QSum: Automated Summarization of Quora Diverse Posts using Llama-2-7b

Saima Sanjida Shila[*]
sshila1@lsu.edu
Louisiana State University
Baton Rouge, Louisiana, USA

Nushrat Jahan Ria[†]
nria1@lsu.edu
Louisiana State University
Baton Rouge, Louisiana, USA

Abstract

Does anyone want to spend hours to find out the desirable answer by sifting through large Q&A online posts? The most probable answer is no, most people don't like to spend hours looking for the answers to their questions. Quora is a great platform for asking diverse questions online. Acknowledging the mentioned issue we want to automate the summarization of Quora posts, this study aims to improve user accessibility by making it easier for users to find theoretical answers inside the platform. Our method has been established through background and dataset studies. We have experimented with different transformer models like: Llama-2-7b, BERT, BART and baseline models like: TextRank, SentenceRank, T5, K-Clustering, TFIDF to summarize Quora articles. After evaluating the models we have proposed a hybrid model using the highest scored models from our experiment. However, the hybrid model doesn't perform well and we decided to go with the Llama-2-7b to create the API for easy access to the users. This work furthers the development of automatic summarising methods and can improve the way information is retrieved from online communities such as Quora.

CCS Concepts: • **Long Quora Post → Automated Quora Post Summary**.

Keywords: text summarization, BERT, LLM, NLP, Quora

1 Introduction

In the current era of rapid digital advancement, the pursuit of information remains constant, yet the process of sifting through vast online repositories can be daunting and time-consuming. Recognizing the limitations of conventional methods for retrieving information, this study aims to enhance user experience by automating the summarization of Quora posts. With its diverse range of questions and answers, Quora stands as a valuable platform for the sharing and dissemination of knowledge. However, the sheer volume of content can overwhelm users, hindering their ability to quickly find relevant information.

To address this challenge, our research aims to leverage cutting-edge natural language processing techniques such as BERT, Llama 2, BART, K-Clustering, T5, TextRank, TFIDF, Sentence Rank, and Hybrid models to generate concise and informative summaries of Quora articles.The field of natural language processing (NLP) has achieved great strides in text summarization using deep learning (DL). Deep learning (DL) techniques need substantial training using sizable parallel corpora of text documents and summaries. The CNN/DailyMails collection, for example, has 286K pairings of news stories and summaries that were authored by humans. For Quora postings, these kinds of sizable datasets are scarce. A pretrained language model is called Llama-2-7b. It has been trained on a large corpus of text data before being fine-tuned for specific NLP tasks such as text summarization, question answering, or text generation. This pretraining process allows Llama-2-7b to learn rich representations of language patterns and structures, enabling it to perform well on a variety of downstream tasks with minimal additional training. By simplifying complex information into easy-to-understand snippets, our approach aims to improve user accessibility and streamline the process of retrieving information. Through meticulous background and dataset studies, we have established a robust methodology for automated summarization. Our goal is to provide users with thorough summaries that accurately capture the essence of the original text, facilitating quicker comprehension and reducing the time spent searching for solutions. Anticipated outcomes include simplified information availability, reducing the time spent searching, and improving the overall experience for Quora users. Ultimately, this research contributes to the advancement of automatic summarization methods, offering valuable insights into improving information retrieval from online communities like Quora.

2 BACKGROUND STUDIES

To establish the foundation for the proposed solution of automatic text summarization, it is crucial to delve into prior research endeavors, assessing the strengths and limitations

[*] Both authors contributed equally to this research.
[†] Both authors contributed equally to this research.

