SummIt!: An in-depth analysis of 4 automatic text summarization methods

Nam Ho Koh
University of Michigan
namhokoh@umich.edu

Arshdeep Singh
University of Michigan
arshdp@umich.edu

Joe Plata
University of Michigan
joeplata@umich.edu

Abstract

In recent years, there has been an increase in the amount of text information available online. This widely accessible data is undoubtedly an invaluable source of information and knowledge which may benefit significantly from effective text summarization (Awasthi et al., 2021). This paper aims to provide an in-depth analysis of 4 different summarization techniques ranging from the simplest, such as the frequency-driven approach to more complex extractive methods with transformers. We believe that by providing this analysis, we hope to convey the importance of summarization and a clear depiction of how each method performs against the same dataset, thus informing the readers on which method could be used for their own summarization task. The paper's outline is as follows: Introduction, Related works, Approach, Evaluation, Results & Discussion, Conclusion and Division of work.

1 Introduction

With the growing ubiquity of devices capable of connecting people to the internet and the resulting ease of both creating and consuming content that is available to the whole world at a moment's notice, the quantity of information on the internet is increasing rapidly. The constant stream of content has resulted in an environment of **information overload**, in which the presence of *too much* information may hinder the ability to understand an issue or effectively make decisions.

From politicians and business leaders requiring up-to-date knowledge to inform their strategic initiatives to the average media consumer wanting to stay informed of current events, the necessity of effectively and efficiently identifying and extracting essential information from detailed, extensive sources is imperative. For most consumers of information on the internet, a comprehensive understanding of a particular subject (e.g., a news article or Wikipedia entry) is rarely necessary, and a brief description of the critical points is sufficient. Hence, a reliable summary would provide the consumer with crucial information whilst avoiding less relevant ones. In addition, this would increase the likelihood that the source material is interpreted as intended and give consumers more time to research additional areas of interest, ultimately allowing them to gain further pertinent knowledge.

As such, our team hopes to accomplish this by performing research in text summarization - a subset of natural language processing that focuses on generating a concise and precise summary of voluminous texts while preserving the overall meaning. In particular, we are focusing on the extractive method of text summarization, which seeks to summarize the text by choosing a subset of the most relevant sentences from the original text. We hope to contribute to this research by comparing the performances of different extractive text summarization approaches, which we discuss in the following section.

2 Related works on Extractive Summarization

Recent research on extractive summarization expands upon a wide variety and diversity of approaches. Extractive summarization is a type of text summarization where the summary is generated by selecting and extracting important sentences or phrases from the original text. There have been various approaches to extractive summarization, and some of these approaches are discussed in the following studies:

- (Chen et al., 2017) solved the problem of generalization and the inability of the model to use the source text by improving the decoder in the encoder-decoder neural summarization model.
- (Wu and Liu, 2003) compared two methods for article summarization. The first method is

based on term frequency, and the second approach is based on the ranking of paragraphs based on their relevance to the main topics.

- (Allahyari et al., 2017) Details the benefits of introducing text summarization capabilities and provides a survey of potential techniques that could be adopted for the purpose of text summarization, including frequency-driven approaches, latent semantic analysis, and transformer-based methods, as well as metrics that might allow us to evaluate the performance of our implementations. This paper provided motivation for us to compare some of the methods outlined by the authors and perform a critical comparison of the strengths and weaknesses of each approach and how they perform relative to one another.

- In their paper "A survey of text summarization techniques" (Nenkova and McKeown, 2012) provides an overview of the different techniques that have been used for extractive summarization. They discuss rule-based approaches, which rely on pre-defined rules for identifying and extracting meaningful sentences, and statistical methods, which use algorithms to automatically identify and select essential sentences.

- Another approach to extractive summarization is the use of topic modelling, which involves identifying the main topics in a text and then selecting sentences that are relevant to those topics. This approach is discussed in a paper by Fabbri et al. (2019) (Fabbri et al., 2019), who propose a model for extractive summarization that combines topic modelling with sentence scoring.

