University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Faculty of Computing, University of Ibadan, Ibadan, Nigeria

Volume 15 No. 1, September 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

A Review of Automated Text Summarization Models on Diverse Datasets: An Evaluation Perspective

[1]Gold Ezinwa Egbuonu, [2]Precious Kelechukwu Chika-Ugada, [3]Chinwe Ndigwe, [1]Chigozie Dimoji, [2]John Prince Uzodinma, [4]Ezekiel Gabriel Nwibo, [1]✉Jacinta Chioma Odirichukwu, [1]Obilor Athanasius Njoku, [1]Chukwuma D Anyiam

[1]Department of Computer Science, School of Information and Communication, Technology, Federal University of Technology, Owerri, Imo State Nigeria

[2]Research Assistant % Dr. Jacinta Chioma Odirichukwu, [1]Department of Computer Science, School of Information and Communication, Technology, Federal University of Technology, Owerri, Imo State Nigeria

[3]Department of Computer Science. Chukwuemeka Odumegwu Ojukwu University(COOU), Uli, Anambra State, Nigeria

[4]Department of Computing, (School of Arts and Creative Technologies), University of Greater Manchester, Bolton, UK.

**Corresponding Author: jacinta.odirichukwu@futo.edu.ng*

Abstract

This paper reviews Automatic Text Summarization which is one of the tasks in Natural Language Processing (NLP). It is driven by speedy increase in textual data across domains. The reviews systematically examined the recent advancements in Extractive, Abstractive and hybrid automatic text Summarization Models between 2019 and 2025 using Preferred Reporting Items for Reviews and Meta-Analysis (PRISMA). Selected and relevant related papers were taken from Elsevier, Google scholar, IEEE Xplorer, ACM digital library, and Springer. After removing duplicates (n=96), 174 irrelevant records were removed to meet the inclusion criteria covering models like BERT (Bidirectional Encoder Representations from Transformers), BART (Bidirectional and Auto-Regressive Transformers), T5 (Text-To-Text Transformer), TextRank, LSA (Latent Semantic Analaysis), and PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-to-Sequence Models) across Diverse datasets including news, scholarly and technical corpora. Extractive approaches depicted strong lexical accuracy and computational efficiency, whereas transformer-based Abstractive models showed superior semantic coherence but needed higher computational costs. This review paper also highlighted persistent gaps including dataset bias, long-document Summarization, hallucination in generative models, and over reliance on traditional metrics such as ROUGE.The results show the need for cross-domain evaluation, hybrid model integration, and adoption of advanced semantic metrics like BERTScore and MoverScore. Future directions should take into priority cross-domain benchmarks, standardized multi-metric evaluation, hybrid approach exploration and testing for long and multilingual documents. In furtherance, Reproducible Reporting of Computational cost such as GPU-hours and failure modes such as hallucinations will support more practical comparisons.

Keywords: Automatic Text Summarization (ATS), Natural Language Processing (NLP), Extractive and Abstractive Summarization, Transformer-Based Models (BERT, BART, T5, PEGASUS), Deep Learning for Summarization, Large Language Models (LLMs).

Gold Ezinwa Egbuonu, Precious Kelechukwu Chika-Ugada, Chinwe Ndigwe, Chigozie Dimoji, John Prince Uzodinma, Ezekiel Gabriel Nwibo, Jacinta Chioma Odirichukwu, Obilor Athanasius Njoku and Chukwuma D Anyiam (2025). A Review of Automated Text Summarization Models on Diverse Datasets: An Evaluation Perspective. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 15 No. 1, pp. 141 –. 157

1.0 Introduction

With the growth of the internet and multimedia technology, the amount of text data has increased exponentially [1]. This available data, which is increasing, exists across various domains such as the news media, scientific publications, legal documents, conversational platforms, etc. This therefore brings about the need for an effective Automated

Summarization Systems (ATS). Automatic-Text-Summarization (ATS), using Natural-Language-Processing (NLP) processes, aims to make succinct and accurate summaries, thus, it is significantly reducing the human effort required in processing large volumes of text [2].

Text summarization is a research technique used in dealing with plenty data and knowledge discovery in a time-constrained environment [1]. It aims to create short summaries from large documents and makes sure that the originality and key information is maintained. Automatic-Text-Summarization (ATS) is important because of its ability to automatically identify, extract, and synthesize relevant information from large texts and documents, making it easier for users to understand the key concepts without needing to read the entire document thoroughly. This ability makes Automatic-Text-Summarization (ATS) useful for decision making processes, research activities and of course, information distribution across multiple sectors or domains.

Text summarization consists of two main approaches which are Extractive and Abstractive. The extractive approach selects the most important sentences in the input document(s) then concatenates them to form the summary [3]. The extractive approach maintains the originality of the document, that means that it doesn't change anything or words and the summary created is produced from the context of the document. The abstractive approach represents the input document(s) in an intermediate representation then generates the summary with sentences that are different than the original sentences [3].

The abstractive approach is also referred to as the paraphraser approach where it changes the words or context of a document and produces a summary that shows the general idea of what the document is about or depicts but doesn't have to be the exact replica of the document. That is it doesn't copy word for word to produce the summary but instead summarizes the document or text by extracting the key points and paraphrasing it. Both approaches have their own advantages and limitations. The extractive approach is reliable for maintaining the original context of a content but this also means that it has limited flexibility.

This paper proposes the development of an Automatic Text Summarization Model with the purpose of evaluating its performance across 'a diverse range of datasets'. These models have performed extremely well in domain-specific tasks but we still need to understand how they perform in domains that are different than their usual pre-trained data or environment. This project emphasizes on a comparative evaluation perspective, with the aim of understanding how well summarization models like PEGASUS, BART, T5, BERT, Text Rank and LSA generalize across different text domains. Through this evaluation, this project aims to identify the optimal model selection for specific domains, understand cross- domain generalization, and provide recommendations for domain-independent summarization models that can handle data from different contexts.

Large Language Models work quite well with general-purpose data and many tasks in Natural Language Processing. However, they show several limitations when used for a task such as domain-specific abstractive text summarization [4]. It essentially means that the model doesn't know the domain-specific corpus contains words and concepts since they were not part of model's pre-training [5]. Therefore, a comprehensive evaluation of these models trained and tested on diverse datasets to identify their weakness, strength, and generalization capabilities is needed.

The aim of this study is to perform a comprehensive evaluation on text summarization models such as PEGASUS, BART, T5, BERTSUM, Text Rank and LSA models trained and tested on diverse datasets to identify their weakness, strength, and generalization capabilities.