, ,

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in , https://doi.org/10.1145/nnnnnnn.nnnnnnn.

of various methodologies. Text summarization has been a subject of research and exploration for many years. Numerous models have been developed and evaluated on diverse datasets to produce succinct summaries[25]. Automatic Text Summarization (ATS) is becoming much more important due to the exponential growth of textual content on the Internet and in various repositories such as news archives, scientific papers, and legal documents. Manual text summarization is labor-intensive, time-consuming, costly, and often unfeasible given the vast volume of textual data. Researchers have been trying to improve ATS techniques since the 1950s. ATS approaches are either extractive, abstractive, or hybrid. Extractive summarization selects key sentences directly from the original text to create a summary, preserving the wording and structure. Unlike abstractive summarization, it does not generate new sentences. The extractive and abstractive techniques are combined in the hybrid approach. The produced summaries are still very different from the summaries created by humans, even with all the suggested techniques [10].

The research by Tehseen and Omer [1] used different techniques and methods proposed for text summarization based on different types of applications. Some are NLP, supervised ML techniques, NNs, and KNN Different algorithms are used like the word vector embedding, k-nearest neighbor algorithm, human learning algorithm, etc. The research by Zaware and Patadiya [29] implemented a combination of TFIDF and Textrank algorithm with some NLP methods which efficiently summarize the given data and perform better than the other systems. The TextRank algorithm, introduced by Mihalcea et al.[24] in 2004, stands as a classic approach in extractive summarization. This method integrates a graph model into automatic summarization, computing the importance score for each sentence and extracting the top n sentences to form the article summary. This model Lacks the incorporated semantic understanding of the text, leading to potential inaccuracies in selecting key sentences based solely on statistical patterns and graph analysis. Extractive text summarization using k-clustering algorithm showed better results than other text summarizers. Graph-based algorithms for clustering could improve the result [10].

Many studies have been conducted in the past to survey Automatic text summarization(ATS) methods; however, they generally lack practicality for real-world implementations, as they often categorize previous methods from a theoretical standpoint. Moreover, the advent of Large Language Models (LLMs) has altered conventional ATS methods [15]. The study by Abdel-Salem and Rafea [2] evaluates BERT-based models for text summarization, introducing "SqueezeBERT-Sum" with improved efficiency while maintaining performance. This study only focuses on BERT-based models for text summarization techniques or architectures. The research by Akiyama, Tamura, and Ninomiya [3] focuses on the abstractive summarization model (Hie-Bart) which captures hierarchical structures of a document (i.e., sentence-word structures) in the BART model. Although the existing BART model has achieved state-of-the-art performance on document summarization tasks, the model does not have interactions between sentence-level information and word-level information.

3 TASK DEFINITION

We provide the following definition of Quora Post Summarization: Considering a Quora response post with N sentences, the objective is to choose a few sentences to create a brief and independent synopsis. This may be thought of as essentially a contextualized sentence classification task where a sentence is classified as either being in the summary or not. In NLP, this task is often referred to as Extractive Summarization (ES). It contrasts with Abstractive Summarization (AS), a different kind of summarization. AS creates new content to summarize the document rather than using summary phrases from the original version [14]. One distinct problem in AS as opposed to ES is the possibility of distorted or fake text being added during text production [16], [18]. Up to 30% of abstractive summaries include factual inconsistencies, according to many research [7, 11, 12, 17]. The categorization job is for sentences in which a sentence can be included in the summary or not. Furthermore, due to domain shift, these mistakes may become more common when employing abstractive summarization models on Quora posts because most of them are pre-trained on news articles or general-domain corpora. The more difficult abstractive summary problem will be left for future study; in this work, we concentrate on extractive summarization as the initial step toward Quora post-summarization. We first provide a comparison between different transformer summarization

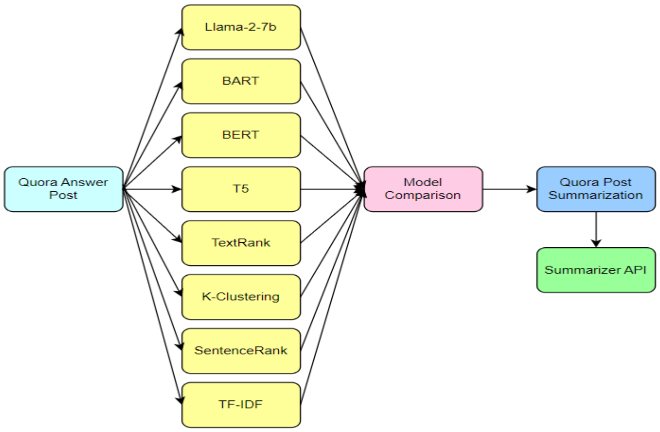


Figure 1. Flowchart of our approach.

models and baseline summarization models specifically for Quora postings in Section 4. Then, in Section 6, we provide an API for the best summarization techniques.

