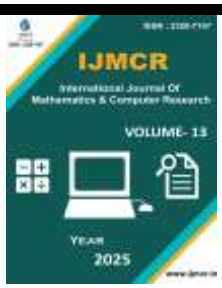
International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 13 Issue 07 July 2025, Page no. –5430-5442

Index Copernicus ICV: 57.55, Impact Factor: 8.615

DOI: 10.47191/ijmcr/v13i7.12



Development of an Enhanced Naïve Bayes and Association Rule Mining Hybridized Algorithms for Multi-Documents Summarization Approach

Jelili Olalekan AMOO[1], Professor Joshua Ayobami AYENI[2], Mayowa Oyedepo OYEDIRAN PhD[3], Damilola Nnamaka AJOBIEWE PhD[4], Enitan Olabisi ADEBAYO[5]

[1,4,5]Department of Computer Science, School of Secondary Education (Science Programmes), Federal College of Education (Special), P.M.B 1089, Oyo

[2,3]Department of Computer Sciences, Faculty of Natural Sciences, Ajayi Crowther University, P.M.B 1066, Oyo.

ARTICLE INFO

Published Online:
18 July 2025

ABSTRACT

The challenge of summarization arises from the fact that computers lack the ability to comprehend human language and the emotions conveyed within it. To address this challenge, several machine learning models have been trained, tested and used in a manner that encompasses crucial information. Researchers were drawn to various aspects of artificial text summarization due to its ability to efficiently extract relevant information in a shorter timeframe and its uses encompasses the generation of concise summaries from many sources such as emails, news articles, mobile news, and corporate information. One hundred thousand (100,000) records of case related to heart disease were fetched from Kaggle online Dataset for Heart Disease, the data were contained in different types of documents They were fed into Naïve Bayes algorithm for proper data cleansing and treatment before classification into 70% training and 30% testing because of the status of the model. Association rule mining was used to collaborate to achieve very reliable summaries. The enhance Naive Bayes classifier was used as initiator to provide set of relevance scores by analyzing word frequencies and other document features. These scores were then used as part of the broader summarization framework, the two output were combined to generate a comprehensive and meaningful summary. Performance evaluation were carried out. The performance evaluation metrics adopted in the study measures the model's accuracy, precision, recall and F1 score, the model, combines both enhanced Naive Bayes (e-NB) and Association Rule Mining (ARM), delivering superior performance across all the metrics.

Overall performance of (e-NB) - ARM outshines NB-ARM in all metrics, including accuracy, precision, recall, F1-score, and time complexity, the hybridized model improves predictive accuracy by combining the probabilistic approach of Naive Bayes with the insightful patterns generated by Association Rule Mining**,** allowing for more robust predictions, accurate and interpretable insights of heart disease making it the preferable choice between the two models. Therefore, adopting the enhanced hybrid Naive Bayes and Association Rule Mining (e-NB) - ARM system for multi-document summarization is the best.

Conclusively, this approach not only enhances the efficiency and quality of multi-document summarization in high-volume textual data environments but also has applications in diverse fields. This study has contributed to the growing need for advanced text summarization methods, facilitating faster access to essential information and supporting more informed decision-making across disciplines.

Corresponding Author:
Jelili Olalekan AMOO

KEYWORDS: Association Rule Mining, Multi Document, Naïve Bayes, Summarization.

1.0 INTRODUCTION

1.1 Background to the Study

These days, the quantity of material that is readily available on the internet is of such a tremendous magnitude. Users now have access to a wealth of information, but there are significant reading hurdles that make it more challenging to rapidly retrieve useful information (Sanchez-Gomez, et al., 2022). Making summaries is one technique to address this information overload issue. Automatic Text Summarization (ATS) is a technique for automatically condensing a document or a collection of documents into a more manageable length by keeping the essential details and the main ideas while removing any irrelevant or redundant information (Anand and Wagh 2022). Without having to read the full page, users of ATS systems may rapidly understand the key points of a document. Ma, et al. (2020) stated that users will benefit from the automatically created summaries since they will save them a significant amount of time and efforts. Likewise, any text document's summary makes it easier to comprehend the ideas and draw useful conclusions. The term "manual summarization" refers to a summary produced by a person, whereas the term "automatic text summarization" (ATS) refers to a summary produced by a computer. ATS are often broken down into many methodologies. Some methods utilize the input documents to classify the text before using it to create the summary.

A single-document summarization (SDS), which creates the summary from a single document, or a multi-document summarization (MDS), which extracts the summary from a collection of documents, may be used to summarize the number of input documents (Waly and Gomaa, 2022). It is considerably easier to extract pertinent lines from a single summary. The summary is coherent if the order of the sentences that were chosen stays the same as it did in the original text. Choosing which phrases to extract from the documents and how to present them (in what order) are challenges for MDS, according to Widyassari, et al., (2022). Coherence is another challenge. There is a lot of duplication since the summarized texts address the same issues. The final summary must include no duplicate material. On the other hand, according to Abid (2022), SDS had no redundant data.

Depending on the task at hand, text summarization methods may potentially be either extractive or abstractive. The most important sentences in the source texts were to be found and selected using an extractive technique, which include leaving the sentences precisely as they were. Instead than just copying the most significant phrases from the book, abstractive summarizing is a strategy for constructing a summary of a text based on its core ideas. Although, abstractive summary produces summaries that are more human-like than extractive summarization, it is more challenging to execute since it requires deep Natural Language Processing (NLP) comprehension methods, according to Bidoki, et al., (2020). Text summarization is a fundamental component of natural language processing (NLP). The use of automation in this process will result in the generation of summaries without the need for human involvement. The challenge of summarization arises from the fact that computers lack the ability to comprehend human language and the emotions conveyed within it. To address this challenge, several machine learning models are used.

As a matter of emphasis from the above, it is essential to know that the two main categories of summarization methods are extractive and abstractive summarization. In contrast to an abstractive summarizing, which creates a concise and informative summary, an extractive summarization deals with choosing the key lines from the source materials and integrating them into a summary. Single and multi-document summarization (MDS) is a different method of categorization. While MDS employs many documents relating to a single subject, single document summarization uses only one document for summary production. Comparing single document summarization to MDS, it is significantly simpler. However, MDS is mostly used in large-scale text retrieval systems. By eliminating repetition in the content, multi-document summaries thereby include the most important details from each document. Since MDS creates duplicate information in several papers on the same topic, it is more difficult than single document summarization. The production of multi-document summaries is significantly influenced by compression ratio. Redundancy and coherence make MDS more complicated (Aote, et al., 2023).

