Project Report CS 6120 - Summer 2023

Text Summarizer for Research Papers

Group 10:

Sumukhi Ganesan - NUID 002682285
LeAnn Marie Mendoza - NUID 002965164
Sindhu Swaroop - NUID 001006558

1 Introduction

Text summarization is the task of producing a concise and fluent summary of a text document while preserving its key information and overall meaning. This task requires the summarizer to understand the text, identify its context, and then generate a new text that captures the content. There are two main approaches to text summarization: extractive and abstractive. Extractive summarization methods identify important sentences or phrases from the text and then combine them to form a summary. Abstractive summarization attempts to learn the context, semantics and relationships of different parts of a text using machine learning and generates summaries consisting of novel sentences that might not appear in the original content.

Research paper summarization is a unique task in the realm of summarization due to the dense, technical, and multifacted nature of scientific literature. As the volume of publications grow, it becomes an increasing challenge for academic professionals to keep up to date with the latest findings. Accurate summaries of research can assist in sifting through volumes of intricate details while maintaining the core competencies of the research publication.

In this project, we explored several different approaches to extractive as well as abstractive text summarization of research papers utilizing the ScisummNet dataset [1]. We will compare the performance of a state-of-the-art text summarization transformer, BART [2] with our devised text summarization methods, TextRank based summarization, Clustering based summarization, and a fine-tuned BART model.

2 Methods

2.1 Datasets

We have used the ScisummNet dataset [1] to train and evaluate our text summarization methods. The ScisummNet dataset is a large annotated corpus of scientific papers and their summaries, specifically collected for scientific paper summarization tasks. The dataset contains 1000 research papers and their corresponding summaries in the form of abstracts, citation sentences and comprehensive manual summaries. The dataset has been derived from the ACL anthology network as part of the CL-SciSumm project by the Yale LILY Lab, originally intended for training their data-driven neural summarization models for summarization.The dataset is publicly available for download.

2.2 Data Preprocessing

The downloaded dataset contains the ScisummNet corpus description and subdirectories for the 1000 papers. Each paper directory contains the paper's XML file, annotated citation information (in JSON format), and a manual gold summary. For the purposes of this project, we read the text contents of each paper by parsing its XML file as a tree and reading from it. We then apply regular cleaning and text filters to remove the title, whitespaces, hyperlinks, punctuations and other unnecessary characters. The text is converted to lowercase for standard processing. The manual summary and abstracts are also extracted in parallel to facilitate the evaluation of the summarization methods.

2.3 Transformer based Summarization (BART)

To serve as a baseline for our summarizer development, we employed the current SOTA text summarizer method, Bidirectional and Auto-Regressive Transformers (BART) developed by Facebook AI [2]. BART utilizes abstractive text summarization that uses transformer models that combine both auto-regressive and auto-encoding strategies. This is done by first corrupting the input text by masking out certain words or shuffling the order of sentences, and then attempting to reconstruct the original from this corrupted version - allowing BART to learn the context from both tokens. To deploy this model, we utilized pre trained bart-large model with 12 encoder and decoder layers and 40M parameters (Available here: https://huggingface.co/facebook/bart-large) to generate summaries from our data.

Additionally, we fine-tuned the BART model, originally trained on a large corpus of sequence-to-sequence tasks, to improve its performance on our ScisummNet dataset. This involved tokenizing our data, setting up training configurations, and training the fine-tuned BART on our dataset.

2.4 TextRank based Summarization

The second method of summarization that we explore in this project is sentence extraction and summarization using TextRank[3] algorithm. Amongst the many available techniques for summarization, TextRank summarization has emerged as a powerful and widely adopted approach.

TextRank algorithm is inspired by graph theory and extends the idea of ranking web pages on the internet based on their relationships with one another to sentences in a document. The method essentially represents the document as a directed graph in which sentences are nodes with edges defined by their similarity between them. There are two important reasons for chossing TextRank summarization for this project. The first reason is that it is an unsupervised method of summarization. Considering the scarcity of organized research paper datasets available and the limited resources of an academic project, TextRank fits perfectly as it requires no model training and can directly generate summaries. Secondly, the extractive nature of the method implies, instead of writing entirely new sentences, which requires semantic understanding i.e. training, the model simply selects the high ranking sentences from the original text to form a coherent summary. This ensures that the summary represents the key concepts and ideas while staying true to the original text.