4 METHODOLOGY

The methodology for this study has been covered in this section. For the Quora post summary, we provided a detailed overview of the transformer models, baseline models, model processing, and problem comprehension. The requirements are used to determine the overall model parameters.

4.1 MODEL DEFINITION

In this task, we have experimented with 8 different models to find out the performances for the summarization. We used four abstractive text summarization and four extractive summarization models where three of which were transformer models and the rest of them were baseline models. We determine the performance of these models and choose the best models to create a hybrid model for our summary. The model descriptions are given below:

Llama 2: Llama 2 is a set of large language models (LLMs) with between 7 billion and 70 billion parameters that have been pre-trained and refined. In this research, we have used 7 billion parameters. These improved LLMs, dubbed Llama 2-Chat, are designed with conversation use cases in mind. Their models perform better than open-source chat models on the majority of the benchmarks they studied, and they might be a good alternative to closed-source models based on user ratings for safety and helpfulness. They offer a thorough explanation of the strategy for optimizing Llama 2's safety features and enabling the community to expand on their efforts and support the ethical advancement of LLMs [28]. The development of open-source instruction-tuned models that perform on numerous benchmarks similarly to GPT-3 [6] despite their lower sizes has been made possible by the Llama family of LLMs.

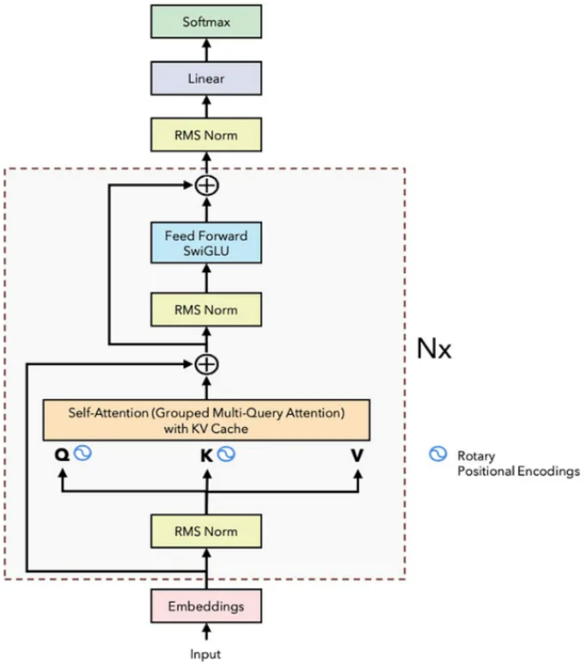


Figure 2. Llama 2 architechture.

BART: After tokenizing input text, the BART (Bidirectional and Auto-Regressive Transformers) model runs it via an encoder-decoder architecture [19]. Text is transformed into numeric tokens for processing during the tokenization process. Through bidirectional input analysis, the encoder extracts contextual data from both past and future tokens. To preserve coherence, BART uses masked autoregressive generation during the decoding stage, which predicts each token based on ones that have already been created while blocking access to future tokens. To modify model parameters during training, it computes the likelihood of each token given the input and previously produced tokens [8]. It does this by minimizing a loss function. Optimizing for certain tasks, such as text summarization, enhances BART's functionality even more. Its efficacy stems from its capacity to extract both local and global dependencies from the input text, which enables it to provide high-caliber summaries that are coherent and pertinent.