The following are the specific objectives of this project to achieve this broad goal:

1. To compile and preprocess diverse text summarization datasets across multiple domains.
2. To develop and fine tune extractive and abstractive summarization models using transformer based architectures.
3. To perform a very comprehensive evaluation of each model performance on each dataset using both automatic and human-centered evaluation metrics

and highlight the evaluation based results.

4. To provide analytical recommendations for building robust domain-independent summarization systems.

1. Literature Review

2.1 Conceptual Framework

2.1.1 Automated Text Summarization (ATS)

Automated Text Summarization is a computer-based approach for reducing large volume of texts into concise summaries while preserving important information and meanings. This concept includes both the theoretical understanding of summarizing texts and the practical implementation of algorithms that will carry out or perform tasks such as identifying, synthesizing and extracting the relevant information from the contents of the documents [3]. ATS systems operates on the principle that not all information in a document is equally important and that through various computer-based techniques, they can distinguish which content is important and which is not.

The Automated Summarization involves several key variables which includes input text characteristics, summary length requirements, domain specificity and quality metrics. The input text characteristics considers factors such as document length, language complexity, structural organization and the domain-specific terminologies that could influence the level of difficulty for the summarization [1]. For summary length requirements, we look at the compression ratio between the original document and summarized content and it is usually expressed as percentage of original length or absolute word count. The Domain specificity tells us the degree to which a content belongs to a specific domain or field such as legal, medical or technical domains, because it can affect the language complexity and the background knowledge required to perform summarization [2].

Quality metrics are dependent variables in summarization evaluation and they include both automated measures such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation), BLEU (Bilingual Evaluation Understudy), BERTScore, and human centered assessments such as coherence, informativeness and readability scores. These metrics provides the qualitative and quantitative measures of an effective summarization, allowing comparative analysis of different data and ATS approaches.

The general architecture of an ATS system as shown in Figure 2.1 consists of the following tasks:

1. Pre-Processing: producing a structured representation of the original text using many linguistic techniques like sentences segmentation, words tokenization, removal of stop-words, part-of-speech tagging, stemming, etc. Some of the most used preprocessing procedures includes Parts of Speech (POS) Tagging, Stop Word Filtering, Stemming, Named Entity Recognition and Tokenization [6]. POS tagging technique includes grouping the texts or words into speech category such as nouns, verbs, pronouns, adjectives etc., and is known as speech tagging. Stop word filtering involves screening out stop words such as ‘A’, ‘An’, and ‘By’ before or after text analysis and eliminates them from the plain text or analyzed text. Stemming involves reducing the inflected form of a word into the ‘stem’ or root form known as ‘lemma’. For instance, the words ‘programmer’, ‘programming’ and ‘programs’ can all be reduced to the common word stem ‘program’. In named entity recognition, some words are recognized as the names of items, that could be a person’s name, company’s name, location name etc.

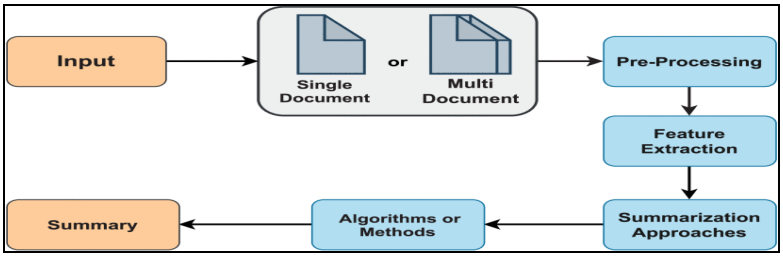


Figure 2.1: The general architecture of an ATS system [1]

Tokenization examines texts by dividing the texts into words, phrases, or symbols which are usually called tokens [6].

2. Feature Extraction: This technique discovers topic sentences, essential data traits or attributes from the source document(s). Automatic Text Summarization (ATS) systems uses two approaches to locate the important sentences in a text and these are: extracting the features and text representation approach.

a. Extracting the Features: This is the first step of the feature extraction process. It is important to represent the sentences as vectors or score them to locate the important sentences from the document. The most used features for calculating the score of a sentence showing the degree to which it belongs in a summary includes Term Frequency (TF), Term Frequency-Inverse Sentence Frequency (TF-ISF), Position Feature, Length Feature and Sentence-Sentence Similarity. Term frequency metric is used to represent a word's weight in a single document. TF-ISF generates the weights of words which shows their importance in a document. Position feature usually considers that the beginning and last sentences would provide more information about the document. In length feature, a sentence length can determine whether it is summary worthy. Sentences that are too long or too short are usually not included in the summary. Sentence-sentence similarity focuses on the resemblance of sentences to one another as it may be important for text summarization.

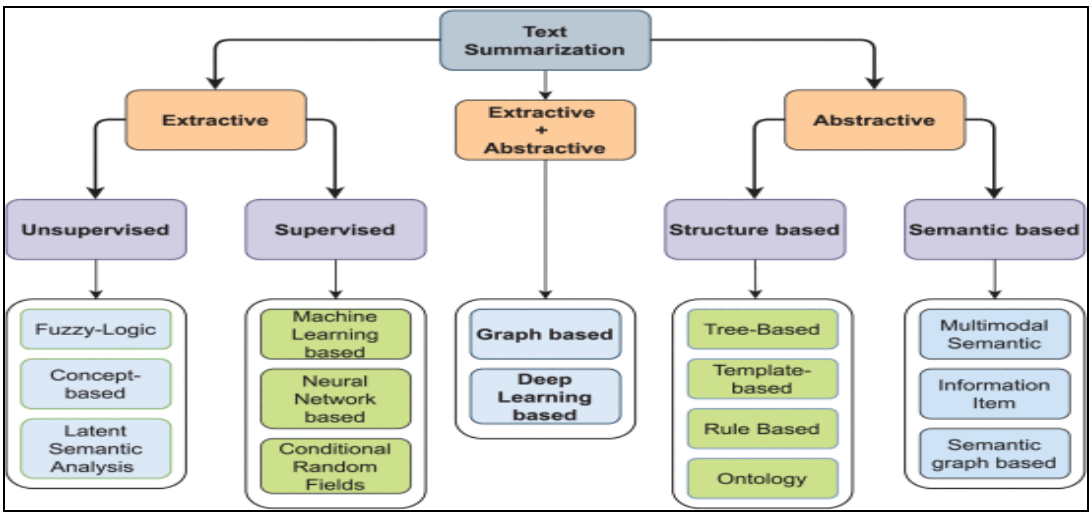
b. Text Representation: In NLP, this approach involves translating words into numbers so that computers can understand and decode the patterns in the language. Some popular text representation approaches includes the N-gram, Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and Word Embedding. The n-gram is a collection of words or characters with N components and the greater the 'N', the better the model. It is simple to create and the texts may be represented by vectors of reasonable size. Bag of words describes the occurrence of words within a document. In TF-IDF, 'IDF' measures how important a word is while 'TF' measures how frequently a word appears in a text. In word embedding, each word or phrase in a lexicon is mapped to an N-dimensional vector of absolute values. The three most widely used deep learning methods for word embedding includes the World2Vec, Global Vectors for Word Representation (GloVe) and FastText.