- A different approach to extractive summarization uses a combination of different techniques, such as rule-based and statistical methods. This approach is discussed in (Sarkar, 2013), where the authors propose a hybrid system for extractive summarization that combines a rule-based approach with a statistical method called Latent Semantic Analysis.

- Transformer-based methods are a neural network-based approach to extractive summarization. These methods use transformer architectures, a type of deep learning model that has been shown to be adequate for various natural language processing tasks. In the context of extractive summarization, transformer-based methods can be used to automatically identify and select essential sentences or phrases from a given text. One example of a transformer-based approach to extractive summarization is the model proposed by (Singh et al., 2017). The authors use a transformer architecture in this model to generate a summary by selecting essential sentences from the input text. They also use a pointer network to allow the model to copy words directly from the input text, which can help to improve the coherence of the generated summary.

- Another example of a transformer-based approach to extractive summarization is the model proposed by (Xu et al., 2020). In this model, the authors use a hierarchical transformer architecture, allowing the model to capture global and local contextual information from the input text. They also use a pointer network to allow the model to copy words directly from the input text.

  In addition to these approaches, some researchers have also explored using neural networks for extractive summarization. For example, in a paper by Nallapati et al. (Nallapati et al., 2017) (2016), the authors propose a neural network-based model for extractive summarization that uses a combination of convolutional and recurrent neural networks.

  Overall, there are various approaches to extractive summarization, each with its strengths and limitations. As a result, researchers have continued to explore new methods for improving the performance of extractive summarization algorithms.

3 Approach

This section will discuss four approaches to text summarization that will be explored as part of this project.

3.1 Frequency-driven approach

The main essence of the frequency-driven approach is assigning a binary weight (0 or 1) to a word that is more correlated to a particular topic. The two most common techniques in this category are:

word probability and TFIDF (Term Frequency Inverse Document Document Frequency) (Sreenivasulu et al., 2022).

- Word probability. Word probability is one of the simplest methods of utilizing the frequency of words as the main indicator of importance. The probability of a word w is determined by the number of occurrences of the word, f(w), divided by the total number of all words in the input (this may be a single document or multiple):

$$P(w) = \frac{f(w)}{N} \quad (1)$$

  Therefore, sentences containing the most frequent words in a document stand a higher chance of being selected for the final summary. The assumption is that the higher the frequency of a word in a text, the more likely that it indicates the subject of the text (Allahyari et al., 2017).

- TFIDF. TFIDF (Term Frequency Inverse Document Frequency) is a more advanced method of assigning word weights. This weighting technique evaluates the importance of words. It identifies the most common words, which will be omitted from the evaluation, on the document(s) by giving lower weights to words appearing most frequently in most documents. The weight of each word w in document d is represented as follows:

$$q(w) = f_d(w) * \log \frac{|D|}{f_D(w)} \quad (2)$$

  Where $f_d(w)$ is the term frequency of word w in document d, $f_D(w)$ is the number of documents that contain word w and $|D|$ is the number of documents in the total collection D. In essence, if there are "specific words" in a given sentence, then the sentence relatively holds more weight and importance (Allahyari et al., 2017).

3.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA), first introduced by (Deerwester et al., 1990), is an unsupervised method for extracting a representation of text semantics based on observed words. Gong and Liu (Gong and Liu, 2001) first proposed a method of utilizing LSA to select highly ranked sentences for single and multi-document summarization in the news domain. LSA first builds a term-sentence matrix (nxm), where each row corresponds to a sentence (m). Each entry a_{ij} of the matrix is the weight of the word i in sentence j. The weights of the words are computed by TFIDF technique and if a sentence does not have a word the weight of that word in the sentence is zero. Then singular value decomposition (SVD) is used on the matrix and transformed the matrix A into three matrices $A = U\Sigma V^T$. Matrix U represents a term-topic matrix having weights of the words. Matrix Σ is a diagonal matrix where each row i corresponds to the weight of a topic i. Matrix V^T is the topic sentence matrix.

