WILEY | Hindawi

Research Article

News Classification and Categorization with Smart Function Sentiment Analysis

Mike Nkongolo Wa Nkongolo [ID]

Department of Informatics, University of Pretoria, Gauteng, South Africa

Correspondence should be addressed to Mike Nkongolo Wa Nkongolo; mike.wankongolo@up.ac.za

Search engines are tools used to find information on the Internet. Since the web has a plethora of websites, the engine queries the majority of active sites and builds a database organized according to keywords utilized in the search. Because of this, when a user types a few descriptive words on the home page of the search engine, the search function lists websites corresponding to these keywords. However, there are some problems with this search approach. For instance, if a user wants information about the word Jaguar, most search results are animals and cars. This is a polysemic problem that forces search engines to always provide the most popular but not the most relevant results. This article presents a study of using sentiment technology to help news classification and categorization and improve the classification accuracy. We have introduced a smart search function embedded into a search engine to tackle polysemic issues and record relevant results to determine their sentimentality. Therefore, this study presents a topic that involves several aspects of natural language processing (NLP) and sentiment analysis for news categorization and classification. A web crawler was used to collect British Broadcasting Corporation (BBC) news across the Internet, carried out preprocessing of text by using NLP, and applied sentiment analysis methods to determine the polarity of the processed text data. The sentimentality represents negative, positive, or neutral polarities assigned by the sentiment analysis algorithms. The research utilized the BBC news site to collect different information using a web crawler and a database to explore the sentimentality of BBC news. The natural language toolkit (NLTK) and BM25 indexed and preprocessed patterns in the database. The experimental results depict the proposed search function surpassing normal search with an accuracy rate of 85%. Moreover, the results show a negative polarity of BBC news using the Sentistrength algorithm. Furthermore, the Valence Aware Dictionary and sEntiment Reasoner (VADER) was the best-performing sentiment analysis model for news classification. This model obtained an accuracy of 85% using data collected with the proposed smart function.

1. Introduction

Over the past few years, there have been significant advancements in natural language processing (NLP) [1]. These advancements have been driven by improvements in linguistic models that can predict words, characters, and sentences from textual data [1, 2]. Chatbot models are considered the most efficient NLP programs, demonstrating accurate performance across various existing datasets for different NLP problems, including question answering, translation, news article generation, sentiment analysis (opinion mining), and unscrambling words. This has been highlighted in literature by several researchers [3–5].

Nevertheless, the current evaluation and training approach for NLP favors only algorithms that have access to large datasets. In addition, the polysemy of textual patterns can negatively impact the performance of deployed NLP models.

To achieve human-like language capability, an NLP program must employ complex and disruptive technologies, while also addressing the need for feature engineering [1, 2]. The study aimed to address the challenges in generating accurate news using NLP by bridging the gap between NLP expectations and the difficulties encountered. The main methodology used in NLP is the engineering of textual patterns. However, the study recognized the shortcomings of implementing standard NLP to textual data without feature

engineering. To address this, the study proposed a methodology that incorporates feature engineering by utilizing NLP models like embeddings and tokenizers to extract relevant features from existing data.

The proposed methodology aims to improve computational efficiency by reducing empirical errors and increasing accuracy levels. This article advocates for the adoption of an NLP system that focuses on extracting relevant textual data from the web for sentiment analysis. Although NLP has achieved notable progress, it is not yet equipped to address real-world problems with accuracy [3, 6], since its success is dependent on big data, evaluation metrics, and training approaches that favor probabilistic and heuristics learning. There are currently numerous well-known news websites available [7, 8], and the majority of them include an integrated search feature. Two popular news sites, China (https://www.chinanews.com) and BBC (https://www.bbc.co.uk), have built-in search functions, but these are limited and do not include advanced options like sentiment analysis. This study aims to design and implement a smart search engine with sentiment analysis capabilities to determine the opinion of search results and categorize them as negative, positive, or neutral. Such a search engine can automatically extract brand visibility or reputation from the Internet in real time by scoring search mentions positively or negatively.

Thus, the implementation of a smart search engine that incorporates sentiment analysis can provide a real-time understanding of people's attitudes towards specific brands during their Internet news search. The sentiment analysis results can provide relevant insights to improve a brand's market share, competitive advantage, and reputation. Additionally, search engines with smart functions and sentiment analysis can influence consumers to purchase products with a positive reputation.

In the context of news classification and categorization, a smart function based on sentiment analysis can foster brand trust and elevate a brand's reputation if the sentiment analysis results are positive. Text categorization and classification share many similarities, with the latter being a subset of the former [9–11]. In text categorization, the first step is to represent the text by preprocessing the documents and creating a vector space containing the words present in the documents using the bag-of-words (BoW) model.

Consequently, the classification of text documents relies on the proximity of keyword vectors, with the significance of keywords in the documents often determined by weighting schemes such as term frequency or word frequency. In contrast, sentiment analysis involves identifying relevant keywords in a textual document using linguistic patterns [11].

The sentiment classification field has adopted the n-gram technique, which involves dividing sentences into tokens and using the sequence of tokens for text representation. Additionally, the part of speech (POS) technique is employed to tag words with their grammatical attributes, such as nouns, adverbs, verbs, or adjectives. These types of representations are commonly used in sentiment classification research [9, 11, 12]. This study employed a smart function to obtain unambiguous search results from the BBC search engine and then conducted sentiment analysis to assess the reputation of BBC news.

Users typically use keywords to search for news and expect the most relevant results. However, search engines often prioritize results based on their popularity rather than relevance, leading to a multiconnotation issue where desired results cannot be obtained due to the multiple meanings of the keywords. Search engines such as Google, Baidu, and Yandex prioritize recent or credible results, but these may not always be relevant to the user's needs. For example, searching for information about "apples" may yield results about both fruits and phones. The issue described is related to the problem of polysemy [8], which refers to multiple meanings of a word.

This challenge can be addressed by enabling users to provide more specific information to refine their search. However, only a few studies have addressed this issue. Some studies have utilized techniques such as anomaly detection and neural networks to classify search results that were manually collected [13].

This study aimed to optimize the categorization and classification of search results by implementing a search function on the search engine, resulting in fast and automated data collection. The proposed method also integrated sentiment analysis to determine the sentimentality of BBC news. The goal was to present a computational pipeline capable of quickly finding the desired data on the Internet to classify patterns and analyze an entity's web presence. To achieve this, a search function was employed to extract BBC news and classify it into three categories (COVID-19, vaccine, and travel), and sentiment analysis algorithms were used to detect the news polarity as either negative, neutral, or positive. By using sentiment analysis as a proxy, this research utilized a classified search engine to understand the polarity assigned to a web entity. A database with various categories of news was created, but the search function was not able to search the entire Internet.

However, it worked together with the search engine to automatically gather data. Each news item was tagged in the database to make sentiment analysis computation easier on the labeled data. The tags were fixed and divided into different categories. This article proposes a framework for processing textual data that brings engineering practices and paradigms to NLP in order to extract sentiment from web news. The proposed search function is envisioned as an optimized algorithmic pipeline that provides the most relevant results from the search engine. This article discusses how data quality and quantity can be addressed in engineering textual patterns and emphasizes the importance of including unstructured feature engineering, which is currently not widely available due to limited packages, in NLP problems [14].

1.1. Research Aim. The main goal of this study was to create a search function that could gather BBC news from the Internet, while also minimizing polysemous results. This search function was integrated into the BBC search engine, and the collected data were stored in a database. Sentiment analysis algorithms were then applied to this database to determine the polarity of the BBC news. The performance of

the sentiment analysis was evaluated using metrics such as precision, accuracy, F1 score, and recall. The following research questions were the focus of this study:

(i) What effect do smart functions have on a search engine for news categorization?

(ii) Which sentiment analysis model is the most suitable to classify news?

(iii) How does NLP preprocess news data?

The implementation of the smart function not only aims to improve the user's search experience but also facilitates the automated collection and sentiment analysis of relevant BBC news data. By reducing the polysemous issues, the function enables quick and efficient collection of data, which can be further categorized through automated sentiment analysis. The results of the study demonstrate that VADER was the most accurate and precise sentiment analysis model suitable for integration into a search engine for automated sentiment analysis.

Moreover, this research highlights the importance of NLP techniques in the processing of unstructured data. By engineering unstructured patterns and standardizing file formats in the database, data preparation and analysis become more manageable. This NLP-based approach enables the utilization of native formats of unstructured patterns, which can be processed and analyzed efficiently to provide insights into the sentimentality of BBC news data. Therefore, this study provides a comprehensive framework for the collection, processing, and sentiment analysis of unstructured data, which can be adapted and utilized for other similar applications.

1.2. Research Rationale and Contribution. Polysemy is a phenomenon in language where a word can have multiple meanings or interpretations [15]. This can lead to ambiguity and confusion in NLP tasks like search queries, where the context of the query may not always be clear. Sentiment analysis of search results can help with this problem by providing additional context and clues to disambiguate the meaning of a query. Sentiment analysis involves analyzing the emotional tone or sentiment of a piece of text by using NLP techniques to classify it as positive, negative, or neutral [16]. When applied to search results, sentiment analysis can provide valuable information about the context and connotations of the words in the results. For example, consider a search query for the word *bank*.

Depending on the context, this could refer to a financial institution or the side of a river. Sentiment analysis of the search results could help determine which meaning is most likely based on the emotional tone of the text. If the majority of the results are associated with positive financial news or reviews, it is more likely that the user was searching for a financial institution. Conversely, if the results are associated with negative river pollution or natural disaster news, it is more likely that the user was searching for the side of a river. In summary, sentiment analysis can help with the problem of polysemy by providing additional context and emotional clues to disambiguate the meaning of a search query. By analyzing the sentiment of search results, it is

possible to infer the user's intended meaning and provide more accurate and relevant results.

The aim of this research paper is threefold. First, it aims to present a solution for the polysemous problems encountered in search engines. Second, it seeks to identify the requirements that optimized search functions must meet. Third, it argues for the use of optimized search functions to retrieve relevant data and conduct sentiment analysis to evaluate the strength of an entity's web presence. The paper introduces a search function that can be utilized for search engine optimization in practical applications. The research contributes to the fields of opinion mining and search engine optimization by presenting a relevant theory that can be easily implemented and tested for news categorization and classification. The primary objective is to present a computational methodology that analyzes the web presence of an entity to study its sentimentality using sentiment analysis as a proxy. The paper introduces such a framework and evaluates its performance on BBC news for news categorization and classification. The design of the study is depicted in Figure 1. The article is organized into several sections, including the background (Section 2), research methodology (Section 3), and results (Section 4). The conclusion and recommendations are presented in Section 5.

2. Background

Although NLP has made significant advancements in achieving human-level performance in various tasks, researchers who seek to apply NLP to real-world problems still encounter challenges that demand more data annotation and training. To address this issue, an iterative and human-centered framework for NLP that incorporates feature engineering has been proposed.

This study recommends involving human participation in each stage of the data analysis process by focusing on the engineering of textual data [12] to enable the mitigation of technical issues in the proposed computational framework. The proposed framework, which adopts an NLP approach that focuses on the challenges of engineering textual data, holds promise for analyzing BBC news to determine their sentimentality. With this approach, it is expected that systems can be implemented to capture or understand the meaning of the analyzed text, thus mitigating technical issues in the proposed computational framework [17].

One major obstacle in developing robust NLP systems is the "symbolic problem," in which symbols are interpreted based on their morphology rather than their intended meaning. This complication presents a significant challenge in assessing the capacity of computational machines to comprehend word meanings [17, 18].