There is a strong requirement to summarize a great set of multiple documents in a short period of time. In this research, a new model that depends on enhanced Naïve Bayes and association rules mining algorithms was developed for the task of summarizing health records of a dataset which predict probability of heart disease in patients because it has been discovered that the common leading causes of death in the developed world presently is heart disease, therefore, it becomes inevitable to acquire and summarize vital related documents that contains information about this ailment to properly guide health practitioners and other scientists to further prevent the risks of having a heart attack or stroke probably as a result of not having access to relevant and useful summarized information available on time.

1.2 Statement of Problem

The major limitation of current summarization models is their inability to remain factually consistent with the respective input document. Summary inconsistencies are diverse from inversions (i.e., negation) to incorrect use of an entity (i.e., subject, object swapping), or hallucinations (i.e., introduction of entity not in the original document). Recent studies have shown that in some scenarios, even state-of-the-

art pre-trained language models can generate inconsistent summaries in more than 70% of all cases (Pagnoni et al., 2021). This has led to accelerated research around summary inconsistency detection.

There have been various methodologies and approaches; but still, some limitations are being experienced such as: low coverage of important parts of source documents which pose inaccurate extraction of essential sentences from them. Also, the shortcoming of inconsistency and lack of coherency among the few captured sentences is alarming. Hence, the need to develop a robust model of an enhanced Naïve Bayes and Association Rule Mining (e-NB) - ARM out of the combination power of two chosen algorithms (NB-ARM) that will solve these problems of non-contextually accurate and coherent summary so that a complete useful, meaningful and reliable information could be realized as a summary.

1.3 Aim and Objectives of the Study

The aim of this study is to develop an enhanced hybridized system for multi document summarization, and the specific objectives are to:

i. formulate an enhanced hybridize Naïve Bayes and Association rule mining (e-NB) -ARM for multi documents summarization;
ii. design a multiple textual documents super-summarizer using (i);
iii. develop the system designed in (ii) using Python programming language;
iv. evaluate the performance of the summarization system using performance metrics based on time complexity, accuracy, precision, recall and F1-Score; with ROUGE and then compare with an existing MDS system.

1.4 Scope of the Study

The scope of this study is limited to the development of an enhanced system for identifying the most important and meaningful information in a set of related textual documents and compressing them into a shorter version while preserving its overall original meanings. The research is focused on multi documents summarization with enhanced Naïve Bayes algorithm (e-NB) machine learning and Association Rule Mining (ARM) summarization system for an optimal and reliable results by information seekers.

1.5 Significance of the Study

The significance of this study lies in advancing the field of automated multi-document summarization by developing a robust hybrid approach that integrates an enhanced Naive Bayes classifier with optimized Laplace smoothing and Association Rule Mining (ARM). Current summarization techniques often struggle to maintain a balance between precision i.e. selecting highly relevant sentences and, recall which is capturing a comprehensive set of important ideas. This proposed hybrid approach addresses these challenges by combining the strengths of machine learning and pattern mining to improve relevance scoring and thematic consistency across summaries.

2.0 LITERATURE REVIEW

2.1 Text Summarization

Text summarization is the process of condensing text into a summary that retains key information from the source text (El-Kassas et al., 2021). The mainstream approaches to text summarization based on PLMs are either extractive or abstractive. Extractive summarization selects a subset of sentences from the source text and concatenates them to form the summary Liu et al., (2019). In contrast, abstractive summarization generates the summary automatically from the abstract representation of input texts (See et al., 2017, Zhang et al., 2024). As abstractive summarization is more related to text generation, we only discuss abstractive summarization in this section.

2.1.1 Document summarization.

Document is a widely-used literary form, such as news, opinions, reviews, and scientific papers. PLMs, such as UniLM (Bao et al., 2020). Masked Sequence to Sequence pre-training for language generation **(MASS)** (Song et al., 2020), pre-Trained Text-To-Text Transformer (**T5)** (Raffel et al., 2020), Bidirectional Autoregressive Transformer **(BART)** (Lewis et al., 2020) and Pre-Training with Extracted Gap-Sentences for Abstractive Summarization **(PEGASUS)** (Zhang et al., 2020), can be directly fine-tuned for document summarization. During pre-training, these models learn to predict the masked important sentences in the input document based on the remaining ones, which shares the similar idea of summarization. Without directly generating summaries, several studies first extracted keywords, key sentences or relations as guidance and then combined these with PLMs for generation. **CIT** employed **RoBERTa** Liu et al., (2019) to extract the important words and sentences from the input document. In addition, topic models are used to capture the global topic semantics of the document, which can be integrated into the summarization model Nguyen (2021). **GSum** Dou et al., (2021) proposed a general framework taking different kinds of guidance signals into the generation model, including keywords, triples, highlighted sentences and retrieved summaries. Apart from external guidance, several tricks can be applied to document summarization.

2.1.2 Dialogue summarization.

Dialogues, such as chat and medical conversation, consist of multiturn utterances by two or more individuals. Hence, it is critical to capture the semi-structured dialogue content and users' interactions in dialogue Feng et al., (2021). For dialogue summarization, it is straightforward to directly reuse document summarization models. Zhang et al., (2021) first truncated the dialogue text into several chunks, then summarized each chuck into partial summaries, and finally rewrote these partial summaries into a complete summary.

Meanwhile, several studies also explored some specific characteristics of dialogue for improving dialogue summarization. Considering the low information density, topic drifts and frequent coreferences of dialogue (Feng et al., 2021), some researchers conducted auxiliary tasks to extract intrinsic information of dialogue. Feng et al., (2021) utilized DialoGPT Zhang et al., (2020), a PLM specially designed for dialogue, to automatically extract keywords, detect redundant utterances and divide a dialogue into topically coherent segments.

2.2 Classification of Automatic Text Summarization

There are different classifications for an automatic text summarization (ATS) system based on its input, output, purpose, length, algorithms, domain, and language. There are many other factors that can be considered while discussing the classification of summarization. Different researchers have also considered some other factors of categorizations apart from the above mentioned.