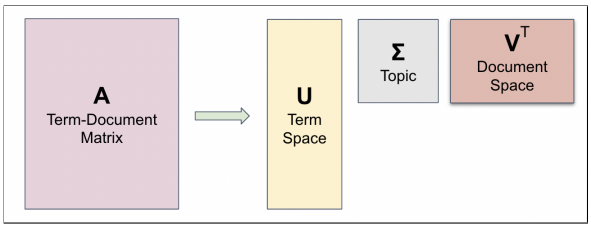

Figure 1: Singular Value Decomposition to Term-Topics-Document Matrices

The matrix $D = \Sigma V^T$ represents how much a sentence represents a topic. Thus d_{ij} shows the weight of the topic i in sentence j (Gong and Liu, 2001; Merchant and Pande, 2018). As the term matrix, U gives the relevance of the words to the topics in Σ matrix and as we also computed the relevance of the documents belonging to each of the topics we can estimate the relevance of the word to each of the documents, i.e. sentences in the text. In this way, we can get the weights of the query word belonging to each document. Summarization is done by selecting the top K sentences in each topic depending on the weightage. Also, the documents are sorted according to semantic similarity.

3.3 Neural Extractive Summarization

3.3.1 Transformers

A recent breakthrough in natural language processing was marked by Google's announcement of a seminal language representation model known as BERT (Bidirectional Encoder Representations from Transformers). This model yielded state-of-the-art results on 11 different NLP tasks (Vaswani et al., 2017) and has been extensively adopted and studied by researchers. The general architecture of BERT is shown in figure 2. The input text is

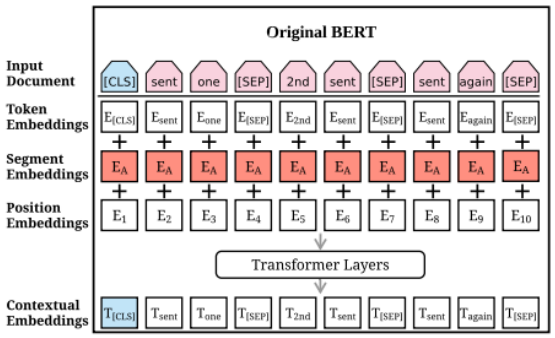

Figure 2: Architecture of original BERT (Zhang et al., 2019).

first preprocessed by inserting two special tokens. [CLS] is appended to the beginning of the text; the output representation of this token is used to aggregate information from the entire sequence (i.e. classification tasks). Furthermore, a [SEP] token is inserted after each sentence to indicate sentence boundaries. The modified text is then represented as a sequence of tokens $X = [w_1, w_2, ..., w_3]$. Each token w_i is assigned three kinds of embeddings (Zhang et al., 2019; Liu and Lapata, 2019):

- Token embeddings represent the meaning of each token.
- Segmentation embeddings are used to discriminate between two sentences.
- Position embeddings indicate the position of each token within the text sequence.

These three embeddings are summed to a single vector x_i and fed to a bidirectional Transformer with multiple layers.

3.3.2 Clusterization and Summarization

Transformers have wholly rebuilt the landscape of natural language processing (NLP). Before transformers, we had okay translation and language classification thanks to recurrent neural nets (RNNs) — their language comprehension was limited and led to many minor mistakes, and coherence over larger chunks of text was practically impossible. Since the introduction of the first transformer model in the 2017 paper 'Attention is all you need' (et al., 2017), NLP has moved from RNNs to models like BERT and GPT. One of the most widely used of these pre-trained models is BERT or Bidirectional Encoder Representations from Transformers by Google AI. BERT spawned a host of further models and derivations such as distilBERT, RoBERTa, and ALBERT, covering tasks such as classification, Q&A, POS-tagging, and more. So far, so good, but these transformer models had one issue when building sentence vectors: Transformers work using a word or token-level embeddings, not sentence-level embeddings.

SBERT outperformed the previous state-of-the-art (SOTA) models for all common semantic textual similarity (STS) tasks. SBERT produces sentence embeddings — so we do not need to perform a whole inference computation for every sentence-pair comparison. Finding the most similar sentence pair from 10K sentences took 65 hours with BERT. With SBERT, embeddings are created in 5 seconds and compared with cosine similarity in 0.01 seconds using the below formula.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The fastest and easiest way to begin working with sentence transformers is through the sentence-transformers library created by the creators of SBERT. This gives you an N-dimension sentence vector, aka. Sentence embedding can be used for the text summarization use case.