NLP systems assume that the meaning of a sentence can be determined by analyzing a set of words in it. However, in technical and textual contexts, this assumption may not hold true since generalization is not always closely associated with sentence meaning. Furthermore, NLP systems do not learn effectively from training features and require significant retraining to improve the model's performance, as reported in recent studies [18–20].
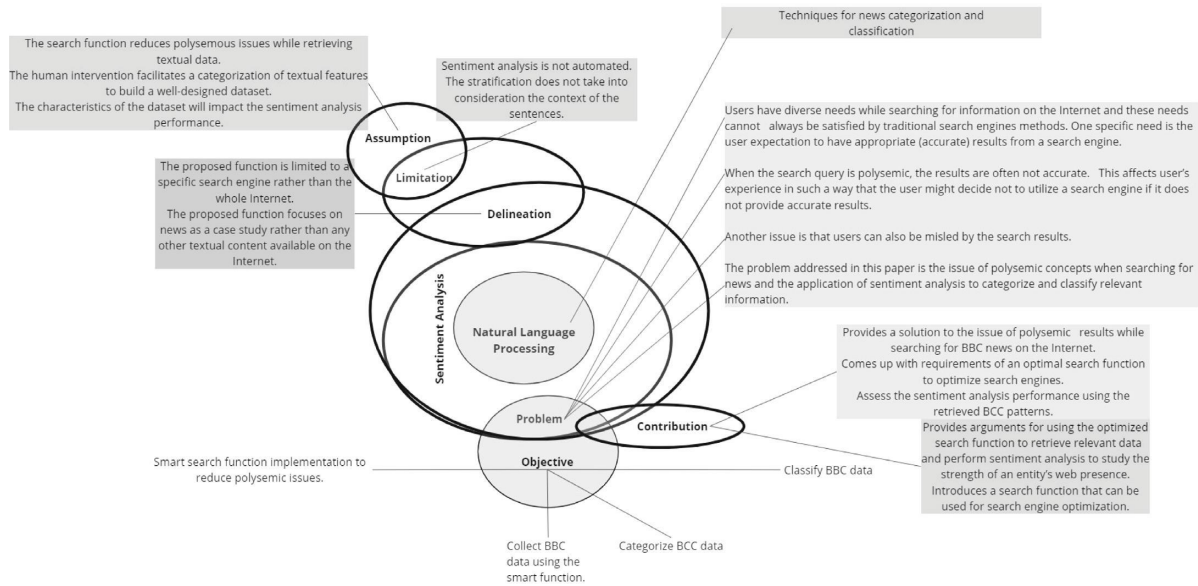
Figure 1: The study design.

The semantic disadvantage can be attributed to the evaluation metrics and current training methods that do not promote the implementation of human-like generalization models. As a result, NLP systems learn heuristically rather than learning the expected generalizations [19]. To address this issue, the proposed engineering approach involves human supervision and intervention, along with computational techniques such as annotation, tagging, and textual structuring, to facilitate sentiment analysis computation. Additionally, machine learning algorithms can be implemented as a tagging and forecasting approach to mitigate spurious correlations and heuristics that affect NLP learning.

In a study conducted by Bi [11], a novel approach was introduced for sentiment analysis that utilized a belief functions framework to merge sentiment analysis outputs. The study evaluated the effectiveness of sentiment stratification individually and in combination. The results showed that the combined algorithms performed better than single classifiers across five review datasets. Bi [11] developed a dichotomous ensemble learning method for sentiment analysis by using a triplet evidential scheme [9] to formulate negative and positive polarities, and negative and positive propositions for neutral polarity representation. This approach reduced the negative effects of neutral sentiments through ensemble learning and classified patterns with evidential reasoning schemes.

The proposed approach achieved a reduction in both the training time's complexity and cost, even when the neutral polarity was not considered. The method represented text reviews using a bag-of-words (BoW) approach and performed tenfold cross-validation. The proposed methodology was evaluated using the F score metric, and a total of 10,000 algorithms of triplet belief functions were utilized through three combination techniques with two groups of nine and eight learning classifiers. The results showed that the proposed method achieved an accuracy of 87%–90%, outperforming existing techniques [11]. Jalil et al. [21] employed

both deep learning and machine learning techniques to perform sentiment analysis on data related to the corona virus disease 2019 (COVID-19). Their experiments yielded an accuracy of 93% to 95%, which was higher than that of other techniques used in the study. In another study, Reddy et al. [22] applied an NLP algorithm to segment a collection of medical publications. The methodology involved segmenting all the words and the associated information.

Yogesh Pawade [23] demonstrated the benefits of a search engine optimization approach by using Google to extract essential information such as users' location and time spent on each website. Although the study did not involve pattern matching, it highlighted the potential of NLP algorithms in enhancing search engine optimization.

In the proposed methodology, user behavior analysis can be utilized to enhance the search engine experience by predicting user intent when searching for polysemous keywords. For example, if a user frequently searches for animal-related information and inputs the keyword Jaguar, the search engine can predict that the user is likely searching for animal-related data, rather than information about the car brand. By analyzing user behavior and preferences, the search engine can improve its ability to provide relevant and accurate search results. This approach is similar to previous studies that have used user behavior analysis to enhance search engine optimization and improve the quality of user experience [24].

Bi [10] conducted an experiment to evaluate the relationship between accuracy and diversity of classifiers using pairwise and nonpairwise diversity measures and evidential combination rules. The study utilized Yager's and Proportion's rules to generate ensemble learning negative classifiers, which did not prioritize minimizing the error of selected classifiers. Empirical results showed that increasing diversity decreased the accuracy of ensemble learning. However, the study did not investigate the behaviors of member classifiers concerning the efficiency of the

ensembling scheme. Huang et al. [25] introduced a novel approach, called the inverted index method, for efficient execution of temporal queries, which was demonstrated on a COVID-19 dataset [25].

The proposed method utilized a reverse sort-index approach, enabling real-time query processing to facilitate COVID-19 research.

The authors developed several categories of queries, including nontemporal, relative temporal, and absolute temporal queries. The results of the experiment suggest that the reverse sort indexing method is more efficient than current techniques in facilitating fast-time query execution for search engines. The inverted index is capable of rapidly retrieving nonpolysemous big data. The proposed methodology is limited to searching and storing BBC news search results in the database. In the future, this approach can be applied to other search engines to extract more data. Nevertheless, sorting indexes, as highlighted by Huang et al. [25], remain an essential aspect of search engine optimization.

Xu et al. [20] devised a text-based approach for aircraft fault diagnosis by employing Word2vec and convolutional neural networks (CNNs). The experiments utilized a large corpus of text files, and Word2vec was employed to retrieve textual pattern vectors which were then passed to the CNNs for final decision-making. A cloud similarity measure (CSM) was used by the CNN model to detect faulty knowledge, resulting in improved classifier performance and supporting aircraft maintenance. By combining structured and unstructured patterns for fault diagnosis, Xu et al. [20] were able to identify the underlying cause of the aircraft fault.

The proposed framework utilized Word2vec to extract the contents of search results and suggest results based on the similarity of their vectors. This approach allows users to view results relevant to the keywords they have entered. Similar techniques have been employed by many search engines but with various optimization search functions.

Reddy et al. [22] utilized NLP to extract nonpolysemous information from a vast medical corpus, while with the proposed methodology, NLP was used to select relevant information from BBC news. Yogesh Pawade [23] highlighted the importance of search engine optimization using an inverted index on a large dataset. The proposed methodology also employs the inverted index technique but with fewer entries than Yogesh Pawade [23]. However, the proposed technique supports the idea that the search function can be scaled to search big data efficiently.

The study contends that a large dataset may contain spurious correlations as a result of its size. These correlations can overpower the identification of pertinent information that cannot be differentiated algorithmically. The more extensive the features examined, the higher the likelihood that spurious correlations will result in inaccurate conclusions and dominate the ultimate outcomes.

Dilrukshi and De Zoysa [26] developed a news classification and categorization system based on stratifying news by relevance. The study utilized a Twitter dataset and focused on articles related to Sri Lanka, utilizing dimension-reduction techniques. Frequent words were found to be less informative for text categorization, so irrelevant words were removed as noisy data. The models were evaluated using precision and recall values for each category, and the evaluation was independently conducted for each category.

Bun and Ishizuka [27] conducted a study on news articles to group them based on their topics. They utilized the term frequency proportional document frequency (TF-PDF) approach, which calculated the relevance of a word in a specific text. The study revealed that the significance of a word proportionally increases based on its frequency of occurrence in the text.

Kapusta and Obonya [28] conducted research on identifying fake news and categorizing news articles as either authentic or fake. Their objective was to develop a feature set from floating languages, such as Slovak, and apply it to detect fake news. They created a dataset using news articles from various publishers and labeled the features after scrutinizing the authenticity of the information. The study emphasized morphological learning approaches over contextual learning approaches. They introduced a technique to classify floating languages using part-of-speech (POS) tagging, which yielded reasonable accuracy.

Daud et al. [29] utilized hyperparameter-optimized support vector machines (HOSVMs) to categorize online news articles into different categories, highlighting the cost-effectiveness and simplicity of this approach. Five other adaptive computation methods were also optimized and compared for news categorization. The results showed that the HOSVM performed better than the other models, while the nonoptimized scheme performed worse than the alternative methods.

Yousef and Voskergian [30] introduced TextNetTopics, a feature selection model that employs bag-of-topics (BoT) instead of the bag-of-words (BoW) technique. The BoT method selects topics instead of individual words. The model utilized neural network layers to capture and analyze relevant information from sentences. The experiments demonstrated that TextNetTopics outperformed other feature selection models and achieved better results on various textual datasets.

Bi et al. [9] proposed a class-indifferent approach for merging classifier outputs using evidential structures (triplet and quartet) with Dempster's rule of combination. This ensemble methodology aimed to distinguish relevant observations from irrelevant ones by representing classifier results and providing pragmatic ways to apply the Dempster–Shafer theory of evidence to the ensemble learning scheme. To combine the mass functions, a formalism modeling classifier outputs as triplet mass functions were designed to provide decision support. In addition, a comparative analysis was conducted with dichotomous structures to compare the proposed method with majority voting and Dempster's rule.

The experiment was conducted using the UCI dataset, which demonstrated the advantages of the proposed approach. Table 1 presents a literature-based comparative analysis to validate the superior performance of the proposed model. According to Table 1, recent studies have not extensively explored news classification/categorization based on sentiment analysis. While many studies have utilized benchmarked datasets to discover news categories,

TABLE 1: Literature-based comparative analysis.

Author	Method	Data	Recall	Precision	Limitation
Jalil et al. [21]	Sentiment analysis	COVID-19	93%	95%	False negative
Reddy et al. [22]	NLP	Medicine	90%	85%	Symbolic morphology
Yogesh Pawade [23]	Search	Internet	76%	63%	Polysemy
Huang et al. [25]	Inverted index	COVID-19	64%	70%	Weighting
Dilrukshi and De Zoysa [26]	Naive Bayes	Twitter	90%	62%	Ambiguous
Bun and Ishizuka [27]	TF-PDF	News archive	—	—	Stop words
Kapusta and Obonya [28]	Decision tree	Slovak text	68%	75%	Noise
Zhu et al. [29]	Sentiment analysis	Chinese text	90%	68%	Feature engineering
Yousef and Voskergian [30]	TextNetTopics	Textual data	80%	71%	Misclassification
This work	Search function-based sentiment analysis	BBC news	94%	97%	Embedded category
Bi [11]	Triplet belief functions	Review	—	—	Not self-adaptable

these datasets may not be applicable in real-world scenarios due to the presence of unstructured data. Furthermore, while much attention has been given to classifying news as authentic or fake, very few studies have focused on news classification and categorization using sentiment analysis. The proposed function, as demonstrated in Table 1, outperforms most of the papers discussed in this section. In summary, the literature review highlights these key findings.

There is no single sentiment analysis method that performs better than others across different datasets. Additionally, there are only a few studies that explicitly apply established fusion techniques to combine classifier results for sentiment classification. Therefore, more extensive experimental work is needed to apply evidential reasoning approaches to the combination of classifiers for sentiment classification [9, 10].

To determine sentiment classification, it is necessary to address the uncertainty that arises due to the ambiguity between positive, negative, and neutral categories. Moreover, there is a current lack of effective methods to deal with this uncertainty, particularly in the context of sentiment classification.