3. Summarization Approaches: when it comes to text summarization strategy, it is important that the first step is to determine which approach will be applied or used. Text summarization consists of two main approaches which are Extractive and Abstractive. The extractive approach selects the most important sentences in the input document(s) then concatenates them to form the summary [3]. The extractive approach maintains the originality of the document, that means that it doesn't change anything or words and the summary created is produced from the context of the document. The abstractive approach represents the input document(s) in an intermediate representation then generates the summary with sentences that are different than the original sentences [3].

4. Algorithms: Algorithms or methods are definite and finite ways of defining text summarization. There are different algorithms under various approaches that are applied to obtain a better version of summarized text [2]. This will be further discussed in details later in this chapter.

2.1.2 Automatic Text Summarization Approaches

ATS is generally viewed as a complex and time-consuming operation that most times lacks better results because computers do not have a proper understanding of human language. Text summarization approach varies based on the number of input documents, such as a single or multiple document(s), objective-wise generic, domain-specific, query-based and performance-wise. The two main approaches includes the extractive approach and the abstractive approach as shown in Figure 2.



a) Figure 2.2: Extended Classification of Automatic Text Summarization Approaches and their Learning Methods[1]

2.1.2.1 Extractive Text Summarization (ETS)

When it comes to extractive text summarization, we see that it uses sentence selection and ranking algorithms to identify the most informative sentences within the source document. Its processes usually involves sentence preprocessing, feature extraction, importance scoring, and sentence selection based on the specified criteria. It aims to identify words in sentences and use them to create a summary [7]. Extractive text summarization consists of three tasks, the splitting, assigning scores and selecting tasks.

i. Splitting Task: The source document or file is split into sentences and then an intermediate representation of the text (which highlights the tasks) is created. This representation may be a topic or an indicator.
ii. Assigning Score Task: The ETS system assigns scores to each sentence to show their importance after the text representation. Topic representation scores sentences based on their topic words, while the indicator representation scores words based on the features of the sentence.
iii. Selecting Task: The ETS system chooses the highest scoring sentences and forms the summary from them.

There are two machine learning approaches or methods applied when it comes to extractive text summarization systems as highlighted in Figure 2. These approaches are the Supervised and Unsupervised machine learning approaches.

1. Supervised Learning Methods: For this method, the first step is learning how to label the documents by training the system to identify if it's summarized or not. And this would require already classified datasets for training where summarized and non-summarized texts are available with labels.
2. Unsupervised Learning Methods: This method performs the summarization without any help and usually requires advanced algorithms such as the concept-based, graph-based, Latent semantics, and fuzzy logic to take user input and work automatically which is beneficial for extensive data.

2.1.2.2 Abstractive Text Summarization

Abstractive Text summarization systems are more sophisticated system because of how they generate the summaries by understanding and paraphrasing the source content. It requires deep semantic understanding of input texts and the ability to generate grammatically correct sentence structures not originally present in the source document [8]. Abstractive Text Summarization systems identifies the key sections and main parts of a document by paraphrasing it. This text summarization process follows the following steps:

i. Analyzing the contents of the source document for the key points or relevant

data using a vocabulary set different from the source document's own.

ii. Paraphrasing the relevant data that follows the proper summary semantics and utilizing NLP models.

Abstractive summarization models also have two machine learning approaches or methods, which are the Structure-based and Semantic-based approaches.

1. Structure-based Method: This algorithm applies abstract or cognitive algorithms to continually filter the most critical data from the source documents. Algorithms like tree-based, rule-based ontology, template-based ontology are common algorithm used.
2. Semantic-based Method: This method applies NLP on the entire document(s) in attempt to refine the sentences. This method can easily find nouns and verb phrases with the use of some algorithms. These algorithms include the multimodal semantic method (MSM), semantic graph-based method (SGM), information item-based method (IIM), sematic text representation method (STRM), etc.

2.1.2.3 Extractive vs Abstractive Summarization Approaches

The difference between the extractive and abstractive summarization approach is important to divide and show how summarization works. Table 1 illustrates the differences between these approaches in multiple areas.

2.1.3 Automatic Text Summarization Algorithms

1. Unsupervised Learning Methods:

Unsupervised learning models are better and more efficient than supervised learning model and more suited for lengthy text summaries. They are efficient because they do not require human overviews or feedbacks to determine the essential features of a document. Various unsupervised learning methods or techniques are discussed below:

i. Fuzzy Logic Based Method: This design typically involves the use of fuzzy rules and membership functions to select the most important sentences from the source documents. It consists of four components which are the fuzzifier, defuzzifier, inference engine and a knowledge base [9]. Despite how effective this method may seem, it usually requires redundancy removal techniques for better results [10].

ii. Concept-Based Method: This method extracts the concepts and uses similarity measure to reduce redundancy in summaries. The concepts extracted are calculated and the sentences are scored by importance. Although this method has the same limitations as fuzzy logic based methods, fuzzy logic based methods handles ambiguous situations better than concept-based methods [11].

Table 1: Comparison of Extractive and Abstractive Summarization Approaches

S/N	Characteristics	Extractive Approach	Abstractive Approach
1	Content Generation	Picks existing sentences	Creates new sentences
2	Linguistic Fidelity	High (as it preserves the original content of the document)	Variable (as it may paraphrase the contents of the document)
3	Computational Complexity	Lower	Higher
4	Risk of Hallucination	Minimal	Moderate to High
5	Flexibility	Limited	High
6	Training Data Requirements	Moderate	Extensive

iii. Latent Semantic Analysis (LSA) Methods: LSA is an algebraic statistical method for extracting hidden semantic sentence structures. This method is important because it doesn't require any special training to find similar words that appear in separate sentences. Its limitation includes not analyzing word order, syntactic relations, morphologies, and it solely relies on information contained inside the document rather than outside knowledge [12].