Once a sentence is transformed into a vector, various operations can be performed on it. It is important to note that the accuracy of sentence embedding plays a significant role. Each sentence can be clustered into k groups based on the cosine distance, which is the bedrock of a KMeans clustering algorithm. In subsequent steps, we update the centroid of the clusters and label them 1 through 10. These clusters represent the topic of our text, and the closest sentence based on the cosine distance is selected as the most relevant sentence to our article. Thus clusterization gives us reliable and effective ways of extractive article summarization.

3.3.3 MatchSum

Most extractive summarization systems score and extract sentences or smaller text units one by one from the original text, model the relationship between the sentences, and then select several sentences to form a summary (Xu and Durrett, 2019). Given this relationship, an extractive summarization task could be formulated as a sequence labelling task which explains the wide adoption of

encoder-decoder frameworks to perform these tasks (Cheng and Lapata, 2016; Nallapati et al., 2017). However, these approaches introduce redundancy, where the models make independent binary decisions for each sentence. In addition, removing this redundancy through methods such as Trigram blocking has yielded performance improvements on the CNN/DailyMail dataset (Paulus et al., 2017).

The approaches aforementioned in the previous subsection may be viewed as sentence-level extractors as, instead of considering the semantics of the entire summary, it is modelling the relationship strictly between sentences. As a result, these methods will be inclined to select highly generalized sentences while ignoring the coupling of multiple sentences. Advanced summarization techniques involving reinforcement learning may address these shortcomings (Narayan et al., 2018) yet still perform sentence-level extractions. Therefore, we have decided to investigate *MatchSum*, a novel summary-level framework introduced by (Zhong et al., 2020), which conceptualizes the summarization problem as a semantic text-matching problem. Semantic matching refers to estimating semantic similarity between a source and a target text fragment.

The architecture of MatchSum is shown below in Figure 3

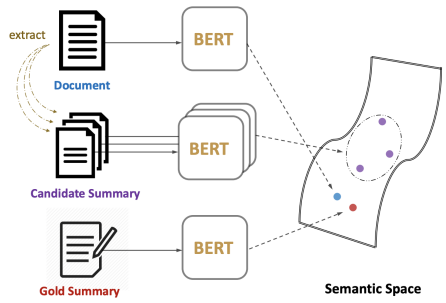

Figure 3: Architecture of MatchSum (Zhong et al., 2020). Better candidate summaries would be semantically closer to the document, while the gold summary should be the closest.

A Siamese-BERT architecture has been used to compute the similarity between the source document and the candidate summaries. The primary motivation outlined by the authors for using such architecture was that the Siamese BERT leverages the pre-trained BERT (Devlin et al., 2018) to derive semantically salient text embeddings that can be compared using the cosine-similarity. Ultimately, a pertinent summary will yield a high summary amongst a set of candidate summaries (Zhong et al., 2020). For our implementation, the alternative `RoBERTa-base` model was used due to its robust architecture and optimizations in fine-tuning/training (Liu et al., 2019).

4 Evaluation

Evaluation for summary is a challenging task as there is no ideal summary for a document or collection of documents. In addition, the definition of a good summary is an open-ended question (Saggion and Poibeau, 2013) open to subjectivity. It has also been found that even human evaluators have a low agreement for evaluating and producing summaries. This may be attributed to the fact that individuals may have different semantic distributions from each other stemming from the differences in language acquisition and culture. Moreover, the lack of standard evaluation metrics has caused summary evaluation to become a challenging and complex task to accomplish.

4.1 Human Evaluation

One of the simplest ways of evaluating a summary is to have a human assess its quality and validity. For instance, in DUC, the judges would evaluate the coverage of the summary, i.e. how much the candidate summary covered the originally given input (Saggion and Poibeau, 2013).