BERT: By pre-training copious quantities of text data using masked language modeling and next-sentence prediction tasks, the BERT (Bidirectional Encoder Representations from Transformers) model can function [27]. In pre-training, BERT considers both the left and right context when learning contextual representations of words. Before feeding the subwords into a multi-layer bidirectional transformer encoder, the input text is tokenized. To capture the complex connections between words, each layer pays attention to every token in the input sequence. BERT can produce contextualized representations for each token and concentrate on pertinent data thanks to its attention mechanism [26]. In fine-tuning, downstream tasks like text categorization or question answering are tailored to the pre-trained weights of BERT. BERT is capable of achieving state-of-the-art performance on a variety of natural language processing tasks, such as text classification, named entity identification, and sentiment analysis, by utilizing its pre-trained knowledge and fine-tuning task-specific data. Because of its attention mechanism and bidirectional nature, BERT can collect rich contextual information, which makes it an effective tool for producing and comprehending natural language content.

K-Clustering: In K-means clustering, a dataset or paragraph is iteratively divided into k groups according to how similar the data points are. It first allocates centroids to each cluster at random. Next, each data point is assigned to the cluster of the closest centroid based on the distance between each point and the centroids. Based on the average of the data points in every cluster, it then modifies the centroids.

Until centroids settle or a certain number of iterations is achieved, this procedure continues to loop [20]. By minimizing intra-cluster variance, K-means successfully forms cohesive clusters out of related data points.

Sentence Rank: Sentence ranking is the process of evaluating a text's sentences according to several factors, including coherence, relevance, and informativeness. Graph-based algorithms such as TextRank or PageRank, which represent sentences as nodes in a graph and capture the links between words as edges, are common techniques [22]. Sentences are rated based on their centrality or relevance in the text by iteratively updating ratings based on these linkages. The summary is then formed by choosing the sentences that rate the highest.

T5: Text-to-text tasks are transformed into a single format by T5, or Text-To-Text Transfer Transformer, such that both the inputs and outputs are text sequences. It uses self-attention processes to record contextual connections within the input sequence, according to the transformer design [4]. T5 learns to minimize a loss function to produce target sequences from matching input sequences during training. This makes it capable of carrying out a variety of natural language processing activities in an integrated way, such as question answering, translation, text summarization, and more.

TFIDF: A method for representing a term's relevance inside a document concerning a corpus of documents is called TF-IDF, or Term Frequency-Inverse Document Frequency [9]. The way it operates is by first tokenizing the text into words and then figuring out how frequently each term appears in the document. It then calculates the inverse document frequency, which expresses the amount of information a phrase offers throughout the whole corpus. Lastly, to give each phrase a weight and emphasize its significance in the document, TF-IDF combines these two measures.

TextRank: An algorithm used for extractive summarization is called TextRank. It functions by first segmenting the text into sentences, after which it builds a graph with each phrase serving as a node. The closeness of the words determines how much weight is assigned to the edges connecting nodes. The significance of each sentence is then assessed by TextRank using an iterative process akin to PageRank, taking into account the relationships between the sentences [23]. Finally, depending on their significance rankings, it chooses the top-ranked phrases to serve as the summary.

4.2 EXPERIMENTAL SETUP:

Out of the three transformer models that were evaluated, Llama 2 showed better performance than the others. In terms of summary quality, it performed noticeably better than other abstractive text summarizing models in our experiment. Additionally, K-Clustering demonstrated greater performance when compared to other baseline models and extractive text summarizing approaches. Thus, we decided to experiment with a hybrid model using Llama 2 and K-Clustering. The mechanism of our hybrid model is given below: To get the text ready for more processing, we preprocess it first. To properly utilize K-means clustering, we tokenize the article into sentences. Reason for this decision: Sentences are better suited for clustering since they usually have more atomic units of meaning than paragraphs. Furthermore, the information in words is frequently more uniform, which might aid in lowering noise during the clustering process. Nevertheless, several variables, like the kind of text, the required degree of granularity in the summarization, and the particular clustering technique used, might affect how effective sentence-level clustering is. After tokenizing the text, we vectorize the sentences and specify the maximum number of clusters. Next, we group each sentence into a cluster based on the number of groups that K-means clustering optimally yields. By choosing one exemplary sentence from every cluster, we create a synopsis that encompasses the many themes or subjects found in the text.