3. Research Methodology

A search function was developed to retrieve BBC news information based on specific keywords (Figure 2), aiming to address the problem of polysemy and enhance search function accuracy. The web mining method was used to collect data by extracting content from web documents. A web crawler was employed to crawl the BBC news website and retrieve the required data [13, 31]. The collected data were subjected to NLTK preprocessing and inverted index phases before being recorded in the database. To improve search accuracy, a search function based on a search engine was developed to crawl the BBC news website for specific keywords. When these keywords matched the database features, corresponding results were produced. Otherwise, the most similar results were provided. The NLTK was chosen to implement the search function by tokenizing sentences into words. The inverted index helped in collecting relevant information, while Word2vector computed news similarity. The BM25, a probabilistic retrieval model developed by Stephen E. Robertson in the 1970s and 1980s, was

used in the optimization scheme [32]. The BM25 method is utilized to tokenize the user's keywords into distinct words and then apply a ranking function to arrange matching information based on their significance. This method utilizes a probabilistic retrieval approach to match patterns with their corresponding indexed information [32].

To determine the similarity of each document, a score is generally calculated. The primary purpose of BM25 is to rank web documents based on specific queries, and it can also be seen as a measure of relevant information in some cases [33]. In the preprocessing phase, BM25 was implemented as a ranking method for the search engine to estimate a document's relevance for a given query. The BM25 has been found to be more adaptable compared to the traditional term frequency-inverse document frequency (TF-IDF) method [23, 24]. This adaptability of BM25 makes it more flexible. The mathematical expression for BM25 is presented as follows:

$$\text{BM25} = \frac{\text{Number of documents containing } k_i}{\text{Total number of all documents}}. \quad (1)$$

The feature's weight $w_{i,j}$ is computed as follows:

$$w_{i,j} = \frac{f_{i,j}(k_{i,j})}{f_{i,j}}. \quad (2)$$

For $k_{i,j} > 0$ and $f_{i,j} > k_{i,j}$ where the weight of documents containing keywords k_i is represented by $w_{i,j}$, the total number of all documents is represented by $f_{i,j}$. Here, i is the number of characters and j is the index of a character. Moreover, the normalization of term frequency by document length is not accurately lower-bounded with the BM25 technique. The current limitation of this approach is that long documents that do not match the user query may receive an unfair score and be deemed similar to shorter documents that do not include the query.

3.1. The Classification Complexity of the BM25.
The classification complexity of the BM25 technique can be demonstrated by analyzing its time complexity in terms of the number of documents and the length of the documents. Let N be the number of documents, n be the average length of the documents, and m be the number of keywords. The BM25 technique involves the following steps:
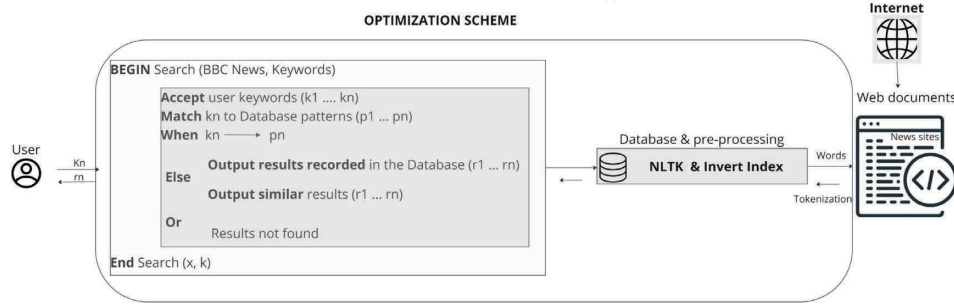
FIGURE 2: The search function optimization scheme.

(i) Preprocessing: the documents are preprocessed to tokenize and stem the words, remove stop words, and calculate the term frequency

(ii) Query Processing: the query is also preprocessed, and the query terms are ranked based on their relevance to the query

(iii) Scoring: the documents are scored based on their relevance to the query, using the following formula:

$$\text{score}(D, Q) = \sum_{i=1}^{m} idf(k_i) \times \frac{f_{i,j} \times (k_{i,j} + 1)}{f_{i,j} + k_{i,j} \times (1 - b + b \times (|D|/\text{avgdl}))}, \tag{3}$$

where D is a document, Q is a query, k_i is a keyword, $f_{i,j}$ is the frequency of k_i in document j, $k_{i,j}$ is a parameter that determines the saturation point of the score function for k_i in document j, b is a parameter that determines the importance of the document length, and avgdl is the average document length.

(i) Ranking: the documents are ranked based on their scores, and the top-k documents are selected as the search results

The time complexity of each step can be analyzed as follows:

(i) Preprocessing: the preprocessing step has a time complexity of $O.\text{Nn}$, as each document needs to be processed individually

(ii) Query Processing: the query processing step has a time complexity of $O.m \log N$, as the query terms need to be ranked based on their relevance to the query

(iii) Scoring: the scoring step has a time complexity of $O.\text{Nm}$, as each document needs to be scored individually for each keyword

(iv) Ranking: the ranking step has a time complexity of $O.N \log N$, as the documents need to be sorted based on their scores

Therefore, the overall time complexity of the BM25 technique can be expressed as follows:

$$O(\text{Nn} + m \log N + \text{Nm} + N \log N). \tag{4}$$

In practice, the value of m can be much larger than the value of n or N, which means that the time complexity of the scoring step dominates the overall time complexity of the algorithm. As

a result, the BM25 technique can be computationally expensive for large datasets and queries with many keywords.

3.2. The Experimental Data. The study selected BBC data for crawling because it is the world's largest news broadcaster [16], which lends credibility and authority to the news examined in this study [16, 34]. The BBC news website encompasses a wide range of topics that are appropriate for the proposed search function. The study randomly selected a sample of 800 articles from the website to ensure the reliability of the search results. The data were stored in a database table called news (refer to Figure 3). The study retrieved several attributes from multiple BBC articles, including the uniform resource locator (URL), content, date, title, and label, which displays the categorized information of BBC news (Figure 3). Afterward, sentiment analysis computation was performed on the database table's content attribute to determine the polarity of textual data. This method proved to be useful in predicting the sentimentality strength of BBC news.

Figure 4 displays the stop word type that was detected and ignored during the feature engineering process. The frequency of stop word occurrence in the BBC dataset is shown in Figure 5. Removing stop words during feature engineering reduced both the size of the original dataset and the time required for sentiment analysis prediction [35, 36]. The elimination of stop words enhanced the prediction accuracy by retaining only relevant tokens or patterns in the BBC dataset. This could improve the values of evaluation metrics. The dataset contains three categorical variables or columns (COVID-19, vaccine, and travel). The extracted features' descriptions (desc) using the proposed search function are presented in Figure 6.
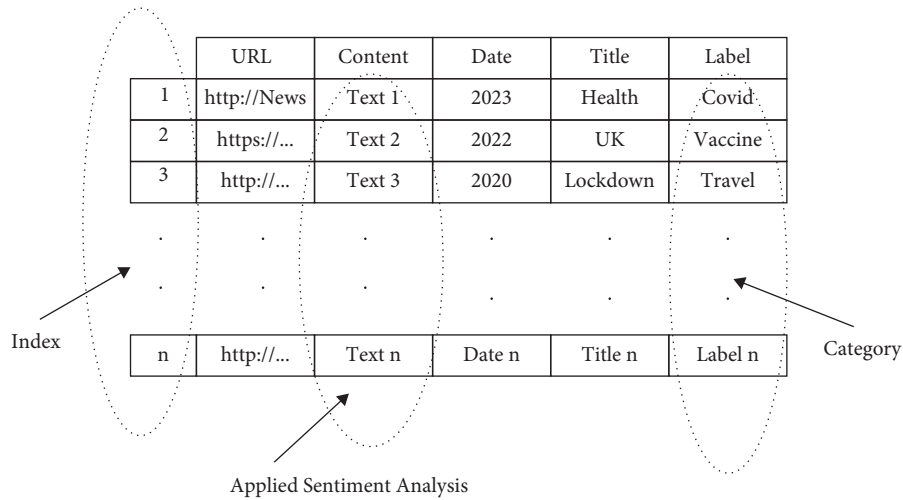
	URL	Content	Date	Title	Label
1	http://News	Text 1	2023	Health	Covid
2	https://...	Text 2	2022	UK	Vaccine
3	http://...	Text 3	2020	Lockdown	Travel
.	.	.	.	.	.
.	.	.	.	.	.
n	http://...	Text n	Date n	Title n	Label n

Index

Category

Applied Sentiment Analysis

FIGURE 3: The news table of the database.

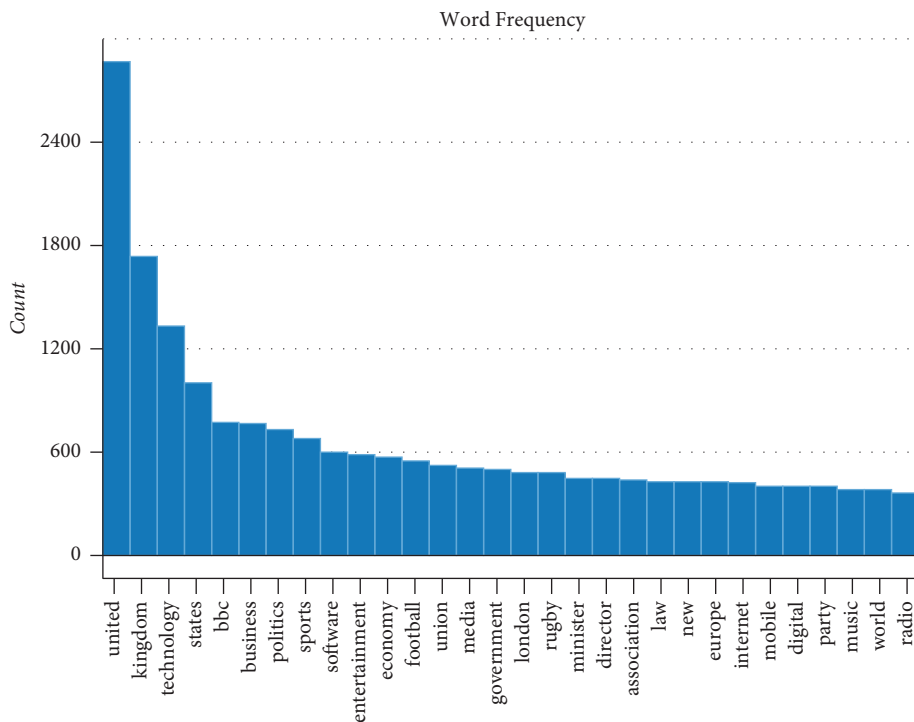FIGURE 4: The types of words ignored in the feature engineering process.

FIGURE 5: Stop words frequency.

Letter		Sample		descr	
Count	4404569	1st row	chelsea sack	categorical	
Lowercase Letter	4404569	2nd row	record fa	Show Details	
Space Separator	954519	3rd row	edu describe	Approximate Distinct Count	2128
Uppercase Letter	0	4th row	ogara reve	Approximate Unique (%)	88.3%
Dash Punctuation	0	5th row	unclear fut	Missing	0
Decimal Number	0			Missing (%)	0.0%
				Memory Size	5515738

- descr contains many words: 31406 words

- The largest value (said) is over 2.49 times larger than the second largest value (mr)

FIGURE 6: Data extracted with the proposed search function.

The dataset contains 2,128 observations, with 88.3% of them being unique patterns and no missing values. The search function identified 31,406 words, with only two stop words (said and mr). The extracted features and their descriptions using the proposed search function are shown in Figure 7. In contrast, the normal search resulted in 2,410 observations with 1,709 dash punctuation and 2,150 decimal numbers, indicating the presence of noisy features. The normal search extracted a total of 13,357 words, significantly less than the proposed search function. The number of words extracted by the normal search is lower compared to the proposed search due to the difference in feature engineering. However, both datasets are fairly balanced with a difference of only 282 observations (2,410-2,128). A comprehensive description of the characteristics of the BBC dataset has been provided in Table 2.