4.2 Automatic Evaluation Methods

Fortunately, there has been a set of automatic evaluation metrics for summary tasks since the early 2000s. ROUGE is one of the most widely used metrics for automatic evaluation.

4.2.1 ROUGE

Lin (Lin, 2004) introduced a set of metrics called Recalled Oriented Understudy for Gisting Evaluation (ROUGE) to automatically determine the quality of a summary by comparing it to human (reference) summaries. Several variations of ROUGE include ROUGE-n, ROUGE-L, and ROUGE-SU (Lin, 2004). For this project's scope, we will evaluate each text summarization approach on 2 ROUGE-N variants: ROUGE-1, ROUGE-2 and ROUGE-L. ROUGE-N will assess informativeness; in contrast, ROUGE-L will evaluate the fluency of the summary. Below are the comprehensive details about the importance and intuitive inference of the evaluation metrics used in this experiment.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

ROUGE-1 refers to the overlap of the unigrams between the reference summary and the system summaries. Precision, Recall and F-score are computed using the number of matched unigrams between the two texts that quantify the amount of information captured by the model compared to the gold standard reference summary available. ROUGE-2 is similar to ROUGE-1 but captures the overlap of bigrams between the reference and system-generated text. ROUGE-L is particularly interesting as it reflects how fluent the predicted text is compared to the original text. ROUGE-L considers the longest common sub-sequence within the two texts and then calculates the precision, recall and F-scores based on the resulting counts.

4.3 Dataset

The CNN/DailyMail dataset (Hermann et al., 2015) was used during the evaluation of this paper. This dataset is a collection of nearly 300k articles from CNN and Daily Mail and questions associated with each article (although we will not use the related questions). This dataset benefits our work as it contains a large number of articles for us to work with. The type of text it contains (articles from popular online news sources) is directly relevant to our motivation of allowing people to stay up-to-date with current events.

Each news article included in the dataset contains both the entirety of the article itself as well as a three-sentence summary of the article that was created by human evaluators. Given the aforementioned difficulties of evaluating text summarization, being provided with an extensive set of articles with human-generated, ground truth values for training and testing helps not only improve the performance of our models but also to be more confident that our evaluation metrics are truly based on the ability to summarize text for human consumption. One thing to note about these provided summaries, however, is that they are abstractive summaries rather than extractive - meaning that they are derived from the evaluators' understandings of the articles rather than extracted from the texts of the articles directly, which is what we seek to accomplish in this project. Although the method used to create the ground truth summaries differs from the methods used in this paper, we still believe this dataset is appropriate for the project and that the difference should not materially impact the results.

5 Results & Discussion

5.1 Results and Interpretation of models

TABLE 1: F-SCORE

Model	R-1	R-2	R-L
Frequency-based	19 %	4 %	16 %
LSA	9 %	1 %	8 %
SD-KMeans	24 %	8 %	22 %
MatchSum	43 %	20 %	39 %

TABLE 2: PRECISION

Model	R-1	R-2	R-L
Frequency-based	15 %	3 %	13 %
LSA	16 %	1 %	15 %
SD-KMeans	34 %	12 %	32 %
MatchSum	37 %	17 %	34 %

TABLE 3: RECALL

Model	R-1	R-2	R-L
Frequency-based	25 %	6 %	20 %
LSA	6 %	1 %	6 %
SD-KMeans	20 %	6 %	18 %
MatchSum	54 %	25 %	29 %

As shown in the result tables above, the MatchSum nearly outperforms all approaches in the *f-score*, *precision* and *recall*. MatchSum yielded an R-1 F-score of 43%, R-2 score of 20% and an R-L score of 39% matching the similar output outlined in (Zhong et al., 2020). An F-score is often hard to interpret, but it is viewed as a harmonic mean between the precision and the recall. Precision, in this instance, is measured as: $\frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_summary}$. Therefore, we can claim that MatchSum was able to yield a higher score on the unigram, bigram and LCS overlap compared to other models.

