Muhammad Tanveer
Ansh Bhargava
CSCI 49362

NLP Project Final Report

Abstract

Text summarization is a critical task in natural language processing, with two primary methods: abstractive and extractive summarization. This project explores the differences between these approaches by leveraging the PEGASUS and BERTSUM pre-trained models and evaluating their effectiveness using various metrics. The research focuses on summarizing long documents and investigates the performance of abstractive, extractive, and hybrid methods. The project utilizes datasets from CNN/DailyMail news articles and WikiHow instructional guides. A baseline model, selecting the first few sentences as the summary, is established for comparison. Results show that the baseline performs well due to the salient information being presented early in these domains, but caution is needed when generalizing to other contexts. Future work involves testing in different domains and on longer texts to enhance method effectiveness.

Introduction

Text summarization is an important task in natural language processing. There are primarily two methods of summarization: abstractive and extractive. Abstractive summarization involves creating a summary of a text which captures the essential information and may include phrases that were not present in the source text to do so. On the other hand, extractive summarization creates a summary only by extracting ‘important’ phrases or sentences from the source text.

Our project is tasked with exploring the differences between the two approaches of text summarization by leveraging the pre-trained models PEGASUS and BERTSUM. We will also explore the effectiveness of hybrid summarization methods using both abstractive vs extractive using a variety of evaluation metrics.

We believe this is an interesting focal point because the model architectures and training for these approaches differ vastly and are both active areas of research. Summarization is also a key task within natural language understanding and we hope to glean a better understanding of these two fundamentally different approaches. Further, text summarization is an important task in its own regard because it can reduce the amount of time required to process information, improve the speed of searching for information, and make learning a topic easier.

To this end, we want to explore the efficiency of a range approaches of summarizing long documents:

1. Using a pre-trained, publicly available abstractive summarization model to directly summarize the long text.
2. Using a pre-trained, publicly available extractive summarization model to directly summarize the long text.

3. Various hybrid approaches which leverage the prior two. We expand on each in later sections, but in summary these are: simple two-step, length weighted two-step, graph methods, ensemble methods, hierarchical and iterative methods.

Our research question asks which approach performs better, as measured by core summarization metrics of ROUGE, readability and entity references.

Previous Research

We highlight relevant prior research and suggest our extension to the literature.

"Extractive Summarization of Long Documents by Combining Global and Local Context" by Wen Xiao and Giuseppe Carenini proposes a model for summarizing long texts using both local and global context. It works particularly well at section level, though redundancy issues are present and it could benefit from more intricate discourse structure integration.

"PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization" by Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu introduces PEGASUS, a top-performing language model. It excels in abstractive summarization and adapts well to tasks with limited examples. However, its performance can drop with longer documents due to finite context windows and lack of document structure consideration.

"A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents" by Arman Cohan et al. presents a model incorporating document structure into abstractive summarization. Despite its innovative approach, the evaluation metric (ROUGE) may overlook factual consistency, and the model doesn't fully address the global context issue in Transformer models.

Our proposed project aims to review these latest models in summarization, particularly focusing on their performance with longer texts.

Data and Baseline

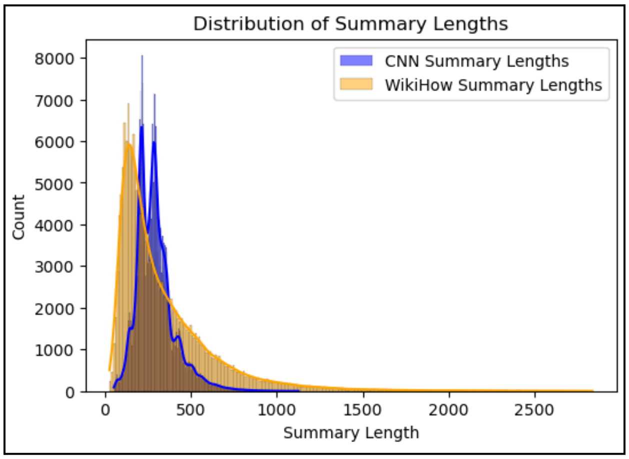
We use two datasets for this project, the CNN/DailyMail news article dataset and the WikiHow instructional guide dataset. These datasets are described further below.