3.3. Crawling and Data Cleaning.

A script or program that automatically searches for information from the web based on certain rules or conditions is called a web crawler. Web crawling is advantageous for downloading web pages and achieving multithreading. This study utilized a web crawler, and specifically, the Python-based Google App Engine (PGAE) introduced by Dominic [37]. The configuration container for the web crawler included a core file that specified the interval time between two fetches, the database path directory, and the starting and ending time of crawling. The crawling frequency from the website is controlled by the interval time between two fetches. If the frequency is set too high, it may lead to the IP address being blocked. In the data engineering process, the structure of the documents was determined using HTML div and span tags to make them identifiable.

Other HTML components such as the paragraph (p) and emphasis (em) tags accurately represented the semantics of the web content. However, the use of div and span tags made the web content more accessible. The crawled features were cleaned by removing any div tags from the web pages, and the cleaned data were stored in a JSON file before being transferred to a database.

3.4. Feature Weighting with NLTK and Inverted Indexing.

The manual term weighting method was utilized, as Salton et al. [38, 39] demonstrated that manually assigning term weights is as effective as automatic weighting methods. To determine the weight of a term based on its relative frequency, the proposed approach applied Zipf's law [40]. The term weighting was computed using the following formula:

$$w = \log_{2(N)} - \log_{2(n)} + 1, \tag{5}$$

where N represents the total number of documents and n is the total number of documents containing a particular term or keyword. The NLTK library in Python was used in this study with two methods, namely, NLTK tokenize and NLTK stem. The NLTK tokenize was used to segment sentences into words, using spaces to split sentences into different words.

Similarly, the NLTK stem was used to normalize words by providing them with an acceptable format, such as changing the past tense to the present tense. The proposed search function tokenized the search information that users entered into the search engine using the NLTK tool, and the search was performed based on the tokenized words. Irrelevant words, known as stop words, were removed during the data-cleaning phase. Stop words are words that are considered to be of little significance and are thus ignored during text analysis. The most commonly found stop words include six determiners (a, that, the, an, and, those) which are used to describe nouns and express concepts related to localization or numbers in the text. However, removing stop words can improve computation efficiency and retrieval performance by reducing the number of indexes in the corpus. In this study, stop words were removed to improve the accuracy of the search function.

For instance, if a user searches for apples, a search engine might return 100 results. But if the user searches for bananas and apples, the search function may tokenize the sentence into three parts [(banana) (and) (apples)], resulting in more than 100 results. The Python library used in this study contains pretrained models and corpora that were utilized in implementing the search function. Inverted indexing was necessary for retrieving information from the database
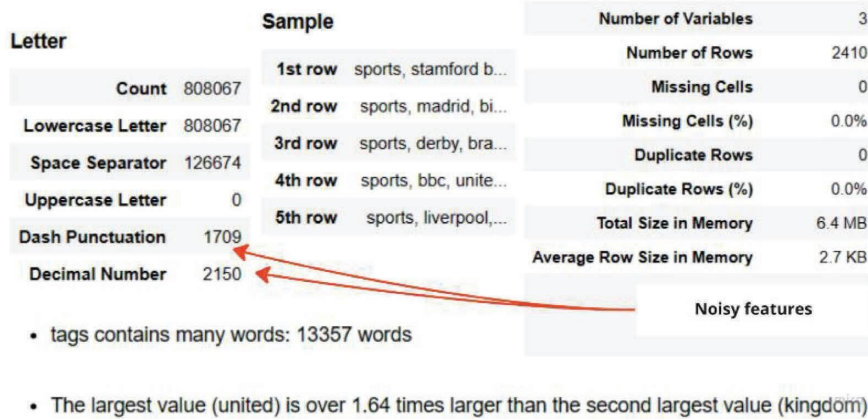
FIGURE 7: Data extracted with a normal search function.

TABLE 2: The dataset characteristics.

No.	Feature	Data type	Example	Category
1	URL	Categorical	https://...	Covid
2	Content	Categorical	News (text)	Vaccine
3	Date	Numeric	2023	Travel
4	Title	Categorical	Corona minute	Covid
5	Index	Numeric	800	Vaccine
6	Keyword	Categorical	BBC Corona	Covid

through indexes. Each entry in the news table includes a specific index that determines the location of textual data. An inverted index file was created beforehand, which allowed for efficient searching of information by the user.

The BBC articles were tokenized using NLTK, cleaned before being linked to a specific index, and stored in the database for sentiment analysis. The index facilitates rapid search results retrieval when a user searches for certain words. The table shown in Figure 8 contains the indexes, where the keywords are recorded in the *term* column, the number of indexes matching the keywords in the *df* column, and the weighted index matches in the *docs* column. For example, a user searching for BBC will receive only 697 indexes and articles as shown in Figure 8. This inverted indexing technique improves concurrency and assists in automatically generating attribute values to determine the record location.

3.5. Search Function with Sentiment Analysis. The BBC news website was crawled using a web crawler to extract patterns such as date, title, and content, and the extracted news features were then stored in a local database as shown in Figure 9. In the data preprocessing phase, the news data were preprocessed and relevant keywords were matched using the NLP module of the framework, which includes the NLTK and inverted index. The NLTK module was utilized to tokenize the BBC news into words, remove punctuation and full stops using regular expressions, convert all words to lowercase, and remove irrelevant words with no meaning. Additionally, stemming was used to normalize words to their primitive form, which reduces incomplete search

results. The inverted index was then used to efficiently and accurately search for keywords and point to relevant BBC articles.

The data wrangling phase of the framework involves a search function that is activated when keywords are entered, and the BM25 algorithm is employed to extract matching indexes from the JSON file. Each search result is assigned a score by the BM25 algorithm, which is used to measure the similarity between the keywords and the web articles. The search results are then ranked based on their scores. In the search function module, the user's keywords are compared with the prestored words in the index table. If there is a match, the corresponding index is retrieved and the matching news is located. If the entered keywords do not match any of the indexed words, the most similar index and words are located using regular expression [41]. The interface module acts as the front end of the framework, and HTML is used to search for content pages. When users enter their keywords, the search results are displayed on the content page.

Figure 9 illustrates the process of the search function that utilizes sentiment analysis. In the data collection phase, a web crawler is utilized to retrieve news from the BBC website, which is then stored in a JSON file. This file is subsequently transferred to the database. The preprocessing module is automated to enable the storage of new keywords in the database if they are similar to the embedded categories of the search function. The news features from the website are tokenized into words using the NLTK, which removes punctuation with a regular expression and converts words to lowercase. Sentence tokenization is applied to remove stop words.

| term	df	docs
sarah	7	[11, 1, 10][12, 1, 9][21, 1, 10][34, 1, 9][36, 1, 10][39, 1, 9][42, 1, 8]
everard	7	[11, 1, 10][12, 1, 9][21, 1, 10][34, 1, 9][36, 1, 10][39, 1, 9][42, 1, 8]
disappearance	2	[11, 1, 10][34, 1, 9]
met	7	[11, 1, 10][329, 1, 11][338, 1, 11][443, 1, 12][502, 1, 10][628, 1, 8][702, 1, 13]
officer	10	[11, 1, 10][18, 1, 12][293, 1, 8][342, 1, 11][347, 1, 10][369, 1, 10][382, 1, 12][59
arrested	5	[11, 1, 10][315, 1, 10][368, 1, 8][599, 1, 10][632, 1, 10]
suspicion	1	[11, 1, 10]
murder	15	[11, 1, 10][29, 1, 7][99, 1, 9][114, 1, 11][135, 1, 9][235, 1, 9][285, 1, 11][304, 1, 9
bbc	697	[11, 1, 10][12, 1, 9][13, 1, 9][14, 1, 8][15, 1, 8][16, 1, 9][17, 1, 9][18, 1, 12][19, 1,
news	697	[11, 1, 10][12, 1, 9][13, 1, 9][14, 1, 8][15, 1, 8][16, 1, 9][17, 1, 9][18, 1, 12][19, 1,
human	1	[12, 1, 9]
remains	1	[12, 1, 9]
found	11	[12, 1, 9][53, 1, 11][93, 1, 9][157, 1, 8][386, 1, 7][391, 1, 11][412, 1, 10][468, 1, 8
kent	3	[12, 1, 9][34, 1, 9][99, 1, 9]
woodland	1	[12, 1, 9]
covid	100	[13, 1, 9][19, 1, 9][22, 1, 8][32, 1, 9][47, 1, 7][58, 1, 8][61, 1, 10][72, 1, 10][78, 1,
scotland	17	[13, 1, 9][22, 1, 8][32, 1, 9][121, 1, 9][153, 1, 10][161, 1, 8][192, 1, 8][367, 1, 9][3
rules	6	[13, 1, 9][19, 1, 9][71, 1, 9][123, 1, 9][321, 1, 11][669, 1, 12]
people	11	[13, 1, 9][134, 1, 11][245, 1, 9][346, 1, 8][363, 1, 9][370, 1, 7][484, 1, 10][510, 1,
meeting	1	[13, 1, 9]
outdoors	1	[13, 1, 9]
eased	2	[13, 1, 9][321, 1, 11]
pmqs	3	[14, 1, 8][428, 1, 9][661, 1, 9]
happened	23	[14, 1, 8][61, 1, 10][98, 1, 6][155, 1, 8][168, 1, 10][202, 1, 7][246, 1, 8][270, 1, 10

FIGURE 8: The index table with weight.
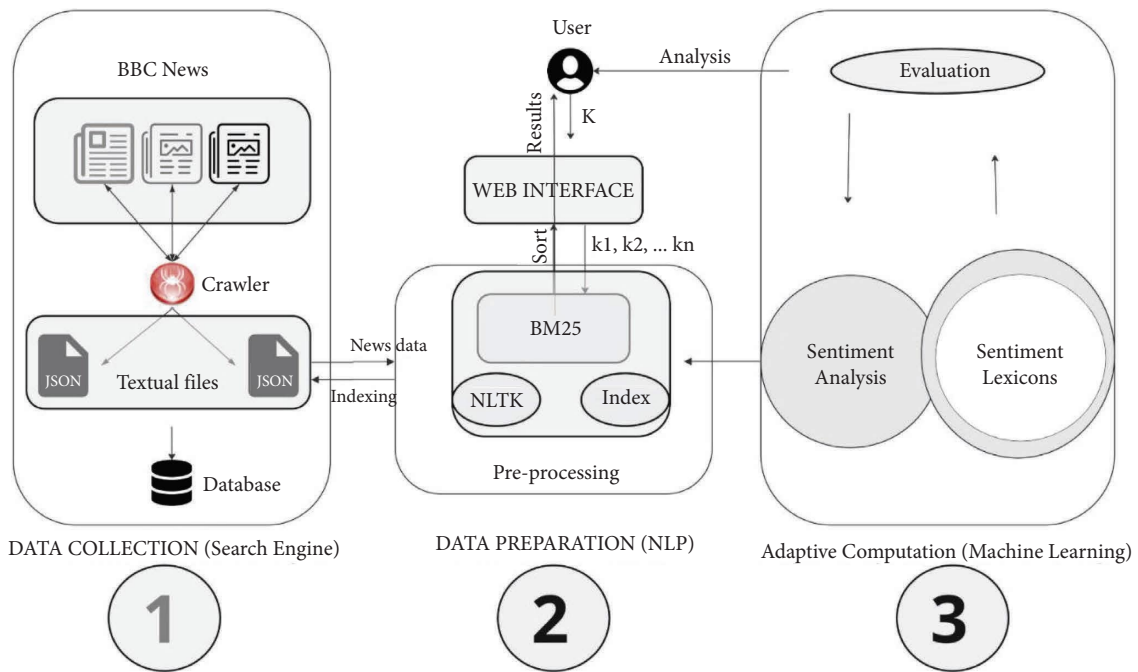
FIGURE 9: Search function and sentiment analysis.

The process of the search function using sentiment analysis is illustrated in Figure 9. The first step involves collecting data from the BBC website through web crawling and storing it in a JSON file, which is then transferred to the database. The preprocessing module is automated, such that when users enter new keywords, the system stores them in the database if they match the existing categories of the search function. The news features are tokenized into words using the NLTK and cleaned by removing punctuation and converting them to lowercase.