Another critical thing to observe in this experiment is the performance of SDKMeans against the LSA analysis. SDKMeans performs better than the LSA, which can be explained due to its power to get sentence embeddings, which boosts its scores. Both methods try to capture and work with the same principle behind them. LSA tries to capture the most critical topics using the SVD technique, whereas SDKMeans tried to find the centroid in a cluster using sentence embeddings and machine learning methods. In a sense, they try to capture a similar phenomenon but use different methodologies. Once the topics or centroids are found, sentences are combined together based on their relevance to generate the summary. The working of both algorithms is exciting and fascinating in how they achieve their goals. The building block of the clusterization method is the sentence embeddings generated using the sentence transformer (SBERT) that makes it stand out and perform far superior than the LSA model. Having accurate sentence embeddings makes the clustering method sufficiently correct to group the sentences together with similar meanings. The closest sentence to the centroids computed gives many precise and accurate sentences for our extractive summarization compared to the LSA method, where we are extracting the sentences merely based on the weightage of the topic in the Σ matrix. This explains the superior performance of SDKMeans over the LSA method working on the same idea yet different approaches.

Interestingly, despite the frequency-based method being the simplest approach among others, it still outperformed more complex architecture, such as LSA on the F-Score and Recall. For the latter, this means that 25% for R-1, 6% for R-2 and 20% for R-L of the n-grams in the reference summary were also present in the generated summary from the frequency-based model. Moreover, for the former, the frequency-based method yielded an overall higher f-score over LSA beating our previous expectations.

5.2 Performance Relative to Article Length

The above results evaluated the average ROUGE scores obtained via each method for the entire dataset. While this evaluation provides insight into how these methods perform relative to one another for any given article, it gives no information about how performance is affected by the article itself. As such, we decided to explore this question by evaluating the performance of one method, the frequency-based approach, broken out by the number of sentences making up the articles. To perform this comparison, we separated articles into six buckets based on the number of sentences, with 11-25 and 26-40 sentences being the most common, making up 36% and 28%, respectively, and the tails of 10 or fewer and 71 or more sentences combining for a little more than 10% of the dataset.

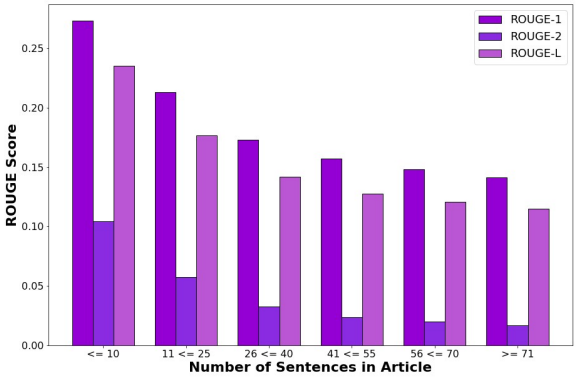

Figure 4: ROUGE scores achieved by the number of sentences in articles.

As we summarized all articles in three sentences regardless of length, we anticipated the ROUGE scores being higher for articles with fewer sentences as the summaries are able to contain a more significant proportion of the total article. This hypothesis was correct, as shown in Figure 4. All ROUGE scores were highest for articles with ten or fewer sentences and gradually decreased as the number of sentences increased, starting to show only marginal decreases as the number exceeded around 50. This decrease possibly could have been mitigated if, instead of summarizing all articles using three sentences, we selected a constant percentage of the total sentences to create the summary. Still, we elected to use our approach in order to stay consistent with our mission of providing brief, easily digestible summaries for all articles.

5.3 Comparison against ChatGPT-3

One of the most recent advancements in natural language processing was marked by the release of ChatGPT by OpenAI (OpenAI, 2022). ChatGPT is a high-performance dialogue agent trained to interact in a conversational way. Given that GPT-3 (Generative Pretrained Transformer 3) is a state-of-the-art language model also developed by OpenAI trained on nearly 175 billion parameters, it is capable of a wide range of applications such as language

translation, linguistic modelling and many more.