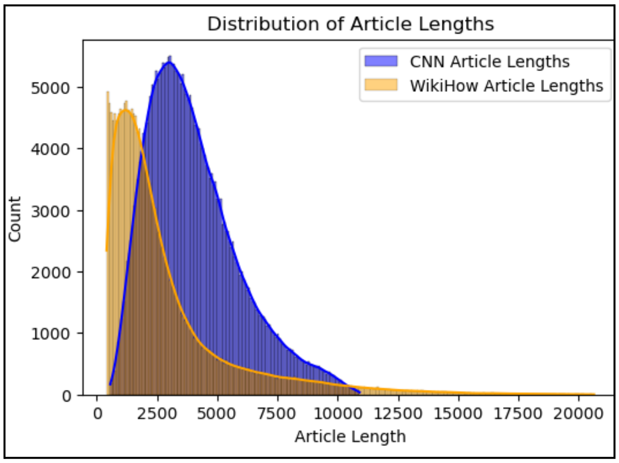
CNN/Dailymail (Source):
This is an English language dataset with over 300K unique news articles gathered from CNN and DailyMail publication sources. Each instance in the dataset contains the following features: ID (hex hash article source URL), article (string containing the body of the news article), highlights (string containing summary of article).

WikiHow (Source):
Large scale dataset sourced from WikiHow knowledge base. Each article is made up of multiple paragraphs with each paragraph starting with a sentence that summarizes the paragraph. The summary is constructed as a concatenation of the first sentence of each sub-section (each instruction).

We provide some basic summary statistics and distribution plots for these datasets:

Dataset Statistic	CNN/Dailymail	WikiHow
Number of article-summary pairs	311820	181925
Avg Length of Article	4018	2885
Avg Length of Summary	293	347
Summary to Article Ratio	0.088	0.204





Baseline

Our baseline model for summarization was simply to select the first k sentences from the article as our summary. We also didn't add sentences if they were less than three words long. This was done in an attempt to minimize boilerplate sentences such as "By Associated Press" or "PUBLISHED" from being part of our summary. We implemented the baseline model for k between 2 and 6. Our final baseline model selects *k=5*.

We chose this approach for its simplicity and because of the intuition that texts are structured to include salient information early on to hook readers. We anticipate the extent to which this is true to be domain specific.

Method

We implemented and tested the following summarization methods, from the base models to two hybrid summarizers.

Abstractive Summarizer

For this project we used the pre-trained abstractive summarizer PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization). This is a self supervised model that is trained to produce an abstracted summary of text from an input text that is masked. When tasked with abstractive summarization, it performs really well on major metrics. However since it is a transformer based model, it has a finite context window and therefore loses performance over longer documents. This model also does not take into account the document's structure or sectioning which may impact its performance.

Extractive Summarizer

For the extractive summarizer we opted to use the BERTSUM (Bidirectional Encoder Representations from Transformers) model. This pre-trained model uses words in both the previous and next context of a target word to create word embeddings. It allows for masked inputs to be detected as well as relations between two concurrent sentences. Concurrent sentences are rated based on their relation to one another, from completely related to not related at all. From there BERTSUM builds on BERT by classifying whether or not sentences belong in the summary i.e. their level of importance in relation to the text.

Graph Summarizer

This method first converts the input text into a similarity matrix using TF-IDF vectorization and cosine similarity. The TextRank algorithm is then applied to this matrix to create a graph representation of the sentences, where nodes represent sentences and edges represent the similarity between them. The importance scores of each sentence are calculated using the PageRank algorithm. The top-ranked sentences are selected as the most important ones and are used as input for an abstractive summarization function to generate a final summary.

Two key features of this are:

1. By modeling the sentences as nodes in a graph and considering the similarity between them, the method captures the relationships and contextual information within the text, resulting in more coherent and relevant summaries.
2. The graph-based approach can handle large amounts of text efficiently, as the calculations are based on matrix operations and graph algorithms, which are well-suited for parallelization and optimization.

Ensemble Summarizer

This method leverages both extractive and abstractive summaries, ranking sentences in their outputs to generate a final summary. Specifically, it first extracts all sentences from both summaries and removes duplicates. Then, similar to the graph summarizer, we use PageRank to score each sentence (across both summaries) and finally return the top (n=5)sentences with the highest PageRank scores.

Two key features of this are:

1. Comprehensive Information Coverage: By combining extractive and abstractive summarization techniques, the ensemble approach aims to capture the strengths of both methods. Extractive summarization ensures that important sentences from the original text are included, preserving factual accuracy. Abstractive summarization, on the other hand, can generate more concise and coherent summaries by rephrasing and paraphrasing the content.

2. Customizability and Flexibility: The ensemble approach allows for flexibility and customization by incorporating multiple summarization techniques. It enables the use of various extractive and abstractive algorithms, giving the ability to adapt to different text types, lengths, and summarization requirements. This flexibility allows for better optimization and tailoring of the summarization process to specific use cases or preferences.

We also implemented the following methods, but weren't able to test fully due to compute constraints: two-step hybrid summarization, length weighted hybrid summarization, hierarchical summarization, iterative summarization. Our appendix contains a description of these and the submitted notebook contains their implementation.