2.2.1 Based on summarization methods

Based on methods that show how summaries are produced, i.e. Just picking up sentences from the source text or generating new sentences after reading source text or a combination of both, summarization can be divided into three types:

i. Extractive automatic text summarization:

Extractive text summarization is the strategy of concatenating on extracting summary from a given corpus (Bidoki et al., 2020). Extractive summaries according to Am, (2021) are formulated by extracting key text segments (sentences or passages) from the text, based on statistical analysis of individual or mixed surface level features such as word/phrase frequency, location or cue words to locate the sentences to be extracted. The "most important" content is treated as the "most frequent" or the "most favorably positioned" content. Such an approach thus avoids any efforts on deep text understanding. They are conceptually simple, easy to implement. Extractive text summarization process according to Myla et al. (2024), can be divided into two steps: Pre-processing step and Processing step.

Pre-processing is structured representation of the original text. It usually includes:

i. Sentences boundary identification: In English, sentence boundary is identified with presence of dot at the end of sentence.
ii. Stop-word elamination: Common words with no semantics and which do not aggregate relevant information to the task are eliminated.
iii. Stemming: The purpose of stemming is to obtain the stem or radix of each word, which emphasize its semantics, is to obtain the stem or radix of each word, which emphasize its semantics.

In Processing step, features influencing the relevance of sentences are decided and calculated and then weights are assigned to these features using weight learning method. Final score of each sentence is determined using Feature-weight equation. Top ranked sentences are selected for final summary.

Problems with the extractive summary are:

1. Extracted sentences usually tend to be longer than average. Due to this, parts of the segments that are not essential for summary also get included, consuming space.
2. Important or relevant information is usually spread across sentences, and extractive summaries cannot capture this (unless the summary is long enough to hold all those sentences).
3. Conflicting information may not be presented accurately.
4. Pure extraction often leads to problems in overall coherence of the summary—a frequent issue concerns "dangling" anaphora. Sentences often contain pronouns, which lose their referents when extracted out of context.

In Processing step, features influencing the relevance of sentences are decided and calculated and then weights are assigned to these features using weight learning method. Final score of each sentence is determined using Feature-weight equation. Top ranked sentences are selected for final summary.

ii. Abstractive automatic text summarization:

Abstractive text summarization involves paraphrasing the given corpus and generating new sentences (Zhang et al., 2019).

Abstractive summarization concentrates on the most critical information in the original text and creates a new set of sentences for the summary. This technique entails identifying key pieces, interpreting the context, and re-creating them in a new way. Due to the difficulty of both extracting relevant information from a document as well as automatically generating coherent text, abstractive summarization has been considered a more complex problem than extractive summarization. The abstractive summarization method works well with deep learning models like the seq2seq model, LSTM, etc., along with popular Python packages (Spacy, NLTK, etc.)

Problems with the abstractive summary are:

The biggest challenge for abstractive summary is the representation problem. Systems' capabilities are constrained by the richness of their representations and their ability to generate such structures-systems cannot summarize what their representations cannot capture. In limited domains, it may be feasible to devise appropriate structures, but a general-purpose solution depends on open-domain semantic analysis. Systems that can truly "understand" natural language are beyond the capabilities of today's technology. Summary evaluation (Curiel et al., 2020), is a very important aspect for text summarization. Generally, summaries can be evaluated using intrinsic or extrinsic measures. While intrinsic methods attempt to measure summary quality using human evaluation and extrinsic methods measure the same

through a task-based (Curiel et al., 2020), performance measure such the information retrieval-oriented task.

iii. Hybrid Automatic Text Summarization

It combines both extractive and abstractive methods. It means extracting some sentences and generating a new one from a given corpus. (Raza and Shahzad, 2024).

When discussing very precious approaches of Automatic text summarization that are Extractive and Abstractive, both come with their pros and cons. Extractive summarization is comparatively easier to implement than abstractive summarization, but extractive summarization is not as efficient as user perception. Combining these methods by strengthening their pros and weakening their cons leads to hybrid methods for text summarization.

Experiments were done on summarization until 1990 were focused on just extracting (reproduced) the summaries from original text rather than abstracting (newly generated). SUMMRIST system (Du et al., 2024) was developed with the help of NLP techniques. We can develop a multi-lingual summarizer by modifying some parts of the structure.

Semantic and statistical features combine extracting and abstracting. The authors of Bhat et al., (2022) used emotions of the text as a semantic feature. Emotions play a significant role in defining the user's emotional affinity, so lines with implicit emotional content are crucial to the writer and should be included in the summary. The extracted summary is then put into the Novel language generator, a hybrid summarizer that combines WordNet, Lesk algorithm, and POS to transform extractive summary into an abstractive summary.

The proposed approach combines the concept of statistical measure, sentiment analysis and finally uses the concept of fuzzy logic to select sentence. Hybrid text summarization therefore combines an approach for producing a summary efficiently.

2.2.2 Based on summarization algorithms

Based on the actual algorithm that is used to generate the summaries, the ATS system is divided into two types as given below:

i. **Supervised**: The supervised summarizer needs to train the sample data by labeling the input text document with the help of human efforts (Mridha et al., 2021)

ii. **Unsupervised**: In the Unsupervised summarizer training phase is not needed (Alami et al., 2021).

To select important content from documents in a supervised system, training data is required. Training data is a large volume of labelled or annotated data is required for learning techniques. These systems are approached as a two-class classification issue at the sentence level, with positive samples being sentences that belong to the summary and negative samples being sentences that do not belong to the summary. On the other hand, unsupervised systems do not require any training data. They create the summary by just looking at the documents they want to look at for summarization. As a result, they can be used with any newly observed data without needing extra adjustments. These systems use heuristic methods to extract relevant sentences and construct a summary. Clustering is used in unsupervised systems (Belwa et al., 2021).