Then, we make use of Llama 2's abstractive text summarizing features to provide an extra summary. Lastly, we integrate the best features of both methods into a hybrid summary by fusing the Llama 2 and K-means summaries. The objective of this method is to provide a thorough and coherent summary of the input text by combining sophisticated summarization with sentence-level clustering. When these methods are combined, they provide a fresh approach to the summarizing job and may result in better summary coverage and quality.

5 EVALUATION

To analyze these models we have found out the ROUGE score. The number of unigrams in R that also occur in C divided by the total number of unigrams in R yields the ROUGE-1 recall ratio. Then, using the conventional F1-score calculation, the ROUGE-1 F1-score may be simply calculated from the ROUGE-1 accuracy and recall. ROUGE performs similarly on both the Abstractive and Extractive algorithms, therefore it doesn't provide very good results; most of the time, numerous executions outperform a single one [5]. That is why we didn't depend on the only ROUGE score. We determine the BLEU score and F1 score for the model evaluations. The evaluation results are shown in Table 1 and Figure 1.

ROUGE-R: Recall-Oriented Understudy for Gisting Evaluation, or ROUGE-R, is a measure that compares the recall of the generated summary to the reference summaries to assess the quality of text summarization [21]. It determines the percentage of n-grams (usually bigrams, trigrams, or unigrams) that overlap between the reference summaries and the created summary. An improved ability to recall significant details from the summary is indicated by a higher ROUGE-R score.

BLEU: An automated text summarization system's quality may be assessed using a metric called BLEU (Bilingual

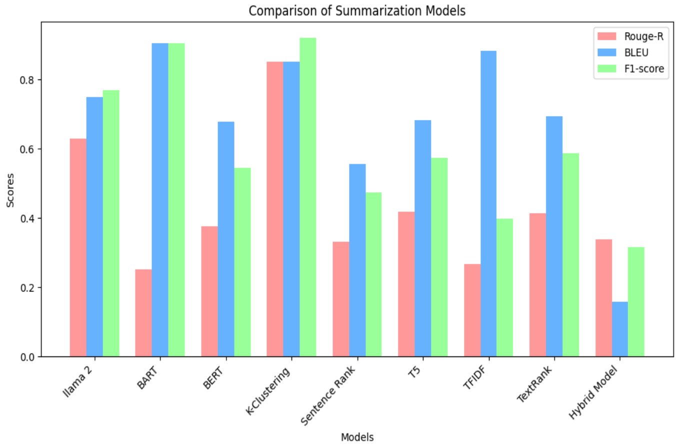


Figure 3. Comparison of Text Summarization Models.

Table 1. Different Text Summarization Models Evaluation

Model Name	Rouge-R	BLEU	F1-Score
Llama 2	0.630	0.750	0.770
BART	0.252	0.905	0.905
BERT	0.375	0.677	0.545
K-Clustering	0.852	0.852	0.92
Sentence Rank	0.331	0.556	0.474
T5	0.417	0.683	0.573
TFIDF	0.266	0.883	0.398
TextRank	0.41	0.694	0.586
Hybrid Model	0.337	0.159	0.316

Evaluation Understudy). Through n-gram overlap comparison, it evaluates the degree to which a produced summary aligns with one or more reference summaries. By comparing the produced summary to the reference summaries and looking for n-grams, BLEU determines accuracy like machine translation assessment [13]. A maximum score of 1 denotes a perfect match, while a higher BLEU value implies a better agreement between the produced summary and the reference summaries.