Metrics

The following metrics were evaluated for 30,000 articles from both sources.

ROGUE-1, ROGUE-2, ROGUE-L Score

ROGUE (Recall-Oriented Understudy for Gisting Evaluation) measures the number of matching n-grams between the generated summary and the reference summary for each model. ROGUE-1 and ROGUE-2 measure similarities based on unigrams and bigrams respectively. ROGUE-L measures the similarity between two texts based on the longest common subsequence (LCS) that two texts have in common. This metric was important to determine how similar the summaries we generated were to the reference summaries provided in the dataset.

Flesch Readability Index

The Flesch Readability Index returns a score of how readable the generated summary is. The readability score ranges from 0 to 100, 0 correlating to reading at the professional level (extremely difficult to read) and 100 correlating to reliability of a 5th grader (very easy to read). The readability score provides an insight into the usefulness of the generated summary and helps us determine which method of summarization might be more usable than others

Entity Grid Score

The Entity Grid Score provides a local coherence between the text and the generated summary. It returns a score that dictates how many of the entities mentioned in the text appear in the generated summary. This is helpful especially in news articles which more often than not discuss current events in which entities are some of the most important aspects of the text itself.

Jaccard Index Similarity

The Jaccard Index Similarity is a measurement of the shared tokens between two texts. The range for the index ranges from 0 to 1, 0 being no common words between the two texts and 1 being identical. Although the ROGUE score serves a similar function the Jaccard Index helps provide an idea of how many words are similar between the generated summary and the reference summary, as opposed to the ROGUE score which informs the n-gram similarity between two texts.

Results and Evaluation

CNN/ Daily Mail

	ROUGE-1	ROUGE-2	ROUGE-L	Readability	Entity Grid Score	Jaccard Index
Baseline	34.2	14.2	21.8	60.5	**0.488**	**0.193**
Abstractive	29.2	9.13	19.4	51.6	0.318	0.093
Extractive	**40.4**	**18.8**	**28.3**	**68.3**	0.382	0.105
Graph	28.9	8.00	18.6	51.9	0.352	0.090
Ensemble	36.0	15.9	22.0	64.5	0.477	0.149

WikiHow

	ROUGE-1	ROUGE-2	ROUGE-L	Readability	Entity Grid Score	Jaccard Index
Baseline	24.3	**5.47**	14.7	70.0	0.208	**0.299**
Abstractive	21.1	3.99	14.2	58.3	0.178	0.086
Extractive	**25.1**	5.43	**15.8**	**76.1**	0.238	0.174
Graph	22.2	4.54	15.0	59.4	0.174	0.091
Ensemble	24.2	5.44	14.3	73.6	**0.255**	0.193

Conclusion

In conclusion, our baseline method has demonstrated remarkable performance relative to most other methods employed in our study. It is tempting to conclude that our machine learning models are not as capable in comparison. However, it is important to consider the specific domains we tested, namely news articles and how-to guides, as they are designed to encapsulate the most salient information within the first few sentences. Consequently, the baseline's superior performance in these domains may not necessarily translate to other subject areas.

Furthermore, the success of extractive models in our study may be attributed to the nature of the data. Selecting key phrases and sentences from the source text aligns well with the structure of news articles and how-to guides, where certain phrases are deliberately intended to be "key."

However, this may not hold true in many other domains, where the main ideas and connections between sections play a more significant role in conveying information.

Therefore, while the baseline's strong performance and the success of extractive models in our specific data are noteworthy, caution should be exercised when extrapolating these results to other domains.

Future Work

In the future, there are several avenues of research that can be explored to further enhance the effectiveness of our methods. One important direction is to test the applicability of our approaches on other source domains where the content structure differs from news articles and how-to articles. This includes domains such as medical reports, legal briefs, and financial documents, where the key information may not always be captured in the first few sentences. By adapting our methods to handle these variations, we can expand the utility of our techniques in a broader range of contexts.

Another area for future work is to test the effectiveness of our methods on longer texts. While our current focus has been on news articles, it would be valuable to evaluate how well our approaches perform on lengthier documents. Longer texts often present additional challenges in information extraction and summarization, as they contain more nuanced and intricate details. By investigating the performance of our methods on longer texts, we can assess their scalability and identify potential areas for improvement.

References

1. “Extractive Summarization of Long Documents by Combining Global and Local Context” - Wen Xiao, Giuseppe Carenini
2. “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization” - Jingqing Zhang, Yao Zhao, Mohammad Saleh, Peter J. Liu
3. “A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents” - Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, Nazli Goharian

Appendix

A. Other Summarizer Methods

Simple Two-Step Hybrid Summarization

This method implements a hybrid extractive-abstractive approach for text summarization using BERTSUM (extractive) and PEGASUS (abstractive) models. The approach consists of two steps to generate the final summary.