2.3 Machine Learning Method

The idea behind machine learning is to use a training set of data to train the summarization system, which is modeled as a classification problem. Sentences are classified into two groups: summary sentences and non-summary sentences (Guan et al., 2020). The probability of choosing a sentence for a summary is estimated according to the training document and extractive summaries (Shukla et al., 2023). Some of the common machine learning methods used for text summarization are naïve Bayes, artificial neural network, and fuzzy logic (Alias et al., (2023)

2.3.1 Naïve Bayes (NB) method

Naïve Bayes is a supervised learning method. In text summarization, the Naïve Bayes classification, introduced by (D'Silva and Sharma, 2023), considers the selection of a sentence as a classification problem. By this classification, each sentence is put in a binary class to determine whether it will be included in the summary or not. The features that are used in this method are word frequency, uppercase words, length of sentence, position in paragraph, and structure of phrase. By considering **k** features and using the Bayes rule, the probability that sentence **s** is included in summary **S** is defined as follows:

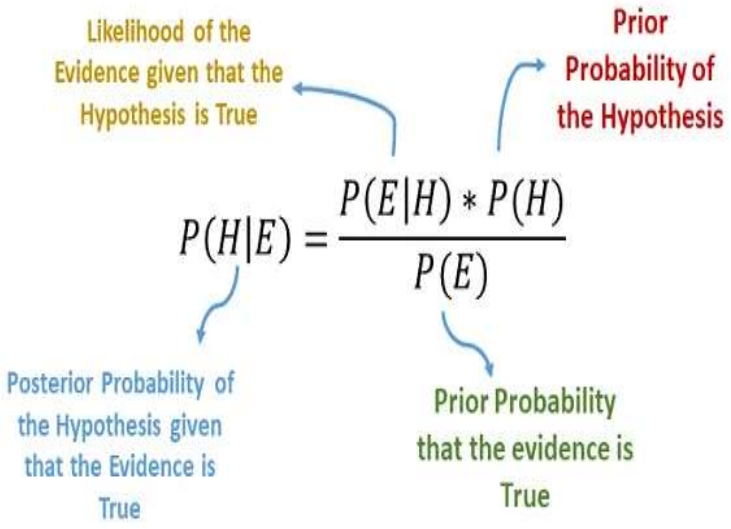


Illustration of Naïve Bayes
Source: Hassan et al., (2022)

2.3.2 Artificial Neural Network (ANN) method

The artificial neural network is a computational model used in computer science and other research areas for solving problems based on machine learning approaches. (Abdelaleem et al., 2020) used artificial neural networks for summarizing news articles to select sentences in an extractive summarization. There are three phases of the proposed approach: neural network training, feature fusion, and sentence selection. The training phase identifies the types of sentences that should be presented in the document summary. A human reader does this, and the system learns the pattern of summary sentences. After training the artificial neural network, the relation among features should be determined. In training the machine, the following seven features are considered:

i. Paragraph follows title
ii. Paragraph location in document
iii. Sentence location in paragraph
iv. First sentence of paragraph
v. Sentence length
vi. Number of thematic words in the sentence
vii. Number of title words in the sentence

This step consists of two phases:
i. Removing uncommon features, and
ii. Removing the effects of common features.

Therefore, this step generalizes the important features that must exist in the summary sentences. After the training and generalizing the network, this system can be used to select important sentences for the summary.

2.3.3 Graph based methods

Graph-based methods are completely unsupervised method in which a graph is constructed consisting of vertices and edges Roy and Kundu (2023). In case of single document summarization, sentences are represented as vertices; whereas in multi-document summarization, each document is represented as vertices. If two vertices are related to each other (share common information), they are connected using edges. Edges can be weighted or unweighted, and graphs can be directed or undirected. Graph-based approaches according to Yadav et al., (2024) rely solely on the text to be summarized and require no training data.

Approaches like hLDA can exploit repetitions both at the word and at the sentence level Razumovskaia, et al., (2021). Graph-based methods form another powerful class of approaches which combine repetitions at the word and at the sentence level. They were developed to estimate sentence Importance based on word and sentence similarities (Belwal, Rai, and Gupta 2021). One of the most prominent examples is LexRank (Ni et al., 2021). A similarity graph G (V, E) is constructed where V is the set of sentences and an edge e_{ij} is drawn between sentences v_i and v_j if and only if the similarity between them is above a given threshold. Then, sentences are scored according to their PageRank score in G. A significant body of research was dedicated to tweak and improve various components of graph-based approaches. For example, one can investigate different similarity measures (Alanzi, and Alballaa, 2023).

2.3.4 Cluster based methods

A cluster of documents can be considered as a network of sentences that are related to each other (Kumar et al., 2021). Some sentences are more similar to each other while some others may share only a little information with the rest of the sentences. The sentences that are similar to many of the other sentences in a cluster are more central (or salient) to the topic. Clustering based summarization uses some of the similarity measures like cosine similarity, sentence similarity, Jaccard similarity, support vector machine, etc.

2.3.5 Deep learning-based methods

Deep learning models can solve complex non-linear relationships (Dima and Arafat 2020). It has been widely used

in many domains such as computer vision and natural language processing. Better performance can be achieved by utilizing deep learning-based approach for multi-document text summarization (Ma et al., 2022).

2.3.6 Recurrent Neural Network (RNN) model

Inui et. al, (2019) proposed a hierarchical Recurrent Neural Network (RNN) model for extractive subtopic-driven multi-document text summarization. RNN models are best in handling sequential data. It is assumed that the documents to be summarized belongs to the same topic but can contain different subtopics. These sub topics can be present across several input documents. Sentence salience is calculated by considering both subtopic salience and relative sentence salience. Attention mechanism is used to estimate subtopic salience. Similarly, for each subtopic, relative sentence salience is estimated by using the contextual information. Sentences are ranked by multiplying these two values and top ranked sentences are extracted for summary generation. The model is evaluated on two datasets – RA-MDS and DUC2004 and has achieved a ROUGE score of 0.456 and 0.443 respectively.

2.3.7 Transformer model

The transformer architecture is proposed by Google in 2017 which makes use of attention layer in the encoder-decoder model. Each of the encoder-decoder layers is connected to an attention layer which helps in remembering the position and sequence of words in the input sequence and assigns a weight to it. Hugging face, pipeline, BART (Bi directional and Auto Regressive Transformers), T5 (Text-to-Text Transfer Transformer) and PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models) are different models that are based on Transformers. Anushka et.al (2021) did a comparison study on these pre-trained models. They used the BBC news dataset for the analysis and found out that the T5 model (ROUGE score: 0.47) outperformed all other models.

3.0 METHODOLOGY

3.1 Research Approach

The following research approach outline under each specified heading for improving Naive Bayes (NB) and Association Rule Mining (ARM) algorithms in a Multi-Document Summarization (MDS) system.

The following steps were taken for this research

Data acquisition

Data preprocessing (Cleaning and normalization)

Data classification

Feature selection

3.2 Data Acquisition

The dataset for the study was harvested from Kaggle online Dataset for Heart Disease. The dataset has about 100,000 related cases contained in different types of documents which have the following features: file name, file size, file type, date of creation and date of last modification.