Stop words are removed using sentence tokenization. The text is then normalized using stemming, and cleaned words are transformed into indexes that correspond to the matched BBC article. The generated indexes are stored in a posting table in the database. In the search process, users input keywords into the web interface, which are passed to

the database to search the index table for matching keywords. The system then matches the keywords found in the database with specific web articles to provide search results, which are sorted based on relevance. Finally, the sorted search results are displayed on the web page and presented to users. Overall, the search function allows users to enter a set of keywords into the web interface, which are stored in the database and evaluated for polarity using sentiment analysis.

3.6. Sentiment Lexicons.

A sentiment lexicon is a collection of words or phrases that are associated with specific emotions or sentiments and is commonly used in sentiment analysis algorithms. The lexicon enables the algorithm to compare input words with previously labeled words in the lexicon to predict the sentiment or polarity of sentences. One such algorithm used in the proposed framework is VADER (Valence Aware Dictionary and sEntiment Reasoner) [42], SentiWordNet [43], SentiStrength [44], Li et al. lexicon [45], and Affective Norms for English Words (AFINN-111) [46]. The sentiment analysis models used in the study had their own set of lexicons. The framework applied each model to the BBC data and utilized their respective lexicons to match the keywords in the news articles to their corresponding polarity labels, in order to predict the overall sentiment of the BBC sentences.

3.7. Sentiment Analysis Algorithms.

This section presents the experimental sentiment analysis algorithms and the evaluation metrics used to assess the sentiment analysis computation. The search methodology comprises of two types of searches: the proposed optimized search function and the simple normal search, both of which produce a dataset. To address the class imbalance in the datasets, the study used feature engineering. The sentiment analysis on these datasets was then evaluated. The dataset was divided into training and testing samples, with 70% of the entire set assigned to data-70 and the remaining 30% to data-30, which served as a validation set. This approach ensured that the sentiment analysis model was not trained using the testing sets and that the validation results were independent of any discrepancies and biases. The study compared the proposed search function to the normal search procedure based on the data collected from the web. The evaluation metrics used to assess the overall experiments were precision, accuracy, recall, and F1 score. Figure 10 depicts the differences between the two searches.

The primary difference between the two search methods is the inclusion of embedded categorization. The same set of keywords, such as Covid, Vaccine, and Travel, are utilized for both search methods to ensure result reliability. The search process is divided into three clusters: Covid, Vaccine, and Travel. The Covid category contains coronavirus news stored using the Covid keywords (as shown in Figure 11). The Vaccine category includes Covid vaccine news stored using the vaccine keywords (as shown in Figure 11).

Lastly, the proposed search function stores Covid-related travel news using the Travel keyword in the Travel category (as shown in Figure 12). The computing environment used in this study is presented in Table 3. The evaluation of the

two search functions is based on the data collected from the web, and metrics such as precision, accuracy, recall, and F1 score are used to assess the overall experiments.

3.8. VADER.

VADER is a sentiment analysis program that uses a rule-based lexicon designed specifically for social media analysis. Its developers, Hutto and Gilbert [47], created the algorithm to address the challenges in analyzing symbols, languages, and text styles that are commonly found in social media. VADER is capable of detecting positive, negative, and neutral polarities of textual data. The VADER lexicon is built by selecting textual patterns from the General Inquirer (GI) and Linguistic Inquiry and Word Count (LIWC) preset lexicons and includes slang, social media abbreviations, emojis, and facial expressions such as :-) to represent positive, negative, or neutral sentiments. The lexicon consists of 7,500 textual features. To compute the polarity of a sentence using VADER, a compound score is calculated by summing equivalent values for each word in the lexicon and adjusting them according to specific rules. The polarity score is then restricted to be between the most negative (-1) and most positive (+1) values. The assigned polarity values for each row in the analyzed text are positive (pos), negative (neg), and neutral (neu), as shown in Figure 13. VADER uses a set of thresholds to assign positive, negative, or neutral polarity, as shown in Figure 14.

3.9. SentiWordNet.

SentiWordNet is an opinion-mining tool widely used in sentiment analysis that employs a lexicon derived from WordNet. This lexicon is divided into synonyms (synsets), nouns, verbs, adjectives, and other grammatical categories. The SentiWordNet algorithm utilizes the WordNet synset dictionary to combine polarity scores and determine the sentiment of the text as either negative, objective (neutral), or positive.

The algorithm generates three scores, each ranging from zero to one, using supervised machine learning methods [43]. The PosScore and NegScore represent the level of positivity and negativity associated with the text. The process depicted in Figure 15 has been used to assign sentimentality using SentiWordNet version 3.0, as explained in Figure 15. To determine whether a given text feature has a positive, negative, or neutral sentiment, the average scores of its associated synsets were used. If the positivity average score exceeds the negativity average score, then the sentiment is considered positive.

The lexicon used contained 64,000 features, and a Python code was written utilizing the NLTK package. The calculation of sentimentality for expressions in the lexicon is shown in Figure 16.

3.10. Sentistrength.

Sentistrength is an open-source sentiment analysis tool that can detect the emotions expressed in text. It has been used in the Cyber Emotions project to analyze over 14,000 social media posts with a level of accuracy comparable to that of a human. The Sentistrength lexicon is based on terms and expressions derived from the Linguistic Inquiry and Word Count (LIWC) [44, 48, 49]. Each textual pattern is assigned

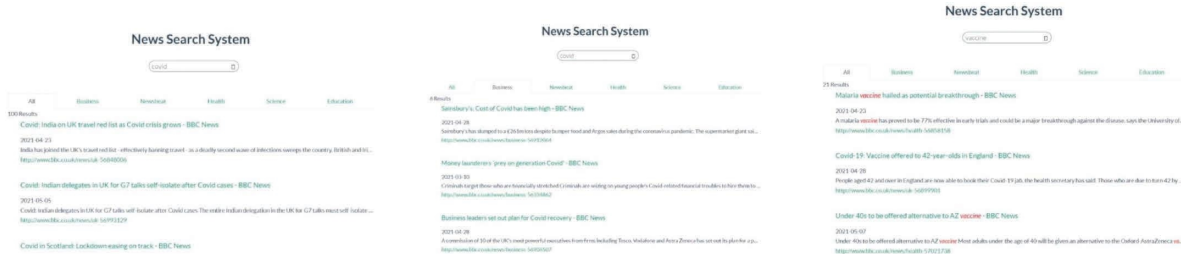|	Embedded category	Words category	Methods
Classify search function	Yes	Covid, vaccine and travel	BM25, NLTK, inverted index, and sentiment analysis
Normal search function	No		Sentiment analysis

FIGURE 10: The type of searches.
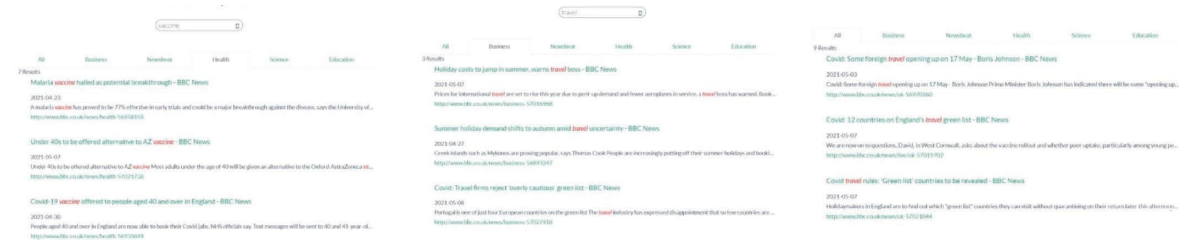

FIGURE 11: The search results and data collected.

FIGURE 12: The data collected using keywords.

TABLE 3: The computing environment.

	Component	Type	Example
1	Search engine	PGAE-based BBC website	Proposed search
		BBC website	Normal search
2	Operating system	Windows and Linux	Windows 10 and Debian
3	Algorithm	VADER, SentiWordNet, Sentistrength, Li and Hu, and AFINN-111	Classification
4	Sentiment analysis	Sentiment lexicons	Training
5	NLP	Inverted index, word2vec, NLTK, and BM25	Text segmentation and search
6	Cross-validation	70% (training) and 30% (testing)	K-fold
7	Data	BBC news	Corona text news
8	Language	Python	Python 3.9.0

```
#from vaderSentiment import SentimentIntensityAnalyzer
sentences=
["The book was good" #positive example
,"VADER is not smart nor funny." #negative example
,"Today only kinda sux! But I'll get by, lol" #mixed
example ]

analyzer = SentimentIntensityAnalyzer()
for sentence in sentences:
scores = analyzer.polarity_scores(sentence)
print(sentence + str(scores))
```

The following table shows the result after applying the previous code:

Sentence	pos	compound	neu	Neg
The book was good	0.492	0.4404	0.508	0.0
VADER is not smart nor funny.	0.0	-0.7424	0.354	0.646
Today only kinda sux!But I'll get by, lol	0.317	0.5249	0.556	0.127

FIGURE 13: The polarity computation using VADER.

```
--------------------------------------------------
(i)   Positive: compound score > = 0.0.5
--------------------------------------------------
(ii)  Neutral: (compound > -0.05) and (compound < 0.05)
--------------------------------------------------
(iii) Negative: compound score <= - 0.05
--------------------------------------------------
```

FIGURE 14: The polarity threshold.

scores for negative, positive, and neutral polarities. Sentistrength predicts the negative and positive polarities in short texts using specific rules, which are as follows:

Negative (−1) to extremely negative (−5)

Positive (1) to extremely positive (5)

```
-------------------------------------------------
PosScore [0,1] : positivity scale
NegScore [0,1]: negative scale
ObjScore [0,1]: objective/neutral scale
-------------------------------------------------

Where the degree of objectivity can be calculated as
follows:

ObjScore = 1 - (PosScore + NegScore).
-------------------------------------------------
```

FIGURE 15: The SentiWordNet computation.

```
---------------------------------------------------
<very.r.01: PosScore=0.25 NegScore=0.25>
<nice.a.01: PosScore=0.875 NegScore=0.0>
<love.v.02: PosScore=1.0 NegScore=0.0>
<worse.a.01: PosScore=0.0 NegScore=0.75>
<bad.a.01: PosScore=0.0 NegScore=0.625>
<truly.r.01: PosScore=0.625 NegScore=0.0>
---------------------------------------------------
```

FIGURE 16: The polarity of SentiWordNet.

The Sentistrength algorithm requires placing textual features in a plain text file (one text per line) to predict text polarities. The resulting output is a copy of the file containing negative and positive predictions at the end of each line. As shown in Figure 17, the negative and positive values of each text can be predicted using Sentistrength. The final and averaged polarity of the file can be obtained with different point scales ranging from −5 (negativity) to +5 (positivity), where 0 represents neutral polarity. Previous experiments have demonstrated the robust performance of Sentistrength in web mining [44, 49]. This tool is available in both Java and Windows formats, with this study utilizing the publicly accessible Windows version, which can be found at http://sentistrength.wlv.ac.uk/. The Sentistrength lexicon is presented in Figure 18.

3.11. Liu and Hu Lexicon. The Liu and Hu lexicon used in this study consists of two word lists: one for positive predictions and the other for negative predictions [50]. It was developed by researchers from the Computer Science Department at the University of Illinois [50]. The Python code used to access the negative and positive lexicon words is demonstrated in Figure 19. To predict the sentiment of texts, a Python code was created to compare the BBC texts with the labeled lexicon words using the NLTK.

3.12. AFINN-111. AFINN-111 consists of a collection of English words that have been assigned a sentiment score ranging from −5 (negative) to +5 (positive) [51]. This lexicon was developed manually by Finn Arup Nielsen over the course of 11 years from 2000 to 2011 [51]. The lexicon includes two lists, AFINN-111 containing 2,477 phrases and words, and AFINN-96 with 1,468 unique phrases and words consisting of 1,480 lines. The list includes two types of columns: the word itself and its corresponding polarity value, which is depicted in the range of [−5, +5]. Figure 20 displays the AFINN-111 list.

Text	Positive	Negative
using Linux and loving it - so much nicer than windows... Looking forward to using the wysiwyg latex editor!	4	-1
@ruby_gem My primary debit card is Visa Electron.	1	-1
@kirstiealley I hate going to the dentist.. !!!	1	-4

FIGURE 17: The Sentistrength computation.