In the first step, the input text is divided into equal parts based on the specified number of steps. Each part represents a segment of the original text. Then, in the second step, extractive summarization is performed on each part using the BERTSUM model, producing extractive summaries for each segment. These extractive summaries aim to condense the content of each part.

Next, in the third step, the extractive summaries are concatenated, forming a cohesive representation of the entire text. This concatenated extractive summary is then passed to the PEGASUS model for abstractive summarization. The PEGASUS model generates the final abstractive summary by paraphrasing and rephrasing the information from the concatenated extractive summaries.

Two key advantages of this hybrid approach are:

1 The hybrid approach directly combines the strengths of both extractive and abstractive methods. The initial step of extractive summarization using BERTSUM provides a condensed representation of the original text, highlighting important information and maintaining factual accuracy. The subsequent abstractive summarization using PEGASUS enhances the summary by generating a concise and coherent version, leveraging the power of language generation models.

2 Improved Coherence and Comprehensiveness: By dividing the text into segments and performing extractive summarization on each part, the hybrid approach

improves the coherence and comprehensiveness of the final summary. Extracting summaries for each segment allows capturing the key points from different parts of the text while maintaining their contextual relevance. This approach helps to ensure that important information from different parts of the text is included in the final summary.

Length Weighted Two-Step Hybrid Summarization

This method implements a hybrid extractive-abstractive approach for text summarization, with a focus on extracting and prioritizing the "most important sentences" from the extractive summary. It is very similar to the simple two-step method above, with the main difference being that only “important” sentences are passed as input to the abstractive summarization model to generate the final summary. The importance of a sentence is simply determined by its length, which may be a reasonable proxy for the output of an extractive summarizer.

Thus the additional key benefit of this relies on length actually being a useful proxy for importance, in that longer sentences contain more essential details. If so, this method extends the previous by helping ensure that the extractive summary focuses on the most relevant and significant content.

Hierarchical Summarization

This method is also very similar to the two-step hybrid summarizer, with the main differences being that it only uses an abstractive summarizer and it attempts to divide the source text by natural sections as opposed to equal breakpoints. Specifically, we first separate the text into meaningful sections that can be summarized independently. Each section is stripped of leading and trailing whitespace before being returned as a list of sections. Then, for each section, an abstractive summary is generated. These section summaries represent condensed versions of the individual sections. Finally, the section summaries are concatenated into a single text, which is then passed to the abstractive summarizer again to generate the summary of summaries. This step aims to create a higher-level summary that captures the key points from the individual section summaries, providing an overall summary of the entire text.

A key advantage of the hierarchical abstractive summarization method is its ability to capture and summarize information at multiple levels of granularity. By dividing the text into sections and generating summaries for each section, the method can provide detailed and focused summaries for individual sections.

Iterative Summarization

This method implements an iterative abstractive summarization method. We first take the input text and merge similar consecutive sentences based on their similarity, using

TF-IDF vectorization and cosine similarity between each pair of consecutive sentences. If the similarity exceeds a specified threshold, the sentences are merged into a single sentence. This step helps to reduce redundancy and improve the cohesion of the summary. Then we remove short sentences, aiming to filter out short and potentially less informative sentences from the summary. We do the previous two steps iteratively for a specified number of iterations.

A key advantage of this iterative abstractive summarization method is its ability to refine and improve the summary over multiple iterations. Each iteration involves generating an abstractive summary, merging similar sentences, and removing short sentences. This iterative process allows for the progressive refinement of the summary, reducing redundancy, improving coherence, and filtering out less informative sentences.

Query-Based Summarization

This method implements a query-based summarization method which aims to generate summaries that are tailored to a specific query. We first identify the most relevant sentences in the input text based on a given query by calculating the similarity between the query and each sentence using TF-IDF vectorization. The sentences with the highest similarity scores to the query are selected as the most relevant sentences. Then, we generate the final summary by applying an abstractive summarization method over these selected relevant sentences.

A key advantage of this query-based approach is its ability to generate summaries that directly address the information needs expressed in the query. By considering the query during the summarization process, the method can identify and prioritize the sentences that are most relevant to the query. This enables the generation of highly focused and targeted summaries that specifically address the query's context and information requirements.

Team Member Responsibilities

Muhammad Tanveer - data cleaning and preprocessing, creating and implementing the methods for summary generation, and creating the final report with results.

Ansh Bhargava - tasked with creating an outline for the project, the evaluation metrics we will be using, performing tests, and evaluating based on metrics selected.