The dataset used for this research is secondary. It is obtained from Kaggle online dataset for Heart disease records dated from 1988 to 2023 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them. The "target" field refers to the presence of heart disease in the patient. It is integer valued; 0 = no disease and 1 = disease.

Stages of data collection

Let

D: Dataset

N: Number of documents in the dataset

D_i: Document i in the dataset

T_i: Text content of document i

Therefore,

for collection and aggregation

$D = (D_1, D_2, \ldots\ldots, D_n)$

for document representation

T_i = Representation (D_i)

for labelling of each document (supervised task)

Label (D_i) represents the Class or category of document

for data splitting

Partitioning was done on the acquired data into training (70%) and testing (30%). The enhance Naïve Bayes algorithm used to applied on the partitioned dataset to summarize the multiple document for health specialists uses and decision making.

Split (D) = D_{train}, D_{test}

Represents the split dataset into training and test sets for model training and evaluation

3.2.1 Data preprocessing

Data preprocessing is the processing of data cleaning, removal of empty space and data normalization and that will be employed in the study.

i. Data cleaning

The acquired dataset from Kaggle were cleaned by removing excess, duplicate and other unwanted data. Although these data have a significant degree of cleanse and accuracy, but this step is necessary to be taken in order to improve the suitability of the purpose for which these data have been adopted. However, these data cleaning steps were followed:

The dataset needed for the analysis was fetched, stored and organized. Then, the preprocessing was done to clean the collected dataset by removing the duplicated, irrelevant and inconsistent data among the acquired dataset.

ii. Data normalization

Data normalization was among the set of steps of data preprocessing; while performing machine learning algorithms in the dataset. Data normalization was performed on the log file to achieve a set of clean data, organized the data to appear similar across all records and fields, and increased the cohesion of entry types to lead to the cleansing,

generation, segmentation and higher quality data and eliminated the unstructured data and redundancy (duplicates) to ensure logical data storage.

for data pre-processing

Ti = Pre-process (T_i)

i.e. cleaning, tokenization, stemming and removal of stop words.

3.2.2 Dataset for heart disease prediction

This dataset contains a collection of medical data related to patients with and without heart disease. The dataset is designed for tasks such as classification, predictive modeling, and data analysis in healthcare.

3.2.3 Structure of the data:

The dataset is stored in a CSV file with the following columns, the following are the attributes information of the heart disease data set

age: Age of the patient.

sex: Sex of the patient (1 = male, 0 = female).

cp: Chest pain type (0: typical angina, 1: atypical angina, 2: non-anginal pain, 3: asymptomatic).

trestbps: Resting blood pressure (in mm Hg on admission to the hospital).

chol: Serum cholesterol in mg/dl.

fbs: Fasting blood sugar > 120 mg/dl (1 = true, 0 = false).

restecg: Resting electrocardiographic results (0: normal, 1: having ST-T wave abnormality, 2: showing probable or definite left ventricular hypertrophy).

thalach: Maximum heart rate achieved.

exang: Exercise-induced angina (1 = yes, 0 = no).

oldpeak: ST depression induced by exercise relative to rest.

slope: The slope of the peak exercise ST segment (0: upsloping, 1: flat, 2: downsloping).

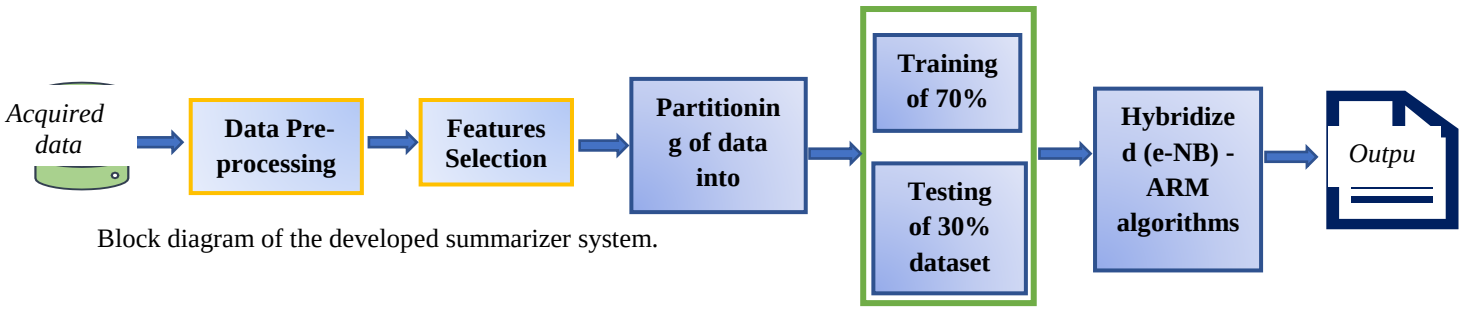
ca: Number of major vessels (0-3) colored by fluoroscopy.

thal: Thalassemia (1: normal, 2: fixed defect, 3: reversible defect).

target: Presence of heart disease (1 = presence, 0 = absence).

Table showing the sample of the data acquired from Kaggle online dataset for heart diseases

Age	sex	cp	Trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	0	0	1	1
57	0	0	120	354	0	1	163	1	0.6	0	0	1	1



Block diagram of the developed summarizer system.

3.3 Enhanced Naïve Bayes (e-NB) with Laplace smoothing

In order to improve the predictive power of Naïve Bayes for the classification task and captures missing values to make a robust summarization system, there is a requirement to carry out a Cross-validation and hyperparameter tuning with Laplace smoothing on the model because it is an essential technique for optimizing the performance of Naive Bayes. Together, they help improve the generalization of the model by ensuring that it performs well to avoid zero probabilities. Cross-validation helps in getting a reliable estimate of the model's performance on unseen data by training and validating the model on different subsets of data while the hyperparameter tuning involves optimizing the settings that control the behavior of the Naive Bayes model.

When used together, these techniques ensure that Naive Bayes model is both well-tuned and able to generalize to new data, resulting in better performance and reliability. One important hyperparameter in Multinomial Naive Bayes for text classification **is** the Laplace smoothing. Laplace smoothing (Hyperparameter **α**) helps to avoid zero probabilities when a particular feature X_i is not present in the training data for a given class C_k.

To calculate the smoothed likelihood estimate is:

$$P(X_i|C_k) = \frac{\text{count}(X_i, C_k) + \alpha}{\Sigma_{xi}\ \text{count}(X_i, C_k) + \alpha . N}$$

Where:

count (X_i, C_k) is the count of feature count X_i occurring in class count C_k

N is the total number of unique features
α is the smoothing parameter i.e. α = 1
The parameter α is tuned using hyperparameter tuning techniques of grid search for finding the best smoothing level that improves performance which determines what to include in the final model.