2. Supervised Learning Methods:

These are sentence-level classifications that learn to differentiate between summarized and non-summarized texts or sentences. Although it will require more labeled training samples for effective classifications. Various supervised learning methods are discussed below:

i. Machine Learning (ML) Based Method: This method classifies the sentences as summary or non-summary classes using training data. Each document's sentences are presented as vectors. Machine learning algorithms are implemented on sets of training data from documents that are trainable. Classification of the sentences are based on its weight or importance. This method is applied when there are multiple documents that require the extractive text summarization type [13].

ii. Neural Network Based Method: This method uses a three-layered feed forward network that learns the features of sentences during training. Removal of infrequent features and combining similar elements followed by sentence ranking are the steps to define meaningful sentences. NN performs better than ML algorithms and can be seen as advanced Machine Learning [14].

iii. Conditional Random Fields (CRF) Method: These are statistical modelling techniques based on Machine learning that provides standard predictions. It uses Non-negative matrix factorization (NMF) approaches to extract accurate sentence features. Then proper elements are used to determine or define the introductory sentence of the document. It offers a more suitable representation of sentences although it specializes in domain-specific which requires an external domain-specific framework for its training phase. This makes it time consuming and ML and NN a better choice for supervised learning methods [15].

3. Structure-Based Methods:

Structure-based methods interprets phrases from the source document in a specified structure without changing or losing their meaning. Various structure-based methods are discussed below:

i. Tree-based Method: This method recognizes sentences that have similar knowledge and facts, then it mixes them together to provide an abstractive summary. This tree like structure is known as tree linearization which comes from dependency trees. These dependency trees are a representation of the text from source documents. The tree-based method helps to process multiple documents and identify important sentences using a syntactic tree. It produces less redundant summaries although it can overlook significant text phrases. It also focuses on syntax and not semantics [16].

ii. Template-Based Method: For template-based method, the topic and content is extracted into phrases by finding the similarities with a template space. This is used when a document requires an adequate guideline or human-made template for summarization. Templates for creating summaries are always predefined, so there is no much variety in the summaries. Therefore the summaries are not so fluent unlike the tree-based approach [17].

iii. Rule-Based Method: This method finds facts and the essential concepts in source documents using questions. "What is the Topic?" or "What are is time-being of the topic or story" are examples of the interrogation-like questions

rule-based method uses to generate abstractive summaries. This method is applied when we need to represent the input document as classes or list of aspects just like a query-based method. The down side of this method is that it is a time consuming process since the rules have to be prepared or predefined [16].

iv. Ontology-based Method: This method is a knowledge-based approach that defines the entity type of a specified domain. A knowledge base is applied to this method to improve the outcome of the summarization process. It performs extensively when the document has a knowledge structure, thus, it focuses on domain-specific related documents and produces coherent summaries. It is time consuming and cannot be generalized to other domains [18].

4. Semantic-Based Method:

This method illustrates the language of a document in Natural Language Generation (NLG) system. It has a significant focus on noun and verb phrase identification and is effective at creating less redundant and grammatically correct sentences. The following are various semantic-based methods discussed below:

i. Multimodal Semantic Method: This method like its name suggests, is used to get both images and text concepts within a source document. It establishes relationships by representing text and images in a multimodal material with its foundation being knowledge representation based on objects [19]

ii. Semantic Graph-Based Method: This method summarizes a document by building a graph for the original content called Rich Semantic Graph (RSG) and reducing the created semantic graph. Making brief, cohesive and grammatically correct sentences with reduced networks [20].

iii. Information Item Method: This method summarizes a document based on its abstract instead of generating an abstract from summarizing the document contents. The minor component of a source document is the information item. This method retrieves information from the logical flow of information in the text file or document and produces less redundant and more concise summaries [21].

2.2 Theoretical Framework

The very first research work on automatic text summarization was done by Luhn in the 1950s. It set the tradition for sentence extraction. His approach was applied to magazine articles and technical papers. He set the idea that shaped much of the later research on ATS systems. This idea was that some words in a document are descriptive of its content and that the sentences that convey the most important information in a document are the ones that contain many of the descriptive words that were close to each other. He also suggested using frequency of occurrence to determine the descriptive words and that words that occur too many times are likely to be the main topic of the document. That is, first he measured the importance of individual words then applied it to sentences. The most important sentences were then picked to form a summary automatically. In 1958, Luhn's work on summarization of scientific articles was published in the IBM journal [22].

2.2.1 Information Theory

Information theory lets us understand which parts of a text in a document contains the most important information. This theory was created by Shannon in 1948 and it lets us know that rare or unexpected information is more valuable than common information [23]. We use Information theory in text summarization to pick the best sentences and calculate how much information each sentence holds by looking at the words contained in it. Those sentences with diverse and unusual words are considered to be more informative and are likely to be included in the summary [24]. This information theory is applied through 'entropy based sentence selection mechanism'. And the entropy of a sentence can be calculated using the formula:

$$H(X) = -\sum_{i=0}^{n} P(x_i)\, log_2 P(x_i) \quad (1)$$

Where p(xi) represents the probability distribution of the words within a sentence.

This theoretical foundation ensures that summarization systems can identify the important information in the source document(s) and use these information for its summarization.

2.2.2 Graph Theory and TextRank Algorithm

In Graph theory, texts are treated like a network where sentences are connected based on how similar they are to one another. The application of graph based approaches to text summarization comes from the page rank algorithm used in web search through TextRank algorithm [25]. This idea comes from how google ranks its webpages. The important pages are linked to by other important pages [26]. In text summarization we create connections between sentences that share similar topics or words. The computer calculates the importance score of each sentence by looking at all these connections and the sentences with a high score are used as the summary because they present the main idea that other sentences also discuss [27]. The TextRank algorithm calculates the importance of a sentence using the formula:

$$S(V_i) = (1-d) + d \times \sum_{j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|}. \quad (2)$$

where **d** is the damping factor (that is usually 0.85), **In(Vi)** is the sentences linked to sentence Vi and **Out(Vj)** is the sentences linked from sentence Vj.

This particular approach works well for extractive summarization as it looks at the sentence quality and how it fits into the whole document, ensuring that the summary sentences maintain coherence to the main idea or context of the document [28].

2.2.3 Semantic Analysis

Semantic analysis helps computers understand words and the meaning behind those words. This is needed because sometimes different words can have similar meanings and at the same time, the same word could have different meanings [29]. The computer uses mathematical methods to now group words that hold similar meanings together even though they are different words. This helps create better summaries since the system can identify sentences that discuss the same topic, even when different words are used [30]. The mathematical model used by semantic analysis involves splitting a matrix, **A** into three places:

$$A = U\Sigma V^T \quad (3)$$

Where U is for the term loadings, Σ contains the singular values that indicates the importance of each semantic dimension, and V^T stands for the document loadings.

2.2.4 Attention Mechanism Theory

When it comes to attention mechanism, computers are taught how to pay attention to details or rather important parts of a text when creating a summary. This is similar to how we pay attention to relevant details while ignoring the less important ones. Attention mechanism was originally developed for neural machine translation [31]. When creating summaries, the computer does not treat all words equally, instead it learns to pay attention to words that would be very helpful for creating the summary [32]. The attention score between input position, **i**, and output position, **j** is calculated using the softmax formula:

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) \quad (4)$$

Where **eij** represents the compatibility function between the hidden encoder state (hi) and the decoder state (sj).