List Name	Sample Word	Score
Sentiment strength word list	Awful	-4
	Blissful	+5
Booster word list	Slightly	-1
	Extremely	+2
Idiom List	Shock horror	-2
	Whats good	+2
Negation word list	Cant	-
	Never	-
Emoticon word list	:'(	-1
	:-D	+1

FIGURE 18: The Sentistrength lexicon.

```
>>> from nltk.corpus import opinion_lexicon
>>> opinion_lexicon.words()
['2-faced', '2-faces', 'abnormal', 'abolish', . . .]
>>> opinion_lexicon.negative()
['2-faced', '2-faces', 'abnormal', 'abolish', . . .]
>>> opinion_lexicon.positive()
['a+', 'abound', 'abounds', 'abundance', 'abundant', . . .]
```

FIGURE 19: The Liu and Hu lexicon.

Based on the analysis conducted, SentiWordNet was found to have the highest occurrence of positive and neutral textual patterns (Figure 21), whereas Sentistrength exhibited the highest frequency of negative patterns (Figure 21). It is important to note that this finding suggests that the size of sentiment analysis lexicons can significantly impact the classification or prediction results obtained, and researchers should carefully consider the size and characteristics of each lexicon before making a selection.

Additionally, it highlights the importance of selecting appropriate lexicons for sentiment analysis to ensure the accuracy and validity of the results. It is important to select appropriate lexicons for sentiment analysis because the accuracy and effectiveness of sentiment analysis largely depend on the quality and suitability of the lexicon used. A lexicon is a dictionary or database that contains words and phrases along with their assigned sentiment polarity (positive, negative, or neutral) [11]. Using an inappropriate lexicon can lead to inaccurate sentiment analysis results, which can have serious consequences in various domains such as business, politics, and healthcare. For example, misinterpreting customer sentiments in business can result in a loss of revenue and reputation, while misinterpreting patient sentiments in healthcare can lead to incorrect diagnoses and treatments. Therefore, selecting the right lexicon that is tailored to the specific domain and language is crucial for obtaining accurate sentiment analysis results. It is also important to continuously update and improve the lexicon as language and context evolve over time.

```
----------------------------------------
Word                         Value
----------------------------------------
Lose                           -3
Loses                          -3
Loser                          -3
Losing                         -3
Loss                           -3
Lost                           -3
Lovable                         3
Love                            3
Loved                           3
Loving                          2
Lowest                         -1
----------------------------------------

The following Python code demonstrates how to calculate
the degree of emotion of a positive statement of degree 3:


----------------------------------------
afinn= dict(map(lambda kv:(kv[1])),[line.split('\t')
for line in open("AFINN-111.txt")]))
score=sum(map(lambda word: afinn.get(word, 0), "
AFINN is very good".lower().split()))
print(score) // 3
----------------------------------------
```

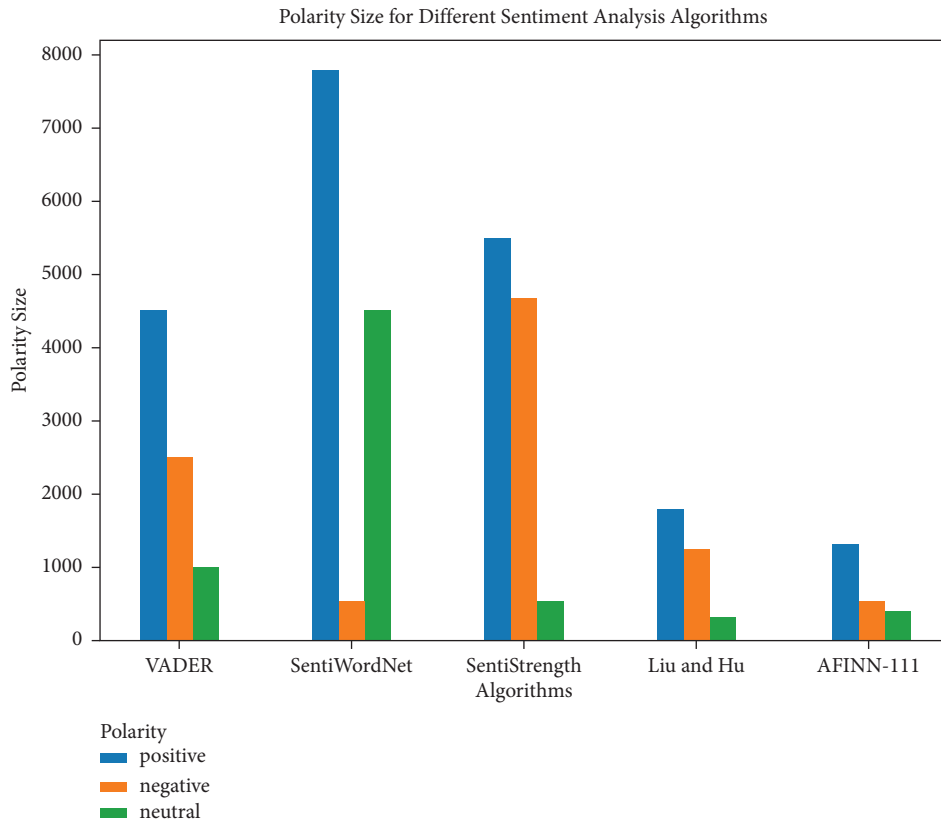FIGURE 20: The AFINN-111 lexicon and computation.

FIGURE 21: The size of lexicons.

3.13. Evaluation Metrics. In the experiments, precision (P), accuracy (A), F1 score (F), and recall (R) [52, 53] are used as evaluation metrics. Precision represents the number of true predicted classes that belong to the accurate class. Recall, on the other hand, determines the percentage of correctly classified categories [52, 54]. It is the number of observations that the sentiment analysis model correctly predicted divided by the total number of observations [55, 56].

$$P = \frac{TP}{TP + FP},$$
$$R = \frac{TP}{TP + FP}. \tag{6}$$

Recall measures the number of instances that the sentiment analysis model predicted correctly and is in a particular class, divided by the total number of instances that actually belong to that class [55, 57]. True positive (TP) refers to instances where the model accurately predicts the positive class, while true negative (TN) refers to instances where the model accurately predicts the negative class [58]. False positive (FP) refers to instances where the model incorrectly predicts the positive class, and false negative (FN) refers to instances where the model incorrectly predicts the negative class [59, 60]. Accuracy measures the overall classification/prediction success of the sentiment analysis and is calculated using the following equation:

$$A = \frac{TP + TN}{TP + TN + FP + FN}. \tag{7}$$

The F1 score is a metric that combines both precision and recall values of the sentiment analysis model. This score is calculated as the harmonic mean of the model's precision and recall, as shown in the following equation:

$$F = \frac{2 \times P \times R}{P + R}. \tag{8}$$

4. Results

The results of sentiment analysis on the BBC data are presented in this section. The experiment employed sentiment analysis lexicons without any modifications to analyze the data. Additionally, a comparative analysis was conducted to determine the best-performing sentiment analysis algorithm. Table 4 displays the evaluation metrics obtained from both BBC samples.

The sentiment analysis performance is summarized in Figure 22. The lexicons were used without any modifications to analyze the data obtained using the proposed search function. The results showed that the VADER model had the highest accuracy of 85%, whereas the precision performance of the Sentistrength, AFINN-111, and Liu and Hu's models was similar (as shown in Table 4). The F1 score of the SentiWordNet model was 65% using the data obtained through the proposed search (Table 4). The AFINN-111 model performed well in the positive sample rating, achieving an accuracy of 78% with the proposed search data (Table 4). However, the Sentistrength model did not perform well in classifying positivity/neutrality with the proposed search data, with an accuracy range of 10%–15% (as shown in Table 4). When analyzing data obtained through a normal search, the precision of the AFINN-111 and VADER models was higher and closer than that of the other three models.

The precision for negative and positive classification using normal search data was best for VADER and SentiWordNet models, with 69% and 65%, respectively (Table 4). However, AFINN-111 showed the highest precision of 60% for neutral classification using both types of data (Table 4).

The performance of Liu and Hu's model was low, as evident from Table 4.

Furthermore, it is worth noting that the performance of the Liu and Hu model's lexicon improved when using data extracted with the proposed search, as compared to a normal search. The utilized sentiment analysis lexicons and models showed promising results, even without preprocessing the data extracted with a normal search. The comparative analysis suggests that the VADER lexicon is a strong candidate for accurate classification of positive and negative news. The experiment also demonstrates the AFINN-111 model's proficiency in rating more positive samples when using the proposed search data.

Furthermore, the Sentistrength model achieved the highest accuracy (75%) in classifying negative samples, as shown in Table 4. However, Liu and Hu's model performed poorly compared to other techniques.

The performance of the VADER model varied with the two different BBC samples, and the model's performance decreased when using data extracted with a normal search. Figure 23 illustrates that the VADER model had a misclassification rate of 15.8% for features extracted by the proposed search and a misclassification rate of 20.3% for patterns retrieved by a normal search.

These results demonstrate the effectiveness of the proposed search function in enhancing the structure and quality of BBC data compared to a normal search. The NLP techniques implemented in the data preprocessing stage led to a decrease in the misclassification rate of sentiment analysis.

Table 5 displays the quantity of textual characteristics obtained by two types of searches, covering the period from 2020 to 2023. The table includes the search terms used for data collection, along with the number of URLs retrieved and used in the process. The content column shows the number of pertinent words (text) employed by the sentiment analysis model for news categorization. This table is a complement to Table 4, providing further information. As an example, news categories such as Covid, Vaccine, and Travel, as well as website titles, were considered. The optimal confusion matrix for sentiment analysis is depicted in Figure 24, illustrating the number of textual characteristics that have been accurately predicted/classified by the sentiment analysis as belonging to the previously mentioned news categories. Specifically, the confusion matrix for the best sentiment analysis technique, such as SentiWordNet, reveals that out of 16,744 textual features, the majority of them were correctly predicted to belong to the Vaccine category.

According to Tables 4 and 5, this forecast has a negative polarity of 68%. Notably, SentiWordNet outperformed other models by predicting 16,744 words with a negative polarity of 68% accuracy or precision concerning the COVID-19 vaccine. Among the sentiment analysis techniques, AFINN-111 required the longest execution time (50 seconds), as shown in Table 4.