Algorithm 3.1: Algorithm to enhance Naive Bayes with Laplace smoothing parameter tuning

Input: A dataset of heart diseases consisting of N documents, where each document D_i contains text Ti.
Objective: Optimize the smoothing parameter α to handle zero probabilities in healthcare datasets for heart diseases.
Output: A concise summary generated from the input documents.
Step 1: Model training with Naive Bayes:
Step 2: Smoothing Parameter Tuning:
Step 3: Process of Cross-Validation on the training set to evaluate different values of α.
Step 4: Grid Search Test a range of α values (α = 0.5, 1.0, 1.5, 2.0) to identify the value that maximizes model performance.
Step 5: Model selection to choose the α that yields the highest accuracy or F1-score.
Step 6: Sentence Extraction for summarization:
Step 7: Prediction (summary-worthy or not)
Step 8: Select the extracted sentences classified as summary-worthy to construct the summary.
Step 9: Summary generation to combine the selected sentences from all documents to form a coherent summary.
Step 10: Remove redundant information to ensuring the summary is concise and diverse.
Step 11: Stop

3.4 Ranking the documents based on their combined scores

To rank documents based on their combined scores, it can sort the documents in descending order according to corresponding weighted scores. The documents with higher weighted scores will be ranked higher. Below is the process:
Calculate the combined weighted score $P_{weighted}(d)$ for each document d using the formula:

$$P_{weighted}(d) = \lambda \cdot P_{NB}(d) + (1 - \lambda) \cdot conf(Ad \rightarrow Bd)$$

Let:

$P_{weighted}(d)$ be the combined weighted score for the document d.
Where:

$P_{NB}(d)$ is the probability or relevance score assigned to document d by the Naïve Bayes classifier.
$conf(A_d \rightarrow B_d)$ is the confidence of the association rule $A_d \rightarrow B_d$ associated with document d.

1. **Ranking:**
   Sort the documents based on their combined scores $P_{weighted}(d)$ in the descending order. Documents with higher combined scores will be ranked higher.

Ranking the documents based on their combined scores:
Rank(d) = Descending order of $P_{weighted}(d)$

In summary, the documents are ranked in descending order of their combined weighted scores, which are calculated by combining the scores from the Naïve Bayes classifier and association rules using the specified parameter λ. This approach allows you to rank the documents based on their overall relevance considering information from both sources. Adjusting the parameter λ allows for fine-tuning the balance between the Naïve Bayes classifier and association rules.

3.4.1 Select the top-ranked documents for the final summary

To select the top-ranked documents for the final summary, we therefore specify a cut-off point in the ranked list of documents and select the documents ranked above this cut-off. This cut-off point can be determined based on a fixed number of top documents to select or based on a threshold score.
Here is the mathematical model
Let:
Rank(d) be the rank of document d based on its combined score.
Top_k be the top k documents to select for the final summary.

3.4.2 Selecting top documents:

Specify a cut-off point k based on the desired number of top documents to include in the summary or based on a threshold score.

If selecting a fixed number of top documents (Top_k):

$$Top_k = \{d_i \mid Rank(d_i) \leq k\}$$

If selecting based on a threshold score ($Score_{threshold}$):

$$Top_{threshold} = \{d_i \mid P_{weighted}(d_i) \geq Score_{threshold}\}$$

3.4.3 Final summary:

The final summary consists of the documents selected in the top k or based on the score threshold.

Model for selecting the top-ranked documents for the final summary.

$$Top_k = \{d_i \mid Rank(d_i) \leq k\}$$

In summary, the top-ranked documents is selected for the final summary by either specifying a fixed number of top documents to include or by setting a threshold score. This approach allows summarizing the most relevant documents

based on their combined scores, incorporating information from both the Naïve Bayes classifier and association rules.

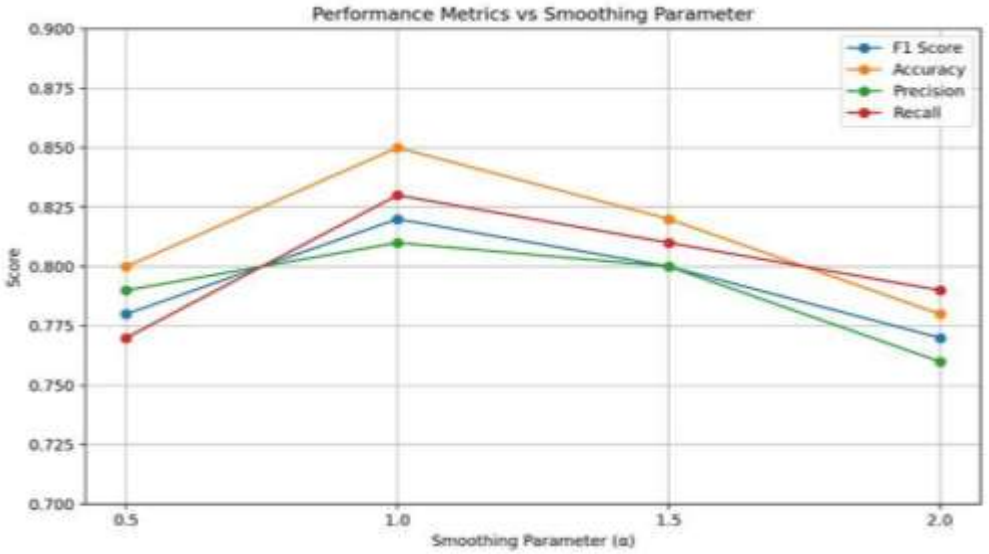
4.0 RESULTS AND DISCUSSION

4.1 Research Outcome

Evaluation of the enhanced Naïve Bayes classifier with Laplace smoothing

Table of Evaluation of enhanced Naïve Bayes (e-NB) with hyperparameter adjustment