2.2.5 Reinforcement Learning (RL) Theory

The reinforcement learning theory is founded on the Markov Decision Process Framework. It involves teaching computers how to improve by learning from their mistakes. When applied in text summarization, the system tries different approaches and is rewarded with quality metrics (such as better ROUGE scores) when it creates good summaries [33]. While traditional methods train the systems using perfect summary samples, reinforced learning trains the system to learn from its mistake, which in turn produces better results in real situations [34]. The policy gradient theorem gives us the mathematical model:

$$\nabla\theta J(\theta) = E\pi\theta[\nabla\theta log\, \pi\theta(a|s) Q\pi\theta(s,a)] \quad (5)$$

Where **θ** stands for the model parameters, **πθ** is the policy (which is the probability distribution over actions), and **Qπθ** is for the action-value function [35].

2.2.6 Text Structure Theory

Text structure theory explains that there is a logical way that documents are arranged and that some may be ranked more important than the others. This theory helps the systems learn the relationship between texts. Say for instance, some sentences provide main information while others perform supporting roles of evidence or examples[36]. This theory ensures that summaries capture the important parts while maintaining the logic flow.

2.3 Empirical Review

Researchers have been trying to improve ATS systems since the 1950s [3]. ATS systems are grouped into extractive approach and abstractive approach. Abstractive Summarization approach includes the BART, T5, PEGASUS models while Extractive summarization approach includes the TextRank, LSA, BERT models.

2.3.1 Recent Advances in Extractive Summarization Systems

1. TextRank Algorithm: The textRank algorithm creates connection between similar sentences and then calculates which sentence is more important using these connections. TextRank and LexRank are both powerful graph-based algorithms that excel in extractive text summarization, each offering precise methodologies for assessing the importance of sentences within a record. TextRank, inspired through the PageRank algorithm, constructs a graph with sentences as nodes and edges representing their similarities, typically measured using cosine similarity. By way of iteratively calculating importance ratings based on sentence connections, TextRank identifies key sentences that make a contribution to a coherent summary [37].The accuracy of TextRank algorithm is above 90%, the average accuracy of 100 text analysis is 94.23%, the average recall rate and F1 value are 96.67% and 95.85%, respectively [38].

2. Latent Semantic Analysis (LSA): LSA understands text meaning by looking at word relationships. LSA is a vigorous unsupervised algebraic-statistical process for developing an implicit portrayal of text semantics based on word co-occurrences [39]. This means that it learns what words appear together to understand document topics or concepts. LSA with a greater number of summary phrases, particularly 10, has better precision (0.12) and recall (0.26) than LSA with only three summary sentences, which has poorer precision (0.03) and recall (0.05). This integration demonstrates the ability of machine learning technologies to modify and optimize summarization procedures. Moving forward, these findings can inform the development of more effective summarization algorithms, allowing for improved extraction and synthesis of critical information from research articles and other textual sources [40].

3. BERT for Extractive Summarization:

Bano et al. [41] explored the application of BERT for extractive summarization of lengthy scholarly articles in which the proposed approach included the assimilation of BERT to handle long documents and integrating BiGRU on top of BERT for better capturing the documents' global context. Also, to further enhance the robustness and applicability of the model, there is a plan to investigate its performance with diverse datasets, including those beyond the scope of the current research [42].

2.3.2 Recent Advances in Abstractive Summarization Systems

1. BART (Bidirectional and Auto-Regressive Transformers): BART [43] was developed as a tool to change corrupted files back to their original form. He explained that BART was introduced as a pre-training approach that learns to map corrupted documents to the original. BART performs comparably to RoBERTa on discriminative tasks, and achieves new state-of-the-art results on several text generation tasks. This makes it efficient for creating new summaries rather than copying the existing ones. He also mentioned that there will be Future work which should explore new methods for corrupting documents for pretraining, perhaps tailoring them to specific end tasks.

2. T5 (Text to Text Transfer Transformer): In 2019, T5 [44] was developed to be used as a

language convertor that treats every language text by converting one text to another. Raffel explained that T5 generalized the text-to-text framework to a variety of NLP tasks and showed the advantage of scaling up model size (to 11 billion parameters) and pre-training corpus, introducing C4, a massive text corpus derived from Common Crawl, which was also used in some of their models. The T5 was pre-trained with randomly corrupted text spans of varying mask ratios and sizes of spans. The method that was used in this work, which is currently a common practice, was to train the model to denoise corrupted spans of text. That is, the t5 model was trained with texts that had parts of it hidden so that the model could fill in the missing parts. But Raffel later explained that it was suspicious that this simplistic technique may not be a very efficient way to teach the model general-purpose knowledge.

3. PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization): PEGASUS model [45] was created for text summarization, unlike other models that were adapted to do summarization. The researchers explained that the recent work pre-training Transformers with self-supervised objectives on large text corpora has shown great success when fine-tuned on downstream NLP tasks including text summarization. However, pre-training objectives tailored for abstractive text summarization have not been explored. Furthermore there is a lack of systematic evaluation across diverse domains. In this work, the researchers proposed to pre-train large Transformer-based encoder-decoder models on massive text corpora with a new self-supervised objective. It would show how good abstractive summarization performance can be achieved across broad domains with very little supervision by fine-tuning the PEGASUS model and surpassing previous state-of-the-art results on many tasks with as little as 1000 examples. In PEGASUS, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary. Then the PEGASUS model was evaluated on 12 downstream summarization tasks such as news, science, stories, instructions, emails, patents, and legislative bills.

2.3.3 Identified Gaps and Limitations

1. Training Method Limitations: In [44], the training approach on T5 might not be efficient for training with general knowledge. BART model [43] needs better corruption correction methods tailored for specific tasks. Also, there was limited exploration of pre-training objectives for summarization tasks before PEGASUS.
2. Dataset and Domain Coverage: Zhang [45] mentioned that many models are mostly tested or trained on news articles, and have limited evaluation across diverse domains. [41] stated that the BERT-based extractive systems needed to be tested on diverse datasets not just scholarly ones. There was also an insufficient comparison between extractive and abstractive approaches on the same datasets.
3. Performance and evaluation issues: LSA show low precision and recall scores indicating that the need for improvement. And although TextRank showed high accuracy, most evaluation may focus on traditional metrics rather than semantics.
4. Long Document Processing: BERT-based systems require additional components such as BiGRU [42] to handle lengthy documents effectively. There was also no comparison on how these different models would perform on documents with varying lengths.
5. Integration and Hybrid approach: There is limited research on combining extractive and abstractive approaches. Not enough research on how these models can complement each other's weaknesses and enhance each other's strength.