F1-Score: The F1 score assesses how well the generated summary balances precision and recall when compared to a reference summary or ground truth in the context of summarization. It takes into account the degree to which pertinent information is recorded (recall) and incorporated in the resulting summary (precision) in the original text. An improved F1 score means that the produced summary minimizes unnecessary details while efficiently capturing the important information from the original text. Here K-Clustering outperforms all other models and Llama 2 performed better than the other 2 transformer models. The scores are shown in a bar chart for data visualization.

Here K-Clustering outperforms all other models and Llama 2 performed better than the other 2 transformer models. The scores are shown in a bar chart for data visualization.

6 RESULTS

From the model evaluation Llama-2 has the highest ROUGE-R, BLEU and F1-score among the transformer models and K-Clustering has the highest score among all other models. On the other hand, they hybrid model of Llama-2 and K-Clustering doesn't perform well. We decided to create an API with Llama-2 model as it perform in specific domain summarization. Flask and Ngrok were used to develop the API, and the Llama-2-7b model requires many phases. First, we installed Flask using pip and created Python scripts to define API endpoints using Flask routes in order to build up a Flask application. The Llama 2 model had been included into our Flask application. Usually, this entails loading the Llama 2 model using the transformers library and applying it for text summarizing tasks. Following model integration, we developed API endpoints for several functionality, including text summarization. These endpoints will respond to requests, process the information according to the Llama 2 model, and provide the client with the output. When the Flask application is prepared, the Flask server is launched locally to verify its functionalities. We have taken feedbacks from 5 programmers who use Quora for their theoretical question understanding. They gave quite a good feedback about the summarize but suggested us to create a chrome extension for easy access to them.

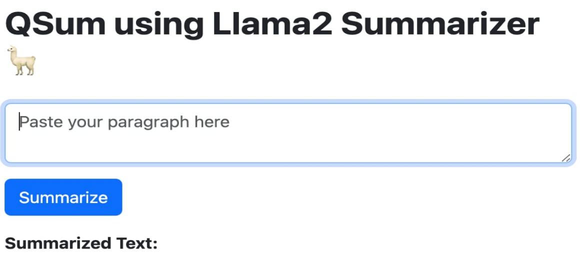
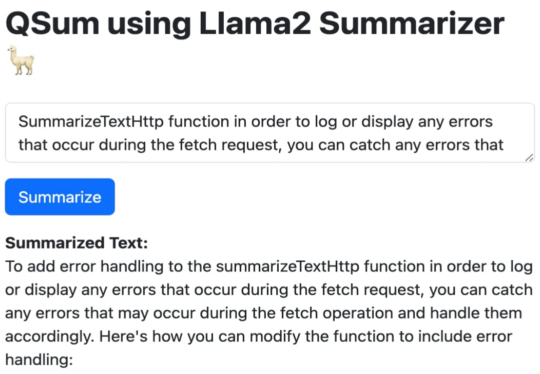


Figure 4. Initial API.

7 FUTURE WORK

To improve the usability and accessibility of summarizing tools, there are a number of fascinating directions that we will be explored in future research. The development of a Chrome plugin designed to improve user accessibility to tools



QSum using Llama2 Summarizer 🦙

SummarizeTextHttp function in order to log or display any errors that occur during the fetch request, you can catch any errors that

Summarized Text:
To add error handling to the summarizeTextHttp function in order to log or display any errors that occur during the fetch request, you can catch any errors that may occur during the fetch operation and handle them accordingly. Here's how you can modify the function to include error handling:

Figure 5. API with summarized Text.

for summarizing content when exploring the web. With the help of this plugin, users will be able to effortlessly summarize internet material and swiftly extract the most important information from web pages. Plans also include creating an abstractive summarizer API, which will use cutting-edge natural language processing methods to produce summaries that are more human-like, in order to enhance the capabilities of the summarizing API. In addition, efforts will be focused on developing a multi-document summarizer API that can be used to summarize groups of documents, meeting the needs of situations when users must extract data from several sources. The goal of these programs is to advance the field of summarization technology by giving users more flexible tools to extract and understand information from a wide range of textual sources on a variety of platforms and settings.