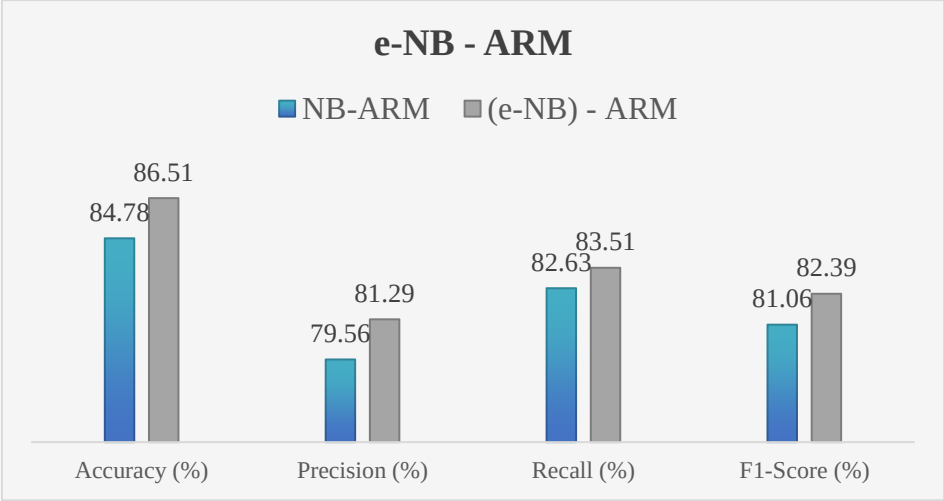
Smoothing parameter (α)	Cross-Validation F1 score	Accuracy	Precision	Recall
0.5	0.78	0.80	0.79	0.77
1.0	0.82	0.85	0.81	0.83
1.5	0.80	0.82	0.80	0.81
2.0	0.77	0.78	0.76	0.79



Graph showing the performance metrics versus smoothing parameter for enhanced Naïve Bayes

Table of performance evaluation of the (e-NB) and (ARM)

Metric	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (s)
NB-ARM	84.78	79.56	82.63	81.06	2.5
(e-NB) – ARM	86.51	81.29	83.51	82.39	2.0



Graph showing the performance evaluation of the (e-NB) and (ARM)

Analysis of results

The table above compares the performances of two models, i.e. the regular Naïve Bayes and Association rule mining (NB-ARM) with the enhanced Naïve Bayes and Association rule mining (e-NB) - ARM, across several key metrics: Accuracy, Precision, Recall, F1-Score, and Time complexity. The Accuracy for NB-ARM is 84.78%, while (e-NB) - ARM achieves 86.51%, indicating that (e-NB) - ARM is slightly more accurate. In terms of Precision, NB-ARM has 79.56% and (e-NB) - ARM has 81.29%, showing that (e-NB) - ARM has a better ability to avoid false positives.

The Recall for NB-ARM is 82.63%, and for (e-NB) - ARM it is 83.51%, meaning (e-NB) - ARM is slightly better at identifying true positives. The F1-Score, which balances Precision and Recall, is 81.06% for NB-ARM and 82.39% for (e-NB) - ARM, indicating that (e-NB) - ARM achieves a better balance between Precision and Recall. Finally, in terms of processing time, NB-ARM takes 2.5 seconds, whereas (e-NB) - ARM takes 2.0 seconds, making (e-NB) - ARM more time efficient. Overall, (e-NB) - ARM outperforms NB-ARM in all metrics, including accuracy, precision, recall, F1-score, and time efficiency. Therefore, (e-NB) - ARM is the preferable choice between the two models.

5.0 SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

It becomes inevitable to acquire and summarize vital related documents which contain information about this ailment to properly guide health practitioners and other scientists. In this research, a new model that depends on enhanced Naïve Bayes and Association Rule Mining was developed for the task of such multi documents summarization.

The Naive Bayes classifier used in this multi-document summarization task provides an initial set of relevance scores by analyzing word frequencies and other document features. These scores are then used as part of the broader summarization framework, where they are combined with the output from Association Rule Mining to generate a comprehensive and meaningful summary. This process highlights the Naive Bayes classifier's ability to quickly and efficiently identify relevant documents, while acknowledging its limitations in capturing deeper context and relationships. Therefore, the need for an enhanced version with Laplace smoothing with insightful patterns generated by Association Rule Mining allowing for more robust predictions, accurate and interpretable insights of heart disease as a result of the powerful model's outcomes.

5.2 CONCLUSION

This enhanced hybridized model improves predictive accuracy by combining the probabilistic approach of Naive Bayes with the insightful patterns generated by Association Rule Mining, allowing for more robust predictions, accurate and interpretable insights of heart disease. The dataset reveals that factors like age, cholesterol levels, chest pain, and blood pressure are critical in predicting heart disease, especially when they occur together in high-risk patterns.

5.2 RECOMMENDATION

Based on this study, it is recommended that adopting the enhanced hybrid Naive Bayes and Association Rule Mining (e-NB) -ARM model for multi-document summarization is the best, as it significantly improves performance metrics, especially in terms of precision, recall, The enhanced model is well-suited for applications that demand concise and contextually relevant summaries, particularly in fields where documents are varied and often sparse in language, such as healthcare, legal, and academic research.

In addition, it is recommended that incorporating hyperparameter tuning for optimal performance. Fine-tuning the Naive Bayes smoothing parameter (α) and adjusting the similarity thresholds within ARM for Regular cross-validation is recommended to maintain robust performance across various document collections.

REFERENCES

1. Abdelaleem, N., Elkader, H. A., Salem, R., Salama, D. D., and Elminaam, A. (2020, November). Extractive Text Summarization using Neural Network. In Proceedings of the 36th International Business Information Management Association (IBIMA). 13119-13131
2. Abid, A. M. (2022). Multi-Document Text Summarization Using Deep Belief Network. International Journal of Advances in Scientific Research and Engineering (IJASRE), 8(8), 56-65.
3. Alanzi, E., & Alballaa, S. (2023). Query-Focused Multi-document Summarization Survey. International Journal of Advanced Computer Science and Applications, 14(6)
4. Alias, S., Majalin, M., and Hayatin, N. (2023, August). A Visualized Hybrid Keyword-Cluster Approach for Extractive Text Summarizer Tool for STEM Education in Malaysia. In 2023 IEEE 8th International Conference on Software Engineering and Computer Systems (ICSECS) (pp. 139-144). IEEE.