Therefore, the efficiency of sentiment analysis models in terms of time complexity and misclassification rates can serve as a crucial performance indicator for news categorization and classification. In general, the evaluation metrics

TABLE 4: Evaluation metrics in the BBC dataset.

| Sentiment analysis	Data extracted with the proposed search						Sentiment classification accuracy		
	Unclassified words	Time (sec)	Accuracy	Recall	Precision	F1 score	Positive	Negative	Neutral
VADER	1,235	30	85%	0.84	0.77	0.78	85%	10%	5%
SentiWordNet	1,865	32	68%	0.64	0.75	0.65	12%	68%	20%
Sentistrength	1,765	31	75%	0.77	0.70	0.73	15%	75%	10%
AFINN-111	1,543	33	78%	0.76	0.74	0.72	78%	12%	10%
Liu and Hu	1,398	35	50%	0.69	0.63	0.62	50%	0%	50%
Sentiment analysis	Data extracted with a normal search						Sentiment classification precision		
	Unclassified words	Time (sec)	Accuracy	Recall	Precision	F1 score	Positive	Negative	Neutral
VADER	2,657	40	82%	0.72	0.69	0.68	69%	1%	30%
SentiWordNet	2,543	43	63%	0.54	0.65	0.65	15%	65%	20%
Sentistrength	2,436	42	73%	0.67	0.62	0.60	62%	18%	20%
AFINN-111	2,873	50	75%	0.66	0.60	0.63	10%	30%	60%
Liu and Hu	2,603	48	45%	0.59	0.51	0.53	9%	40%	51%
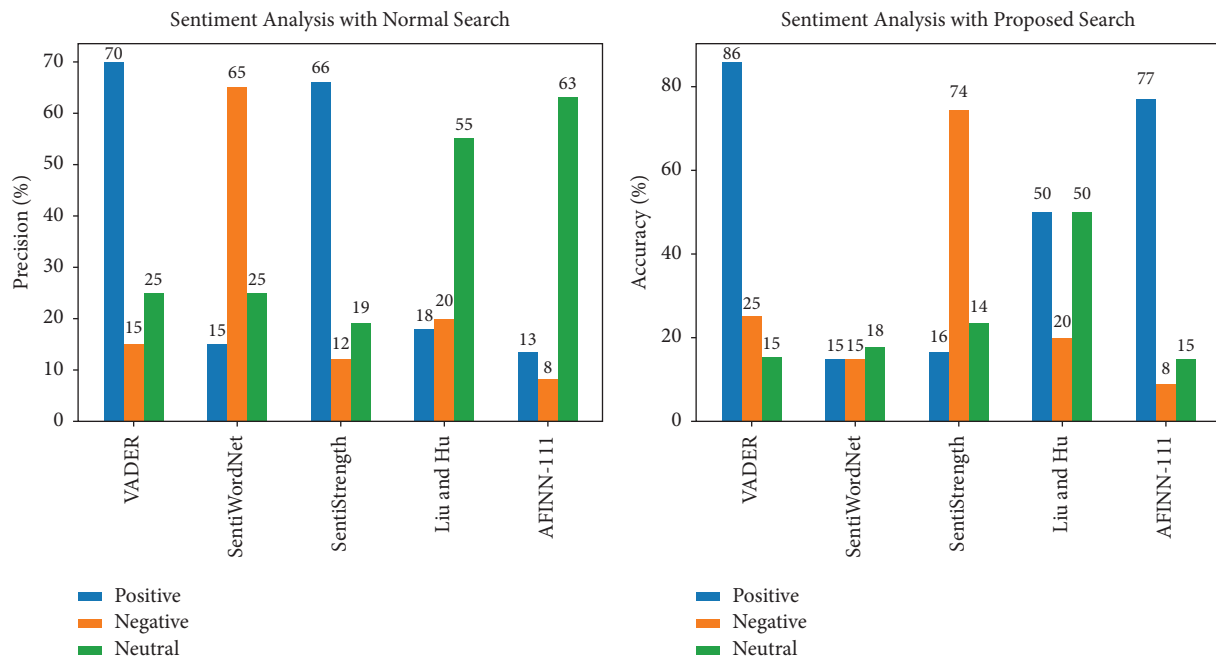
FIGURE 22: The sentiment analysis results.

exhibit a minor improvement when utilizing data obtained through the proposed search function, as illustrated in Figure 25.

The outcomes indicate that the proposed search function retrieves a greater number of URLs, content, and indexes than a standard search, as presented in Table 5. This finding highlights the significance of feature engineering in news classification and categorization. By utilizing NLP for feature engineering, the accuracy and precision of sentiment analysis classification can be enhanced. Crawling with a standard search required more time compared to the proposed search, mainly due to the absence of preprocessed patterns in a standard search. The primary reason for the superior performance of the proposed search function

compared to a standard search is the categorization of features recorded by the proposed search function. As a result, sentiment analysis models were able to achieve a higher level of accuracy and precision.

Among the five models, VADER implementation, SentiWordNet, and Sentistrength techniques achieved the most promising outcomes in terms of the number of URLs and textual content classified. VADER employed 8,000 textual patterns, followed by the SentiWordNet and Sentistrength models, as indicated in Table 5. Adjectives were found to be the most relevant linguistic pattern for sentiment analysis. Nevertheless, combining adverbs and adjectives can also enhance the sentiment analysis classification. An essential issue, however, is that adjectives
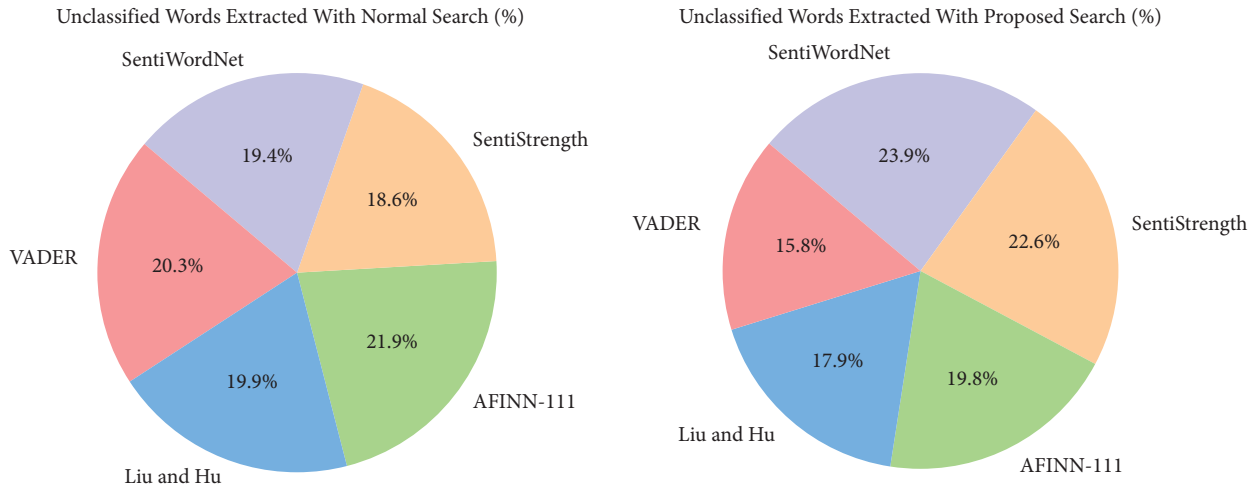
Unclassified Words Extracted With Normal Search (%)

Unclassified Words Extracted With Proposed Search (%)

FIGURE 23: Unclassified words.

TABLE 5: Prediction category in the BBC dataset.

| Sentiment analysis	Data extracted with the proposed search						Sentiment classification accuracy		
	URL	Content (text)	Date	Title	Index	Keyword	Covid	Vaccine	Travel
VADER	285	8,000	2021–2023	Corona virus	19	Covid	85%	10%	5%
SentiWordNet	653	5,546	2022	COVID vaccine	17	Vaccine	12%	68%	20%
Sentistrength	658	4,888	2020–2023	Covid vaccine	30	Covid	15%	75%	10%
AFINN-111	27	530	2020–2022	Travel for Corona	72	Travel	78%	12%	10%
Liu and Hu	60	1,043	2023	Covid-19 in Africa	200	Covid	50%	0%	50%
Sentiment analysis	Data extracted with a normal search						Sentiment classification precision		
	URL	Content (text)	Date	Title	Index	Keyword	Covid	Vaccine	Travel
VADER	185	4,000	2021–2023	Corona virus	10	Covid	85%	10%	5%
SentiWordNet	323	2,435	2022	COVID vaccine	8	Vaccine	12%	68%	20%
Sentistrength	312	1,212	2020–2023	Covid vaccine	12	Covid	15%	75%	10%
AFINN-111	13	255	2020–2022	Travel for Corona	45	Travel	78%	12%	10%
Liu and Hu	40	623	2023	Covid-19 in Africa	93	Covid	50%	0%	50%
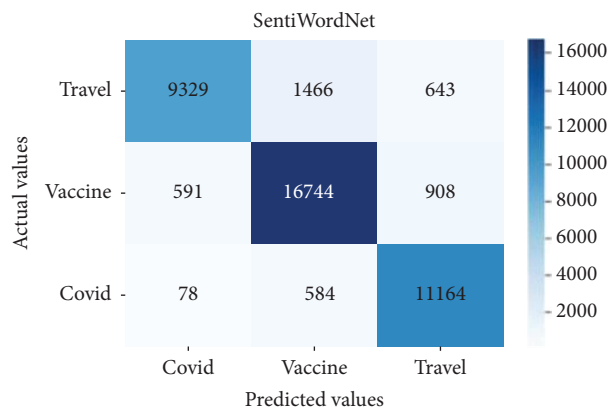
FIGURE 24: The confusion matrix of the SentiWordNet model.

and adverbs should not carry equal weight in predicting the sentiment of a given text. Rather, the scores of adjectives and adverbs should be combined using appropriate feature weighting techniques to achieve optimal results.

The evaluation of five sentiment analysis models using available lexicons revealed that Sentistrength and Senti-WordNet obtained comparable accuracy and precision for both types of search, while Liu and Hu's implementation
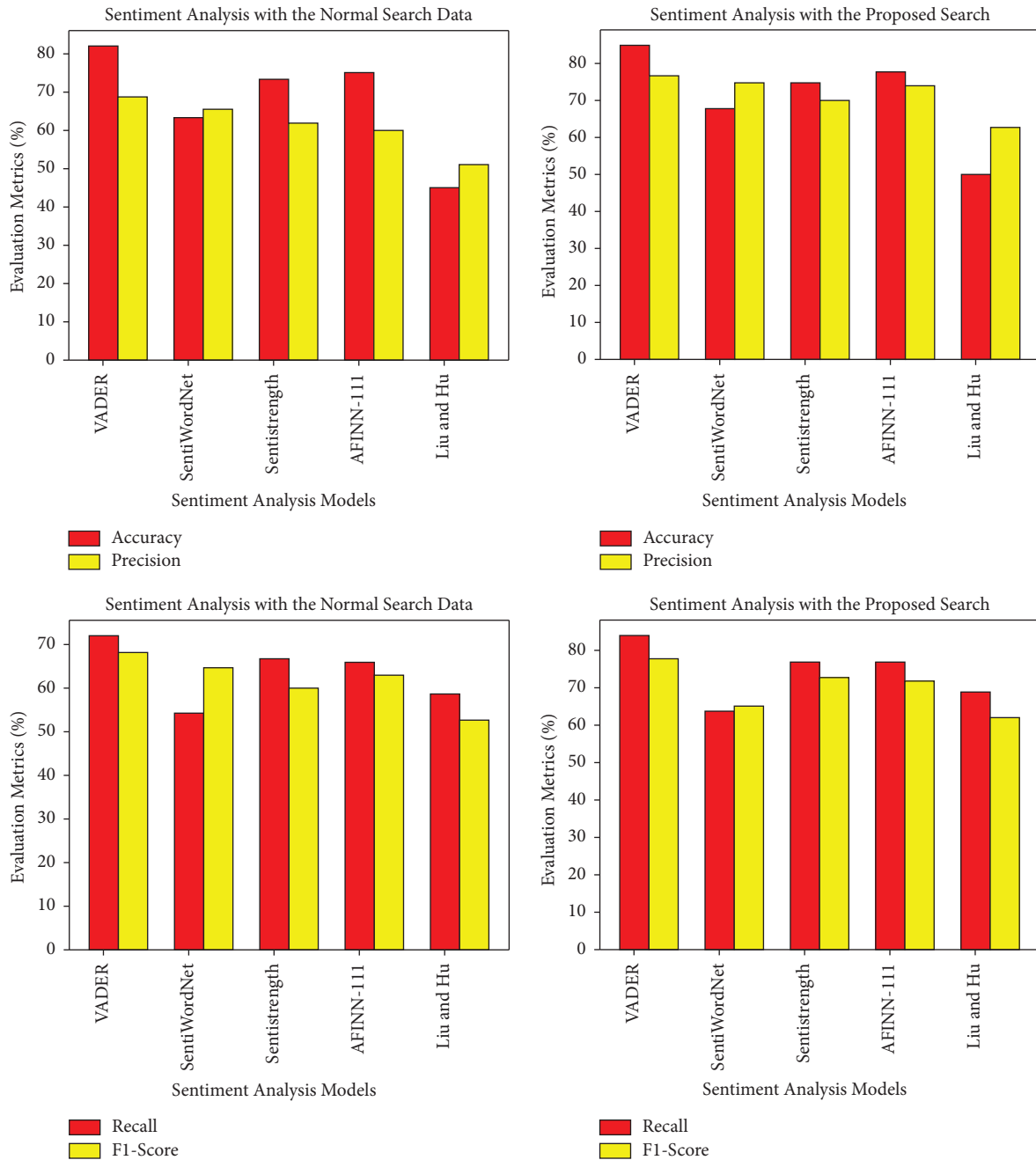
FIGURE 25: Summary of the sentiment analysis performance.

achieved good accuracy and precision levels for Covid and Travel categories. These findings led to two significant conclusions: (i) VADER-based sentiment analysis model can achieve similar accuracy for both types of BBC data and (ii) adverbs can improve precision levels when weighted and combined with adjective scores.

