While our study provides a comparative analysis of six prominent NLP models, the rapid evolution of the NLP landscape necessitates the continuous assessment of new models and architectures. Future work could expand the scope of our research by evaluating emerging models, such as GPT-4, DeBERTa, and RoBERTa, for their potential in text summarization tasks.

6.2 Domain-Specific Adaptations

Our study employs the CNN news dataset for evaluating the performance of the models. However, domain-specific adaptations of these models for areas such as legal, medical, or scientific text summarization remain largely unexplored. Future research could investigate the adaptability and performance of these models in various domains, potentially leading to the development of specialized summarization techniques.

6.3 Multi-document Summarization

The current study focuses on single-document summarization tasks. Expanding the scope to multi-document summarization, which involves generating a coherent summary from multiple sources, could offer valuable insights into the capabilities of NLP models in handling more complex summarization tasks.

6.4 Incorporation of Multimodal Data

As modern data becomes increasingly multimodal, incorporating visual, auditory, and textual information, future research could explore the integration of multimodal data into the summarization process. This could involve developing models that are capable of generating summaries from both textual and non-textual sources, such as images or videos, to provide a more comprehensive understanding of the content.

6.5 Explainable AI for Summarization

The interpretability and explainability of deep learning models, including BERT and BART, remains an open research question. Investigating methods for generating more transparent and interpretable summaries could contribute to the development of explainable AI for text summarization, enabling users to better understand the reasoning behind generated summaries.

6.6 Evaluation Metrics and Human Evaluation

Our study relies on widely-used evaluation metrics, such as ROUGE, to assess the quality of generated summaries. However, future work could explore the development of novel evaluation metrics that capture additional aspects of summary quality, such as coherence, informativeness, and readability. Moreover, incorporating human evaluation into the assessment process could provide a more nuanced understanding of the models' performance from a user perspective.

7 Conclusion

In conclusion, text summarization is an essential task in NLP, and its applications are diverse and far-reaching. Also it holds great potential for improving human-machine communication and facilitating information access and dissemination in various domains. In this paper, we presented an overview of various techniques and methods used for text summarization, including extractive and abstractive summarization, and highlighted some of the recent advancements and challenges in the field. However, there is still room for improvement, especially in dealing with rare or out-of-vocabulary words and generating more fluent and coherent summaries.

While text summarization is not a solved problem, it is an area of active research. Our goal with this project was to illustrate what the state of the art is within this field. As larger and more complex models are developed, this should push the field forward, offering users better options for summarizing long samples of text.

One area of research that has shown great promise that we were unable to cover is large language models used in chatbots, such as ChatGPT. ChatGPT seems uniquely capable of digesting complex topics and creating accurate summaries, befitting its origin as a generative model (Generative Pre-trained Transformer). This style of text summarization, along with the ability to add style transfer ("...write me a letter asking for money in a professional way") appears to be a very promising avenue of research.

References

Dzmitry Bahdanau, Dmitriy Serdyuk, Philemon Brakel, Nan Rosemary Ke, Jan Chorowski, Aaron Courville, and Yoshua Bengio. 2015. Task loss estimation for sequence prediction. *arXiv preprint arXiv:1511.06456.*

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606.*

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

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.*

Siddhant Garg, Goutham Ramakrishnan, and Varun Thumbe. 2021. Towards robustness to label noise in text classification via noise modeling. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3024–3028.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Guangming Lu, Yule Xia, Jiamei Wang, and Zhenling Yang. 2016. Research on text classification based on textrank. In *2016 International Conference on Communications, Information Management and Network Security*, pages 319–322. Atlantis Press.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. ArXiv:1806.03822 [cs].

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Deliang Sun, Haijia Wen, Danzhou Wang, and Jiahui Xu. 2020. A random forest model of landslide susceptibility mapping based on hyperparameter optimization using bayes algorithm. *Geomorphology*, 362:107201.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.