One of the avenues of research we decided to take was to compare the performance of such a powerful language model against a sample from our dataset to compare our best-performing model. The results are described below:

The ground-truth highlight of a random CNN article: *"Kim Ki-Jong is charged with attempted murder and assaulting a foreign envoy. He's accused of stabbing U.S. Ambassador Mark Lippert in the face and arm. Police said Kim opposed the joint U.S.-South Korean military drills ."*

MatchSum output: *"kim ki-jong is charged with attempted murder and assaulting a foreign envoy. he 's accused of stabbing u.s. ambassador mark lippert in the face and arm. police said kim opposed the joint u.s.-south korean military drills."*

ChatGPT output: *"The man accused of stabbing U.S. Ambassador Mark Lippert in Seoul last month has been charged with attempted murder, according to a South Korean court official. Kim Ki-Jong has also been charged with assaulting a foreign envoy and business obstruction. Kim's trial must begin within 14 days of receiving the indictment. The ambassador suffered a gash to his face and jaw that required 80 stitches. There was no serious facial nerve damage. Kim is believed to have opposed the joint South Korean-U.S. military drills."*

When computing the Rouge-1 score between the reference highlight and ChatGPT's output, it yielded 23.53%. In this case, the Rouge-1 score is relatively low, which indicates that there is only a small overlap in the unigrams between the two passages. This is likely due to the fact that the two passages are quite different in terms of their overall structure and content.

6 Conclusion

Text summarization is an intriguing research topic in the natural language-processing community that helps produce concise information. This paper aims to introduce and deep dive into various natural text summarization approaches used in the mainstream by researchers in the past and the current ongoing trends. There has been significant work done in the domain by statisticians and machine learning researchers to implement and test multiple novel algorithms in this domain. Methods such as frequency-driven and Latent semantic analysis that uses statistics-based mathematical models have proven to be successful to some extent. As described in this paper, many unsupervised machine learning approaches have been tried to achieve even better accuracy and meaningful text summaries, such as KMeans clustering. In this experiment, we studied different classes of extractive text summarization techniques that have been widely used across many different applications around the world. We dived into the study of 4 widely used summarization methodologies in this paper. In particular, we compared the traditional summarization methods with state-of-the-art transformer-based algorithms. We analyzed the performance improvement and model efficiencies based on three different evaluation metrics.

This paper answers different qualitative and quantitative approaches to text summaries. Further work can be done to try an ensemble of various methods to achieve even better results. We have used all the available articles in the CNN dataset. The accuracy pertaining to the specific use case can be uplifted by changing the design and algorithms accordingly suited to the problem at hand. A recently released ChatGPT can also be tested against current popular models to study and test the ways of improvements in this domain.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

Ishitva Awasthi, Kuntal Gupta, Prabjot Singh Bhogal, Sahejpreet Singh Anand, and Piyush Kumar Soni. 2021. Natural language processing (nlp) based text summarization-a survey. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 1310–1317. IEEE.

Vincent Chen, Eduardo Torres Montaño, and Liezl Puzon. 2017. An examination of the cnn/dailymail neural summarization task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2367.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

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*.

A. Vashwani et al. 2017. Attention is all you need. *NeurIPS*.

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kaiz Merchant and Yash Pande. 2018. Nlp based latent semantic analysis for legal text summarization. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1803–1807. IEEE.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.

OpenAI. 2022. OpenAI chatgpt.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 3–21. Springer.

Kamal Sarkar. 2013. A hybrid approach to extract keyphrases from medical documents. *arXiv preprint arXiv:1303.1441*.

Abhishek Kumar Singh, Manish Gupta, and Vasudeva Varma. 2017. Hybrid memnet for extractive summarization. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2303–2306.

G Sreenivasulu, N Thulasi Chitra, B Sujatha, and K Venu Madhav. 2022. Text summarization using natural language processing. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, pages 653–663. 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. *Advances in neural information processing systems*, 30.

Chia-Wei Wu and Chao-Lin Liu. 2003. Ontology-based text summarization for business news articles. In *CATA*, pages 389–392.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. *arXiv preprint arXiv:1902.00863*.

Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020. Unsupervised extractive summarization by pre-training hierarchical transformers. *arXiv preprint arXiv:2010.08242*.

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

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*.