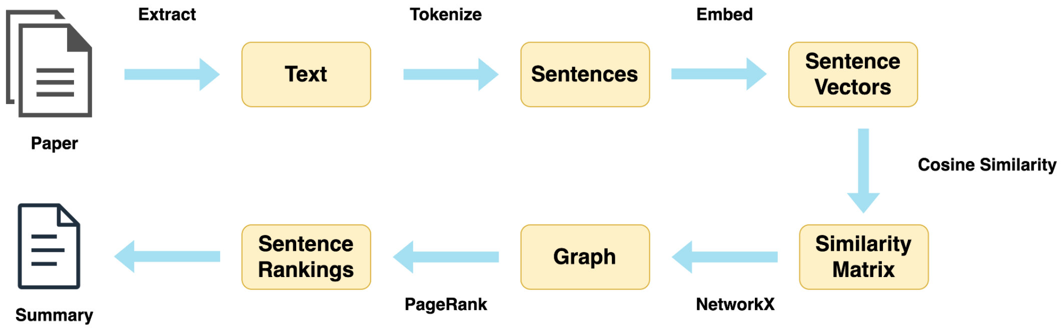

Figure 1. Visual representation of the TextRank algorithm for Summarization

The implementation of TextRank model, specifically in this project, involved a few design decisions. While there are many variants of the TextRank model based on the ranking algorithm , we decided to stick to the original algorithm, PageRank[4] as suggested by the authors[3]. In order to generate the rank order, sentences in the source text needed to be represented as sentence vectors. We selected the pretrained "SPECTER"[5] model for generating sentence embeddings. This model was selected for its fine-tuned knowledge of scientific publications. Interestingly, while most methods aggregate word level embeddings up to sentence level embeddings, this model is at document-level and therefore represents sentences as individual 1-sentence documents.

We calculate the weights of edges between any two sentences using the cosine similarity between their vectors. We explored other methods including common tokens, LCS length and isf-modified-cosine similarity[6] for more sensitive weight values and have identified that cosine similarity is the most effective amongst all for summarization.

The similarity matrix thus generated is converted into a graph and the rank for each node i.e. sentence is calculated. Based on their TextRank scores, we arrange the sentences in their rank order, highest to lowest. A set of top n sentences in chosen from this ranked array as the summary where n is a hyperparameter. We have chosen the n value for this experiment as 10 based on the average count of sentences in the manual summaries. Varying summary lengths could vary the quality of the generated summary and is usually a trade off between more context and conciseness of the summary.

2.5 Clustering based Summarization

This method is based on an unsupervised learning technique - clustering. Sentences are clustered based on their similarity, and representative sentences from each cluster are selected for the summary.

Step 1: The first step was to apply a **sentence embedding model** to the corpus to extract sentence embeddings. We used the **"distilbert-base-nli-mean-tokens"**, a specific pre-trained version of the DistilBERT model, which is fine-tuned for natural language inference (NLI) tasks. Here, "base" refers to the size of the model and is usually designed smaller to ensure a higher computational efficiency. This particular variant of DistilBERT is widely used for tasks that require sentence-level embeddings, such as sentence classification and sentence similarity. Each sentence is encoded into a fixed-size vector of 768 dimensions that captures the semantic information and contextual understanding of the input sentence.

Step 2: After completing the initial data processing, the subsequent phase involved selecting an appropriate clustering algorithm. For this task, we opted for the **K-means clustering algorithm.** Our first objective was to determine the ideal number of clusters for the given problem. To accomplish this, we employed the elbow method to find the optimum number of clusters for each research paper. We used the KneeLocator function and found that the **optimal number of clusters for our dataset most commonly lies between 3 and 4**. This conclusion was supported by observing the point on the plot where the inertia decrease starts to level off, signifying the optimal value for the number of clusters.