5. Am, P. (2021, May). An Efficient Domain-Specific Text Summarization Using Combined Statistical and Linguistic Methods. In Proceedings of the International Conference on Smart Data Intelligence (Pp.154-163) (ICSMDI 2021).
6. Anand, D., and 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.
7. Anushka, R. L., Jagadish, S., Satyanarayana, V., & Singh, M. K. (2021, October). Lens less cameras for face detection and verification. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC) (pp. 242-246). IEEE.
8. Aote, S. S., Pimpalshende, A., Potnurwar, A., and Lohi, S. (2023). Binary Particle Swarm Optimization with an improved genetic algorithm to solve multi-document text summarization problem of Hindi documents. Engineering Applications of Artificial Intelligence, 117, 105575.
9. Bao, C., Liu, X., Zhang, H., Li, Y., & Liu, J. (2020). Coronavirus disease 2019 (COVID-19) CT findings: a systematic review and meta-analysis. Journal of the American college of radiology, 17(6), 701-709.
10. Belwal, R. C., Rai, S., and Gupta, A. (2021). A new graph-based extractive text summarization using keywords or topic modeling. Journal of Ambient Intelligence and Humanized Computing, 12(10), 8975-8990.
11. Bhat, P., Anuse, A., Kute, R., Bhadade, R. S., and Purnaye, P. (2022). Mental health analyzer for depression detection based on textual analysis. Journal of Advances in Information Technology Vol, 13(1). 67-77
12. Bidoki, M., Moosavi, M. R., and Fakhrahmad, M. (2020). A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities. Information Processing and Management, 57(6), 102-217.
13. Curiel, A., Gutiérrez-Soto, C., Soto-Borquez, P. N., and Galdames, P. (2020, November). Measuring the Effects of Summarization in Cluster-based Information Retrieval. In 2020 39th International Conference of the Chilean Computer Science Society (SCCC) (pp. 1-8). IEEE.
14. D'Silva, J., and Sharma, U. (2023). Automatic text summarization of Konkani Folk tales using supervised machine learning algorithms and language independent features. IETE Journal of Research, 69(9), 6162-6175.
15. Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N. A., & Choi, Y. (2021). Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. arXiv preprint arXiv:2107.01294.
16. Du, C., Li, Y., Qiu, Z., & Xu, C. (2024). Stable diffusion is unstable. Advances in Neural Information Processing Systems, 36.
17. El-Kassas**,** W. S., Salama, C. R., Rafea, A. A., and Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. Expert systems with applications, 165, 113679.
18. Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for NLP. arXiv preprint arXiv:2105.03075.
19. Guan, W., Smetannikov, I., & Tianxing, M. (2020, October). Survey on automatic text summarization and transformer models applicability. In Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System (pp. 176-184).
20. Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. Sustainable Operations and Computers, 3, 238-248.
21. Inui, K., Jiang, J., Ng, V., & Wan, X. (2019, November). Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).
22. Kumar, Y., Kaur, K., and Kaur, S. (2021). Study of automatic text summarization approaches in different languages. Artificial Intelligence Review, 54(8), 5897-5929
23. Lewis, D. (2020). Mounting evidence suggests coronavirus is airborne—but health advice has not caught up. Nature, 583(7817), 510-513.
24. Liu, Y., Titov, I., and Lapata, M. (2019). Single document summarization as tree induction. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 1745-1755).
25. Ma, C. (2024). Deep Learning Based Multi-document Summarization (Doctoral dissertation).
26. Ma, Y., Xie, Z., Li, G., Ma, K., Huang, Z., Qiu, Q., & Liu, H. (2022). Text visualization for geological hazard documents via text mining and natural language processing. Earth Science Informatics, 1-16.

27. Ma, C., Zhang, W. E., Guo, M., Wang, H., and Sheng, Q. Z. (2022). Multi-document summarization via deep learning techniques: A survey. ACM Computing Surveys, 55(5), 1-37.
28. Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., and Kabir, M. M. (2021). A survey of automatic text summarization: Progress, process and challenges. IEEE Access, 9, 156043-156070.
29. Myla, S. D., Saini, E. R., and Kapoor, E. N. (2024, January). Auto Text Summarization in Natural Language Processing. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 1258-1267). IEEE.
30. Nguyen, T., Luu, A. T., Lu, T., and Quan, T. (2021). Enriching and controlling global semantics for text summarization. arXiv preprint arXiv:2109.10616.
31. Ni, A., Azerbayev, Z., Mutuma, M., Feng, T., Zhang, Y., Yu, T., ... and Radev, D. (2021). SummerTime: Text summarization toolkit for non-experts. arXiv preprint arXiv:2108.12738.
32. Pagnoni, A., Balachandran, V., and Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. arXiv preprint arXiv:2104.13346.
33. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140), 1-67.
34. Raza, H., & Shahzad, W. (2024). End to End Urdu Abstractive Text Summarization With Dataset and Improvement in Evaluation Metric. IEEE Access.
35. Razumovskaia, E., Glavaš, G., Majewska, O., Korhonen, A., and Vulic, I. (2021). Crossing the conversational chasm: A primer on multilingual task-oriented dialogue systems. arXiv preprint arXiv:2104.08570.
36. Roy, P., and Kundu, S. (2023). Review on Query-focused Multi-document Summarization (QMDS) with Comparative Analysis. ACM Computing Surveys, 56(1), 1-38.
37. Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2022). A multi-objective memetic algorithm for query-oriented text summarization: Medicine texts as a case study. Expert Systems with Applications, 198, 116769.
38. See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.
39. Song, F., Shi, N., Shan, F., Zhang, Z., Shen, J., Lu, H., ... & Shi, Y. (2020). Emerging 2019 novel coronavirus (2019-nCoV) pneumonia. Radiology, 295(1), 210-217.
40. Waly, R. R., and Gomaa, W. H. (2022, May). Extractive Summarization of Scientific Articles. In 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 349-354). IEEE.
41. Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., and Affandy, A. (2022). Review of automatic text summarization techniques and methods. Journal of King Saud University-Computer and Information Sciences, 34(4), 1029-1046.
42. Yadav, A. K., Ranvijay, Yadav, R. S., & Maurya, A. K. (2024). Graph-based extractive text summarization based on single document. Multimedia Tools and Applications, 83(7), 18987-19013.
43. Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., and Hashimoto, T. B. (2024). Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics, 12, 39-57.
44. Zhang, S., and Bansal, M. (2021). Finding a balanced degree of automation for summary evaluation. arXiv preprint arXiv:2109.11503.
45. Zhang, J., Lu, H., Zeng, H., Zhang, S., Du, Q., Jiang, T., & Du, B. (2020). The differential psychological distress of populations affected by the COVID-19 pandemic. Brain, behavior, and immunity, 87, 49.
46. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 52(1), 1-38.