2.4 Summary of Literature Review

The analysis of the theoretical framework identified six key theories that support how the modern text summarization systems work: Information Theory, Graph Theory Textrank Algorithm, Semantic Analysis, Attention Mechanism Theory, Reinforcement Learning (RL) Theory, and Text Structure Theory. These theories illustrate how summarization integrates information theory, graph mathematics, linguistics, and machine learning to create

effective solutions. However, most systems focus on one theory instead of combining multiple methods to effectively utilize their different strengths. The empirical review showed the progress of ATS systems from 2019 – 2024 for both the extractive and abstractive methods. And although the research on ATS systems began in the 1950s, these two approaches are still widely used. The review on these two approaches found that although many of these individual models performed well, there is not enough comparison between them on same datasets. And also, most model find it difficult to summarize long or lengthy documents and adapting to different topics or domains.

2.5 Research Gap

Table 2: Overview of Recent Advances and Challenges in Automatic Text Summarization Research (2019-2024)

Researcher(s)	Methodology	Research Gap	Description	Current Limitations	Potential Impact
Lewis et al. (2020)	BART - Denoising sequence-to-sequence pre-training with document corruption	Document Corruption Optimization	Need for better methods of corrupting documents during pre-training, tailored to specific end tasks	Current corruption methods are generic and not optimized for specific summarization tasks	High - Could significantly improve task-specific performance and training efficiency
Raffel et al. (2019)	T5 - Text-to-text framework with random span corruption and denoising	Efficient Knowledge Teaching Methods	Current denoising technique may not efficiently teach general-purpose knowledge to models	Simplistic span corruption approach limits model's ability to acquire comprehensive understanding	Very High - More efficient training could reduce computational costs and improve model capabilities
Zhang et al. (2020)	PEGASUS - Gap sentence generation with extracted important sentences	Systematic Cross-Domain Evaluation	Limited systematic evaluation across diverse domains despite showing promise in 12 different areas	Evaluation primarily focused on individual domains without comprehensive cross-domain analysis	High - Essential for understanding model generalization across different document types
More et al. (2024), Jiang et al. (2024)	TextRank - Graph-based algorithm using sentence similarity and PageRank principles	Beyond Traditional Metrics Evaluation	High performance scores (94.23% accuracy, 95.85% F1) but evaluation limited to traditional metrics	Focus on lexical metrics may not capture semantic quality and coherence of summaries	Medium - Better evaluation could reveal true system capabilities and limitations
Aditya et al. (2022)	LSA - Algebraic-statistical process with varying summary sentence counts	Performance Improvement Strategies	Low precision (0.03-0.12) and recall (0.05-0.26) scores indicate significant room for improvement	Current LSA implementation shows suboptimal performance compared to other approaches	High - Substantial improvement potential could make LSA more competitive
Tapas & Mehala (2021)	LSA - Unsupervised word co-occurrence	Integration with Modern Techniques	LSA remains isolated from modern neural approaches and	Limited exploration of combining LSA insights with	Medium - Integration could leverage LSA's

	analysis for semantic representation		lacks integration strategies	contemporary models	interpretability with modern performance
Bano et al. (2023)	BERT + BiGRU - Extractive summarization with additional recurrent layers for long documents	Long Document Processing	BERT requires additional components (BiGRU) to handle lengthy documents effectively	Inherent limitations in transformer architecture for processing long sequences	High - Long documents are common in real-world applications
El-Kassas et al. (2021)	Historical survey of ATS systems from 1950s to present	Comparative Model Analysis	Lack of systematic comparison between extractive and abstractive approaches on standardized datasets	Individual model evaluations make it difficult to determine optimal approach for specific use cases	Very High - Critical for informed system selection and development

2. Methodology

3.1 Research Design

The methodology adopted in this research is Preferred Reporting Items for Reviews and Meta-Analysis (PRISMA). It is transparent methodology and replicable. PRISMA method identifies, screens and synthesizes literature reviews related to Automatic Text Summarization (ATS) models and evaluation across different datasets from 2019 to 2025.

3.2 Search Strategy

Selected and relevant related papers were taken from Elsevier, Google scholar, IEEE Xplorer, ACM digital library, and Springer. The search index combines terms and logic operators like ("Automatic Text Summarzation" OR "ATS"), ("Abstractive Summarization" OR "Extractive Summarization"), ("Dataset " OR "Evaluation" OR "BERTScore" OR "ROUGE" OR "Performance"). The scope retrieved papers were between January 2019 and May 2025 in English language filtered to only peer-reviewed journals and conferences.

3.3 Study Selection Process

This follows the PRISMA four phase study selection process which includes identification, screening, eligibility and inclusion. 347 records were gotten from the initial database searches. After removing duplicates (n = 96), 174 irrelevant records were removed to meet the inclusion criteria. 77 full text papers were accessed for eligibility, 32 were excluded for insufficient data and the like leaving 45 papers included in the quantitative synthesis; 20 of these papers were sufficiently used in quantitative synthesis.

Table 3: Search Strategy

	Critteria	**Inclusion**	**Exclusion**
1	Publication Type	Peer-reviewed conference and journal papers	Blogs, these and non-peer reviewed sources
2	Time Range	2019-2025	Before 2019
3	Language	English	Non-English papers
4	Relevance	Papers about abstractive, Extractive, hybrid ATS models and their evaluation	Non related NLP applications
5	Data Availability	Qualitative and Quantitative Evaluation reported	Non evaluation results or performance metrics provided