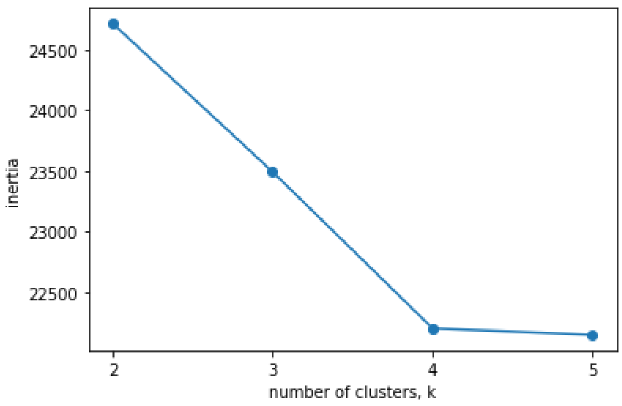

Figure 2. Elbow method for choosing optimal clusters

Step 3: After choosing the best value of K, we proceeded to **train the K-means clustering model** using our corpus embeddings. This resulted in each sentence in the corpus being assigned to a specific cluster based on their similarities. By leveraging the trained K-means model, we were able to effectively group similar sentences together into clusters. This clustering process allowed us to gain valuable insights into the underlying patterns and structures present within the corpus.

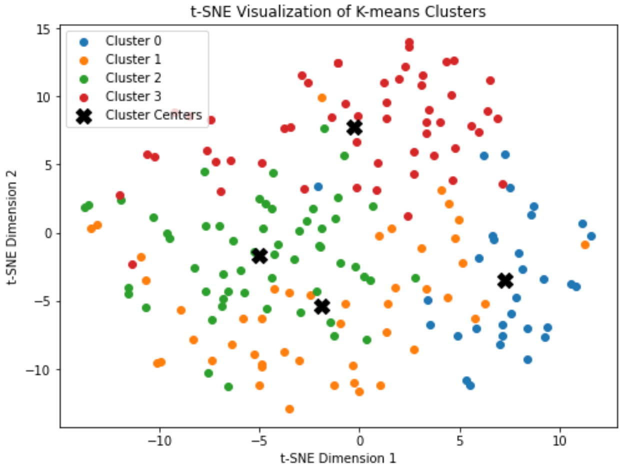

Figure 3. Visualization of K-means clusters

Step 4: The next step was to determine cluster representation and select a representative sentence from each cluster that best captures the main idea of the cluster. After observing the clustering of several different corpora, we noticed a consistent pattern where the first sentence in each cluster appeared to convey the most significant and representative information and did the best job of introducing the main topic and providing essential context for the rest of the sentences. We then collected the representative sentences from each cluster and joined them to form the summary.

3 Results and Discussion

To evaluate our text summarizer application, we used a standard evaluation metric ROUGE (Recall-Oriented Understudy for Gisting Evaluation)[7]. ROUGE measures the overlap between the generated summary and the reference summary in terms of unigrams (ROUGE-1), bigrams (ROUGE-2) and longest common shared sequence (ROUGE-L). The Scisummnet dataset provides us with reference summaries. For each document, we compared the summary generated by our model against the reference summary as well as the abstract using ROUGE metrics and computed the average scores across the test set.

Summarizer	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
Pretrained BART	0.47	0.36	0.42
Finetuned BART	0.50	0.35	0.42
Text Rank	0.34	0.09	0.18
Clustering	0.28	0.11	0.17

Table 1. ROUGE F1-scores similarity results of developed summarizer models.

3.1 Transformer based Summarizer

The pretrained BART model achieved a ROUGE-1 F1 score of 0.47, a ROUGE-2 F1 score of 0.36, and a ROUGE-L F1 score of 0.42. In this context, the scores indicate moderate similarity between the generated summaries and the reference. The pretrained BART model seems to be capturing some essential content, indicated by the ROUGE-1 F1 score, along with some of the sentence structures, indicated by the ROUGE-L F1 score but underperforms in capturing consecutive word sequences, indicated by lower ROUGE-2 F1.

The fine-tuned BART model trained with ScisummNet dataset achieved a ROUGE-1 F1 score of 0.50, ROUGE-2 F1 score of 0.35 and ROUGE-L F1 score of 0.42. This increase in ROUGE-1 could imply that the ScisummNet dataset has potentially provided more content or domain specific knowledge that the BART model picked up on, enhancing it capability to recognize and produce accurate summaries.The ROUGE-2 F1 score lightly decreased, however minimal, suggesting potential structural shirts from the dataset that addicted the consecutive word capture.