8 CONCLUSION

In this study, we introduced novel approach for automated text summarization of Quora answer posts, employing both transformer and baseline models. Our findings reveal that for Extractive summarization, K-Clustering outperforms all other models and Llama-2-7b performed better than other transformer models such as BART and BERT. Despite efforts to improve outcomes with a hybrid model combining k-clustering and Llama 2, the resulting model showed underwhelming performance in Rouge, Bleu, and F1-score metrics. Also, we have implemented QSum including API which generates accurate and concise Quora answer posts summary using Llama-2-7b. The availability of an API for QSum offers users convenient access to succinct summarizations of Quora's answer posts.

In conclusion, while k-clustering represents a notable advancement in automated text summarization, Llama 2 also exhibited commendable performance. Abstractive summarization and ongoing research efforts hold the potential to further enhance the capabilities of automated summarization systems, catering to diverse needs and use cases.

References

[1] [n. d.]. Text Summarization Techniques Using Natural Language Processing: A Systematic Literature Review. 9 ([n. d.]).

[2] Shehab Abdel-Salam and Ahmed Rafea. 2022. Performance Study on Extractive Text Summarization Using BERT Models. *Information* 13, 2 (2022). https://doi.org/10.3390/info13020067

[3] Kazuki Akiyama, Akihiro Tamura, and Takashi Ninomiya. 2021. Hie-BART: Document Summarization with Hierarchical BART. Association for Computational Linguistics, Online, 159–165. https://doi.org/10.18653/v1/2021.naacl-srw.20

[4] Betul Ay, Fatih Ertam, Guven Fidan, and Galip Aydin. 2023. Turkish abstractive text document summarization using text to text transfer transformer. *Alexandria Engineering Journal* 68 (2023), 1–13.

[5] Marcello Barbella, Michele Risi, and Genoveffa Tortora. 2021. A Comparison of Methods for the Evaluation of Text Summarization Techniques.. In *DATA*. 200–207.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[8] Hugh A Chipman, Edward I George, and Robert E McCulloch. 2010. BART: Bayesian additive regression trees. (2010).

[9] Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. 2016. Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications* 7, 4 (2016), 285–294.

[10] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* (2021). https://doi.org/10.1016/j.eswa.2020.113679

[11] Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*. 2214–2220.

[12] Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 166–175.

[13] Yvette Graham. 2015. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 128–137.

[14] Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications* 121 (2019), 49–65.

[15] Dan Meng Jun Wang Jinghua Tan Hanlei Jin, Yang Zhang. 2024. A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods. (2024).

[16] Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839* (2021).

[17] Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960* (2019).

[18] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text

summarization. *arXiv preprint arXiv:1910.12840* (2019).

[19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[20] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern recognition* 36, 2 (2003), 451–461.

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

[22] JN Madhuri and R Ganesh Kumar. 2019. Extractive text summarization using sentence ranking. In *2019 international conference on data science and communication (IconDSC)*. IEEE, 1–3.

[23] Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar. 2019. Graph-based text summarization using modified TextRank. In *Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018*. Springer, 137–146.

[24] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Dekang Lin and Dekai Wu (Eds.). Association for Computational Linguistics, Barcelona, Spain, 404–411. https://aclanthology.org/W04-3252

[25] Rahul, Surabhi Adhikari, and Monika. 2020. NLP based Machine Learning Approaches for Text Summarization. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00099

[26] S Shreyashree, Pramod Sunagar, S Rajarajeswari, and Anita Kanavalli. 2022. A literature review on bidirectional encoder representations from transformers. *Inventive Computation and Information Technologies: Proceedings of ICICIT 2021* (2022), 305–320.

[27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[28] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[29] Sarika Zaware, Deep Patadiya, Abhishek Gaikwad, Sanket Gulhane, and Akash Thakare. 2021. Text Summarization using TF-IDF and Textrank algorithm. https://doi.org/10.1109/ICOEI51242.2021.9453071