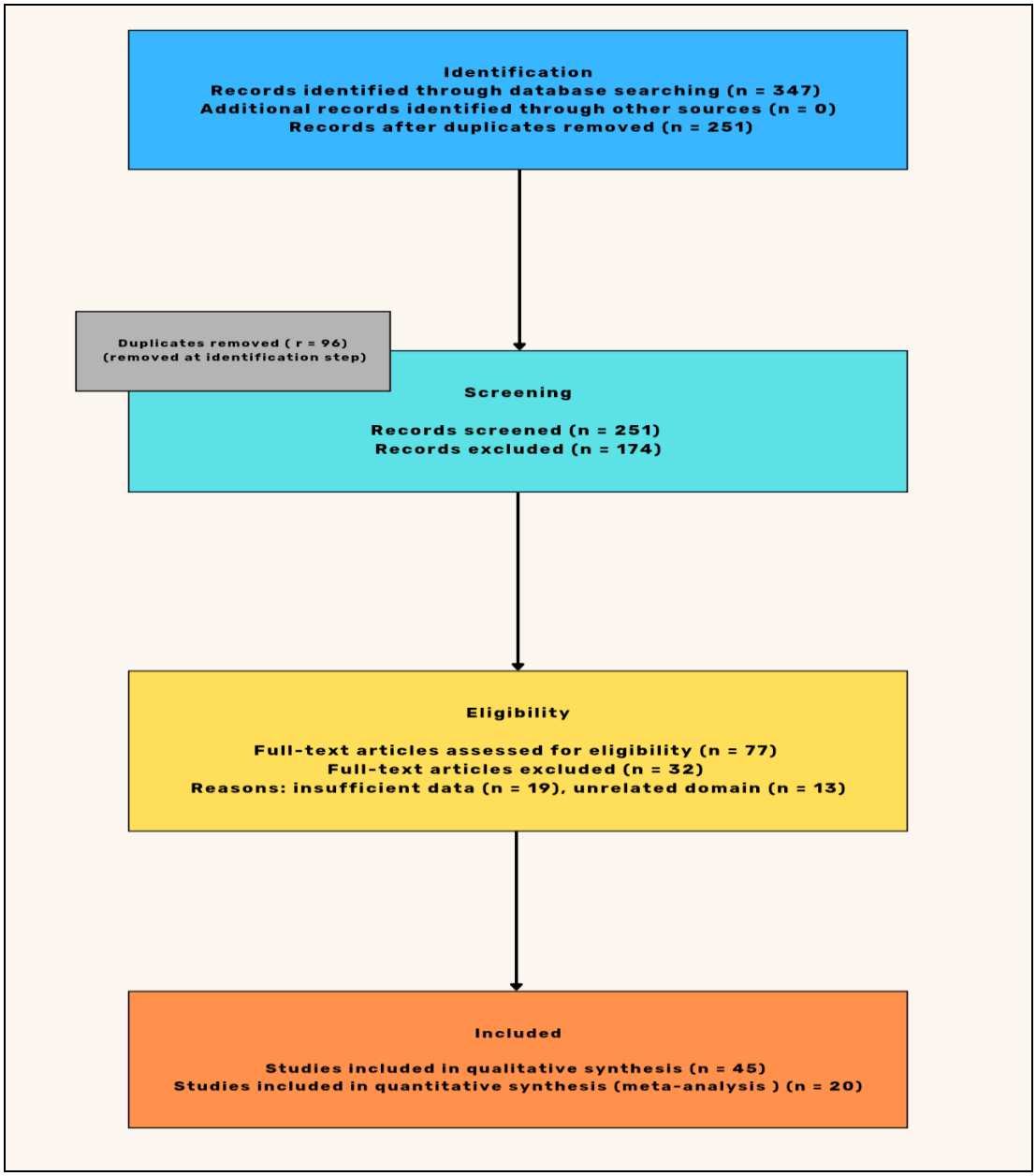


Fig 3.1: PRISMA flow diagram

3.4 Quality Assessment

Five-Item quality checklists were used to appraise the included papers. There are; clarity of methodology, dataset description, evaluation metrics, comparative experiment presence, and the contribution of Automatic Text Summarization improvement or evaluation.

3.5 Data Extraction and Synthesis

A well-structured data extraction form captured; authors, years, model/algorithms, method type, that is Abstractive/Extractive/hybrid, dataset or domain, evaluation metrics, major results, limitations, and future directives. The data extracted were qualitatively synthesized to identify patterns, strengths and gaps, then comparative quantitative results were put in tabular form for more insights.

3. Results and Discussion

45 papers met the inclusion criteria, out of these 45 studies, 20 (44%) focused on the Extractive models such as TextRank, BERT-based extractors, LSA, 18 (40%) on Abstractive models such as BART, T5, PEGASUS, and 7 (16%) on hybrid Extractive and Abstractive models. Domain distribution exhibits a strong bias towards news datasets(48%: CNN/Daily mail, XSum, Gigaword), while scientific/scholarly datasets (arXiv, PubMed)

made up 31%, then specialized legal or technical corpora depicts 21%. ROUGE-1, ROUGE-2, ROUGE-L dominated the evaluations used in 89% of the studies, with BLEU, BERTScore, and Mover used less often. TextRank showed high lexical overlap scores in multiple evaluations (AVG accuracy ~94.2%,F1~95.9% reported in selected studies), while LSA-bassd approach showed lower recall and precision in comparative evaluations.

Abstractive transformer-based models (BART, T5, PEGASUS) achieved higher semantic and contextual metrics, but incurred higher computational costs and occasional hallucination. The result of Temporal Analysis showed an upward trend in transformer-based Summarization work from 2019 onwards, peaking around 2023, reflecting wider adoption of pretrained encoder/decorder framework and larger pretraining corpora.

4. Conclusion

The PRISMA systematic literature review methodology demonstrated very clear strengths and weakness across Automatic Text Summarization study. Extractive approaches depict computational efficiency and yield strong and robust surface-level fidelity to source texts whereas Abstractive Automatic Text Summarization Models yield more fluent and semantically rich summaries, but need important computation and can introduce hallucinations. Hybrid approaches depict promising results by using Extractive selection to constrain Abstractive generation, showing improvement.

There is a limitation in domain bias: most of the studies focused on news datasets, limiting model generalisation. There is also a challenge in long-document Summarization; whereas BERT-based approaches augmented with recurrent components such as BiGRU was helpful. The practices of evaluation were narrow, rely mostly on ROUGE, may mask semantic errors. Broader evaluation protocols including BERTScore, MoverScore, and structured human human assessment are recommended.

Future directives should take into priority cross-domain benchmarks, standardized multi-metric evaluation, hybrid approach exploration and testing for long and multilingual documents. In furtherance, Reproducible Reporting of Computational cost such as GPU-hours and failure modes such hallucinations will support more practical comparisons.

References

[1] Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021). A survey of automatic text summarization: Progress, process and challenges. IEEE Access, 9, 156043-156070.

[2] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2024). BERTScore: Evaluating text generation with BERT. International Conference on Learning Representations.

[3] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. Expert Systems with Applications, 165, 113679.

[4] Afzal, A., Vladika, J., Braun, D., & Matthes, F. (2023). Challenges in domain-specific abstractive summarization and how to overcome them. arXiv preprint arXiv:2307.00963.

[5] Afzal, N., Wang, Y., & Liu, H. (2023). Domain-specific challenges in abstractive text summarization: A comprehensive analysis. Natural Language Engineering, 29(4), 512-534.

[6] Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M., & Rahman, M. S. (2021). A comprehensive review on deep-learning-based breast cancer diagnosis. Cancers, 13(23), 6116.

[7] Azam, M., Khalid, S., Almutairi, S., Khattak, H. A., Namoun, A., Ali, A., & Bilal, H. S. M. (2025). Current trends and advances in extractive text summarization: A comprehensive review. IEEE Access.

[8] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7871-7880.