3.2 TextRank based Summarizer

The TextRank summarizer achieved notably higher ROUGE recall and F1-scores when the generated summaries were compared with the manually written summaries. The difference between ROUGE-1 and ROUGE-2 scores can be explained by the significance of the NGram statistics behind them. ROUGE-1 scores are based on common unigrams i.e. words that are common between the two summaries. Whereas ROUGE-2 scores are based on bigrams i.e. common phrases. Since we are comparing TextRank generated summaries - extractive, sentences from the original text - with the manually written summaries, it is expected that there would be a significant overlap in the keywords level but not beyond as they can be phrased in many different ways. This is further evident in the higer F1 score value in ROUGE-L, suggesting that while the keywords may not be phrased together, they are falling in similar orders in both summaries.

3.3 Clustering based Summarizer

The clustering-based summarization system achieved promising results in terms of ROUGE scores, indicating that it effectively captured the important information from the input text. The ROUGE-1 scores outperformed the ROUGE-2 and ROUGE-L scores. This outcome can be attributed to a higher unigram overlap since keywords and essential terms are more likely to remain consistent across the summaries. The system produced coherent summaries, showcasing the effectiveness of the clustering technique in capturing critical content and keywords from the reference summary, paving the way for potential enhancements. The results demonstrate that the clustering approach with improvements through the incorporation of diversified embedders, a variety of clustering algorithms, and an advanced sentence selection algorithm, can serve as a viable extractive summarization technique with potential applications in various domains.

The following text shows the reference and the generated summaries of a paper (P06-2066[9]) from the ScisummNet dataset:

Reference summary:

```
mildly non-projective dependency structures

syntactic parsing requires a fine balance between expressivity and complexity,
so that naturally occurring structures can be accurately parsed without
compromising efficiency.

in dependency-based parsing, several constraints have been proposed that
restrict the class of permissible structures, such as projectivity, planarity,
multi-planarity, well-nestedness, gap degree, and edge degree.

while projectivity is generally taken to be too restrictive for natural
language syntax, it is not clear which of the other proposals strikes the best
balance between expressivity and complexity.

in this paper, we review and compare the different constraints theoretically,
and provide an experimental evaluation using data from two treebanks,
investigating how large a proportion of the structures found in the treebanks
are permitted under different constraints.

the results indicate that a combination of the well-nestedness constraint and a
parametric constraint on discontinuity gives a very good fit with the
linguistic data.
```

Pre-trained BART Summary:

```
syntactic parsing requires a fine balance between expressivity and complexity
so that naturally occurring structures can be accurately parsed. In
dependencybased parsing several constraints have been proposed that restrict
the class of permissible structures. We review and compare the different
constraints theoretically and provide an experimental evaluation using data
```

from two treebanks. The results indicate that a combination of the wellnestedness constraint and a parametric constraint on discontinuity gives a very good fit with the linguistic data.

Fine-tuned BART Summary:

syntactic parsing requires a fine balance between expressivity and complexity so that naturally occurring structures can be accurately parsed without compromising efficiency in dependencybased parsing several constraints have been proposed that restrict the class of permissible structures such as projectivity planarity multiplanarity wellnestedness gap degree and edge degree while projectivity is generally taken to be too restrictive for natural language syntax. It is not clear which of the other proposals strikes the best balance between the two.We evaluate a subset of them empirically by testing their representational adequacy with respect to two data from two treebanks investigating how large a proportion of the structures found in the treebanks are permitted under different constraints.The results indicate that a combination of the welln

TextRank generated summary:

mildly nonprojective dependency structures

in section 2 we provide a formal definition of dependency structures as a special kind of directed graphs and characterize the notion of projectivity

in section 3 we define and compare five different constraints on mildly nonprojective dependency structures that can be found in the literature planarity multiplanarity wellnestedness gap degree and edge degree

in section 4 we provide an experimental evaluation of the notions of planarity wellnestedness gap degree and edge degree by investigating how large a proportion of the dependency structures found in pdt and ddt are allowed under the different constraints

we write n to refer to the set of positive integers up to and including n

throughout this paper we use standard terminology and notation from graph theory to talk about dependency graphs

in this paper we restrict ourselves to dependency graphs that form forests

this ensures that the extended graph always is a tree

definition 3 a dependency graph is projective if the yields of its nodes are intervals

a gap is a discontinuity in the projection of a node in a dependency graph plátek et al 2001

in contrast to planarity wellnestedness is independent from both gap degree and edge degree in the sense that for every d 0 there are both wellnested and nonwellnested dependency graphs with gap degree or edge degree d all projective dependency graphs d 0 are trivially wellnested

in this section we present an experimental evaluation of planarity wellnestedness gap degree and edge degree by examining how large a proportion of the structures found in two dependency treebanks are allowed under different constraints