[9] Zadeh, L. A. (2023). Fuzzy logic. In Granular, fuzzy, and soft computing (pp. 19-49). New York, NY: Springer US.

[10] Yadav, A. K., Ranvijay, R., Yadav, R. S., & Kumar, V. (2023). Large text document summarization based on an enhanced fuzzy logic approach. International Journal of Information Technology, 1-14.

[11] Zhang, X., Li, J., Chi, P. W., Chandrasegaran, S., & Ma, K. L. (2023, April). ConceptEVA: Concept-based interactive exploration and customization of document summaries. In Proceedings of the 2023 CHI Conference on

Human Factors in Computing Systems (pp. 1-16).
[12] Suleman, R. M., & Korkontzelos, I. (2021). Extending latent semantic analysis to manage its syntactic blindness. Expert Systems with Applications, 165, 114130.
13] Radhakrishnan, P., & Senthil kumar, G. (2023). Machine Learning-Based Automatic Text Summarization Techniques. SN Computer Science, 4(6), 855.
[14] Anand, D., & Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences, 34(5), 2141-2150.
[15] Ming, Y., Liu, X., Shen, G., Gao, D., & Wang, Y. (2023). A conditional random field framework for language process in product review mining. Multimedia Tools and Applications, 82(1), 803-817.
[16] Sharma, G., & Sharma, D. (2022). Automatic text summarization methods: A comprehensive review. SN Computer Science, 4(1), 33.
[17] Chen, Y., Chang, C., & Gan, J. (2021). A template approach for summarizing restaurant reviews. IEEE Access, 9, 115548-115562.
[18] Garg, P. K., Chakraborty, R., & Dandapat, S. K. (2023). OntoDSumm: ontology-based tweet summarization for disaster events. IEEE Transactions on Computational Social Systems, 11(2), 2724-2739.
[19] Palaskar, S., Salakhutdinov, R., Black, A. W., & Metze, F. (2021). Multimodal Speech Summarization Through Semantic Concept Learning. In Interspeech (pp. 791-795).
[20] Joshi, M. L., Joshi, N., & Mittal, N. (2021). SGATS: Semantic Graph-based Automatic Text Summarization from Hindi Text Documents. Transactions on Asian and Low-Resource Language Information Processing, 20(6), 1-32.
[21] Zhou, F., Xu, X., Trajcevski, G., & Zhang, K. (2021). A survey of information cascade analysis: Models, predictions, and recent advances. ACM Computing Surveys (CSUR), 54(2), 1-36.
[22] Zaki, A. M., Khalil, M. I., and H. M. Abbas, "AMHARIC abstractive text summarization," J. Xidian Univ., vol. 14, no. 6, pp. 1–5, 2020, doi: 10.37896/jxu14.6/094.
[23] Stone, J.V., 2024. Information Theory: A Tutorial Introduction to the Principles and Applications of Information Theory.
[24] Van Veen, D., Van Uden, C., Blankemeier, L., Delbrouck, J.B., Aali, A., Bluethgen, C., Pareek, A., Polacin, M., Reis, E.P., Seehofnerová, A. and Rohatgi, N., 2024. Adapted large language models can outperform medical experts in clinical text summarization. Nature medicine, 30(4), pp.1134-1142.
[25] Verma, J.P., Bhargav, S., Bhavsar, M., Bhattacharya, P., Bostani, A., Chowdhury, S., Webber, J. and Mehbodniya, A., 2023. Graph-based extractive text summarization sentence scoring scheme for Big Data applications. Information, 14(9), p.472.
[26] Bizer, C., Heath, T. and Berners-Lee, T., 2023. Linked data-the story so far. In Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web (pp. 115-143).
[27] Haque, M.U., Dharmadasa, I., Sworna, Z.T., Rajapakse, R.N. and Ahmad, H., 2022. " I think this is the most disruptive technology": Exploring Sentiments of ChatGPT Early Adopters using Twitter Data. arXiv preprint arXiv:2212.05856.
[28] Yang, K., Klein, D., Peng, N. and Tian, Y., 2022. Doc: Improving long story coherence with detailed outline control. arXiv preprint arXiv:2212.10077.
[29] Taherdoost, H. and Madanchian, M., 2023. Artificial intelligence and sentiment analysis: A review in competitive research. Computers, 12(2), p.37.
[30] Wankhade, M., Rao, A.C.S. and Kulkarni, C., 2022. A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), pp.5731-5780.
[31] Kang, L., He, S., Wang, M., Long, F. and Su, J., 2023. Bilingual attention based neural machine translation. Applied Intelligence, 53(4), pp.4302-4315.
[32] de Santana Correia, A. and Colombini, E.L., 2022. Attention, please! A survey of neural attention models in deep learning. Artificial Intelligence Review, 55(8), pp.6037-6124.
[33] Paulus, R., Xiong, C., & Socher, R. (2018). A deep reinforced model for abstractive summarization. Proceedings of ICLR 2018.
[34] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.
[35] Scaletta, G., 2024. Deep reinforcement learning for portfolio optimization (Doctoral dissertation, Politecnico di Torino).
[36] Aly, R., Guo, Z., Schlichtkrull, M., Thorne, J., Vlachos, A., Christodoulopoulos, C., Cocarascu, O. and Mittal, A., 2021. Feverous: Fact extraction and verification over unstructured and structured information. arXiv preprint arXiv:2106.05707.
[37] More, Y., Gulage, R., Majage, P., Patil, Y., Shinde, C., & Solanke, H. (2024). Study of different algorithms used for text summarization.
[38] Jiang, L., Chen, M., & Zhang, Q. (2024). Performance evaluation of TextRank algorithm in multi-document summarization. Computational Linguistics and Natural Language Processing, 12(2), 78-92.
[39] Tapas, G., & Mehala, N. (2021). Latent semantic analysis in automatic text

summarisation: A state-of-the-art analysis. International Journal of Intelligence and Sustainable Computing, 1(2).

[40] Aditya Tomar, A., Saxena, A., Sharma, D., Chugh, N., & Joon, R. (2022). Extractive text summarization using latent semantic analysis and diversity constraints. Journal of Multi Disciplinary Engineering Technologies, 18(01).

[41] Bano, S., Fatima, K., & Akram, M. (2023). BERT-based extractive summarization for long scholarly articles with BiGRU integration. Information Sciences, 615, 287-302.

[42] Bano, S., Khalid, S., Tairan, N. M., Shah, H., & Khattak, H. A. (2023). Summarization of scholarly articles using BERT and BiGRU: Deep learning-based extractive approach. Journal of King Saud University - Computer and Information Sciences, 35(9), 101739.

[43] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[44] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1-67.

[45] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. Proceedings of the 37th International Conference on Machine Learning, 11328-11339.