Clustering generated summary:

Dependencybased representations have become increasingly popular in syntactic parsing especially for languages that exhibit free or flexible word order such as czech collins et al 1999 bulgarian marinov and nivre 2005 and turkish eryigit and oflazer 2006. In this paper we review a number of proposals for classes of dependency structures that lie between strictly projective and

```
completely unrestricted nonprojective structures. Many practical
implementations of dependency parsing are restricted to projective structures
where the projection of a head word has to form a continuous substring of the
sentence.
```

4 Conclusion

Both pretrained BART and the ScisummNet fine-tuned BART model performed well on the ScisummNet dataset, with the fine-tuned BART model achieving the highest performance of the 3 developed models, an expected outcome of utilizing a pretrained model.

Of the extractive approaches, TextRank performed the best with the higher unigram ROUGE score. Beyond the ROUGE scores, the quality of the summaries generated are fairly comparable and have their own advantages and disadvantages.

The clustering algorithm divides the source text into clusters of sentences and identifies the most important one in each cluster. This promotes summarization based on local context and is very beneficial in structured texts such as research papers. On the other hand, TextRank uses the concept of recommendation to identify significant sentences. The sentences that are highly recommended by other important sentences are added to the summary. This adds value to the summary by condensing the dense information present in research papers.

5 Future work

This project currently stands at a nascent comparison of a few basic summarization algorithms. With additional time and resources, we could expand this work into a more exhaustive survey of text summarization techniques. Particularly, Facebook AI's BART models have proven to provide promising accuracy and can further be improved for specific use-cases through the process of fine-tuning proving the accessibility of relatively performant pretrained models. In the increasingly abstractive-looking world of summarization, thanks to the latest developments in Large Language Models and their applications[8], it could be interesting to explore the relevance and possibilities of extractive summarization as it still holds the advantage of being effective with lesser resources. A promising low-cost text summarizer would be most appreciated, especially in research paper summarization, as the task is highly popular amongst educational institutions and academics, who have limited access to industry-level computing resources.

In particular to the methods presented in this report, we would explore other clustering algorithms, such as DBSCAN or OPTICS, and other sentence embedders, and compare their performance with the existing methods. We would look into other graph based ranking algorithms that can generate more sensible rankings. We would also investigate hybrid methods for selecting representative sentences from each cluster, such as TextRank, density-based or centroid-based approaches, to further improve the summarization quality.

6 Appendix

This project was completed for CS6120 under the instruction of Professor Uzair Ahmad.

Link to project code repository - https://github.com/sumukhig/nlp-text-summarization

7 References

[1] Yasunaga, Michihiro, et al. "ScisummNet: A Large Annotated Corpus and Content-Impact Models for Scientific Paper Summarization with Citation Networks." *ArXiv*, 2019, /abs/1909.01716.

[2] Lewis, Mike, et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.” *ArXiv*, 2019, /abs/1910.13461.

[3] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

[4] S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1–7).

[5] Cohan, Arman, et al. "SPECTER: Document-Level Representation Learning Using Citation-Informed Transformers." *ArXiv*, 2020, /abs/2004.07180.

[6] Mallick, Chirantana et al. “Graph-Based Text Summarization Using Modified TextRank.” *Soft Computing in Data Analytics* (2018): n. Pag.

[7] 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.

[8] Retkowski, Fabian. "The Current State of Summarization." *ArXiv*, 2023, /abs/2305.04853.

[9] Marco Kuhlmann and Joakim Nivre. 2006. Mildly Non-Projective Dependency Structures. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 507–514, Sydney, Australia. Association for Computational Linguistics.