Adverbs were identified as the most critical linguistic feature for sentiment extraction. The use of NLP lexicons facilitated sentiment analysis computation, highlighting the importance of using a smart search function that can retrieve nonpolysemic search results and provide accurate automated sentiment analysis.

4.1. Discussion. To summarize, the study found that VADER was the most efficient and accurate sentiment analysis model for news classification (Figure 26). AFINN-111 performed well in predicting positive polarity with a high accuracy of 78% when using structured data retrieved with the proposed search function (Figure 26). Liu and Hu's model did not perform well due to the limitations of its lexicons. Overall, the VADER, SentiWordNet, Sentistrength, and AFINN-111 lexicons showed promise for accurately classifying news based on their positive and negative sentiments. The best-performing model in classifying more positivity for the Covid category was the AFINN-111 with 78% of accuracy
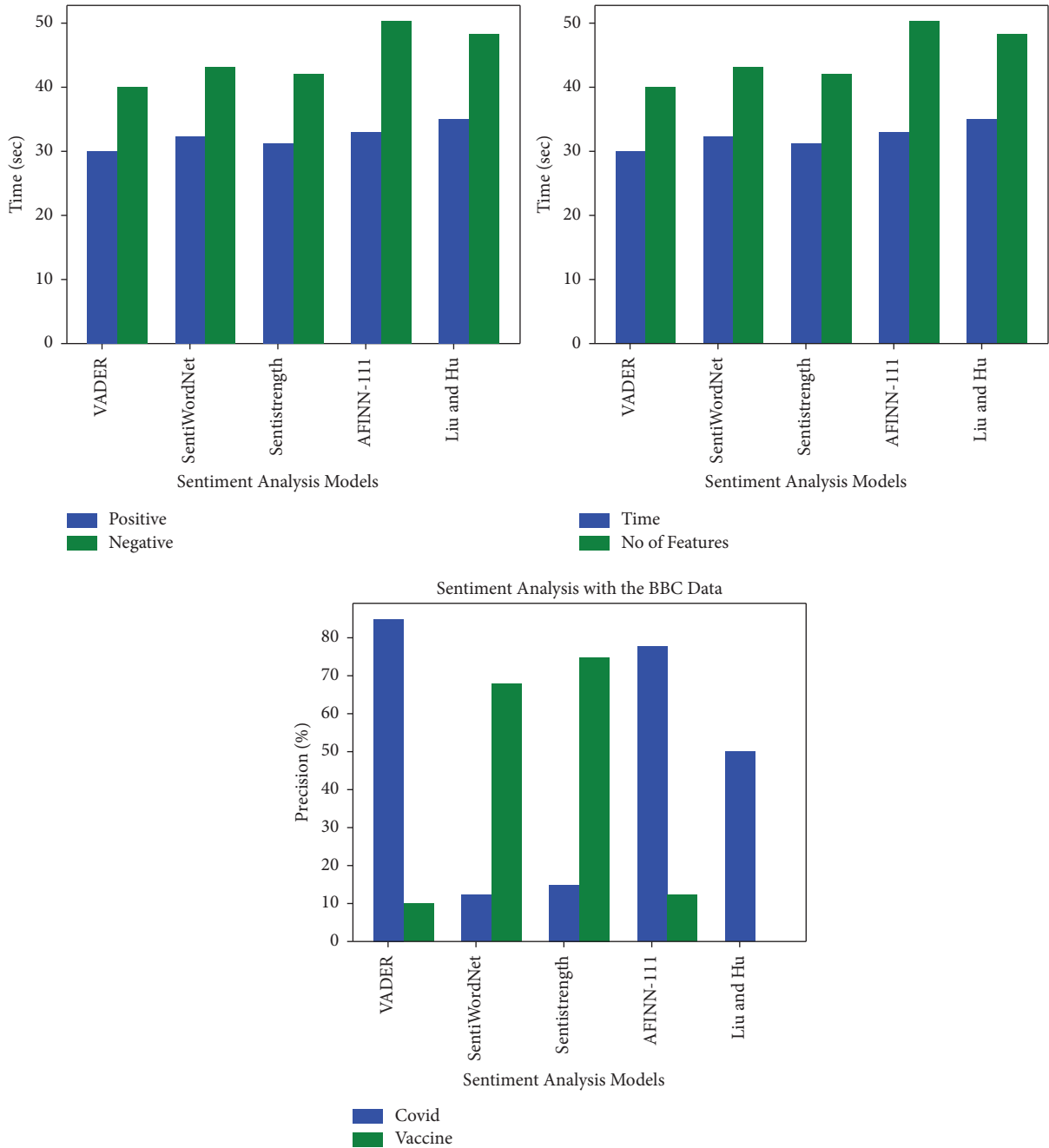
FIGURE 26: Time execution and precision of sentiment analysis.

(Figure 26). This result confirms the effectiveness of the proposed search function in improving the structure and quality of textual data compared to a normal search. The experiment shows the confusion matrix of the best sentiment analysis technique such as SentiWordNet which considered more textual features that were correctly predicted to belong to the Vaccine category. The proposed search function was effective in improving the structure and quality of textual data compared to a normal search. The SentiWordNet technique had the best confusion matrix and was able to correctly predict more textual features to belong to the Vaccine category. The study suggests that the time

complexity, lexicon structure, and misclassification rates of sentiment analysis models are important indicators for gauging the efficiency of a sentiment analysis framework. Table 6 presents a comparative analysis with existing studies to demonstrate the superior performance of the proposed methodology.

Among the five models, VADER, SentiWordNet, and Sentistrength obtained the best results, outperforming some existing techniques. However, the study's focus was on using sentiment analysis as a subset of NLP to enhance the search engine's performance and provide users with a better perspective on the crawled data. The study used sentiment

TABLE 6: Comparative experimental analysis.

Author	Method	Data	Accuracy (%)	Advantage	Limitation
Baccianella et al. [43]	SentiWordNet	Twitter	63	Time	Error
Ayyoub et al. [61]	Lexicons	Tweets	70	Accuracy	Arabic
Ščepanović et al. [62]	Sentistrength	Tweets logs	76	Precision	Polysemy
Islam and Zibran [63]	AFINN	Text	74	Error	Weighting
Charlothorn and Ellman [64]	Sentistrength	Web	70	Time	Accuracy
Hutto and Gilbert [47]	VADER	BBC	69	3-D	Noisy
Stein and Zu Eissen [65]	Topic identification	NLP	72	Accuracy	Engineering
Alan et al. [15]	Summarizing	Text	69	Detection	Engineering
Ferreira et al. [66]	Summarizing	Text	82	Time	Engineering
This work	VADER, SentiWordNet, and AFINN-111	BBC news	85	Search	Embedded category
Bi [11]	Triplet belief	Review	90	Precision	Adaptability
Nkongolo [13]	Neural network	Search results	89	Precision	Manual data collection
	Anomaly detection	Search results	90	Speed	Engineering

analysis in combination with other NLP techniques such as text segmentation and POS (Part-of-Speech) to optimize the search engine's smart search function. The aim was to develop a search engine with embedded sentiment analysis that could retrieve nonpolysemic results and classify them using sentiment analysis models [43, 47, 63–66]. A search engine with embedded sentiment analysis can automatically classify emotions from search results into negative, positive, or neutral categories. However, accurately determining the valence or emotions of text can be difficult due to the subjectivity of language, leading to false positive and negative rates. This means that even with updated lexicons, sentiment analysis models may still have issues predicting the polarity of statements. Sentiment analysis has various practical applications in search engine optimization. For instance, it can be used for real-time assessment of brand reputation, improving customer support interactions, identifying customer needs and anomalies, and monitoring customer behavior using web intelligence. Nevertheless, the absence of lexicons in languages such as French, Spanish, Chinese, and Swahili will impede the deployment of smart search engines and the contextualization of sentiment analysis outcomes. This will lead to inaccuracies in sentiment computation due to false positives and negatives. Moreover, building a robust and smart sentiment analysis framework for search engines requires tackling challenges such as polysemy, irony, sarcasm, and multipolarity.

5. Conclusion

This manuscript presents a topic that involves several aspects of NLP and sentiment analysis. The author has employed a web crawler to collect BBC news across the Internet and carried out preprocessing of text by using NLP and sentiment analysis methods to determine the polarity of the processed text data. The BBC website was utilized to crawl news data using a proposed smart search function. The experimental results portray this search function collecting more than 2,000 textual patterns for sentiment analysis. The proposed function outperformed a normal search in terms of feature quality. However, this study can be extended by embedding more categories in the search function to collect a large amount of data from the entire Internet. In fact, the development of effective methods and functions for each aspect aforementioned in this study is always challenging. Even if experiments were performed with a single database, the author still believe that some of the results presented in this article will be used as a reference in the field of search engine optimization. Different users will have different needs, and traditional search methods cannot satisfy all needs due to polysemous problems, which lead to poor user experiences. To address this issue, the research implemented a search function enabling users to extract relevant BBC information that meets their needs. In addition, the web has a rich and diverse set of news, including fake ones. Unfortunately, the proposed search function cannot remove fake news from the search results. Another limitation is linked to the number of categories embedded in the proposed search, which do not include all groups that are relevant to users. However, the proposed list of categories such as Covid,

Vaccine, and Travel can still be extended to improve the search efficiency and reliability. Furthermore, the proposed function can be improved by adding an advanced classification scheme to the search with intelligent tagging. The tagging mechanism should be automated with machine learning where the classifier can be trained to tag searched contents. Moreover, the optimized search function using machine learning can also be used to study the strength of an entity's web presence. In future works, one can automatically create search categories using artificial intelligence and predict users' behavior with machine learning. An online corporate reputation analysis can also be implemented with the proposed methodology. Additional lexicons such as Google Play Opinion Mining Score (GPOMS) and Opinion Finder can also be added to the sentiment analysis models to improve the sentiment analysis performance. Lastly, ensemble learning can also be utilized to improve the machine learning accuracy on the BBC dataset.

Data Availability

The data supporting the current study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

References

[1] A. Dima, S. Lukens, M. Hodkiewicz, T. Sexton, and M. P. Brundage, "Adapting natural language processing for technical text," *Applied AI Letters*, vol. 2, no. 3, p. e33, 2021.

[2] R. T. McCoy, E. Pavlick, and T. Linzen, "Right for the wrong reasons: diagnosing syntactic heuristics in natural language inference," 2019, https://arxiv.org/abs/1902.01007.

[3] H. Chin, G. Lima, M. Shin et al., "User-chatbot conversations during the COVID-19 pandemic: study based on topic modeling and sentiment analysis," *Journal of Medical Internet Research*, vol. 25, Article ID e40922, 2023.

[4] B. Rawat, A. S. Bist, U. Rahardja, Q. Aini, and Y. P. A. Sanjaya, "Recent deep learning based NLP techniques for chatbot development: an exhaustive survey," in *Proceedings of the 2022 10th International Conference on Cyber and IT Service Management (CITSM)*, vol. 1–4, Yogyakarta, Indonesia, September 2022.

[5] F. R. Rahat, M. S. Mim, and A. Mahmud, *Data Analysis Using NLP to Sense Human Emotions through Chatbot*, Springer, Berlin, Germany, 2022.

[6] P. Rathee and S. K. Malik, *An Analysis of Semantic Similarity Measures for Information Retrieval*, Springer, Berlin, Germany, 2023.

[7] C. Sotis, "How do Google searches for symptoms, news and unemployment interact during COVID-19? A Lotka–Volterra

analysis of google trends data," *Quality and Quantity*, vol. 55, no. 6, pp. 2001–2016, 2021.

[8] A. Li, X. Wei, D. Wu, and L. Zhou, "Cross-modal semantic communications," *IEEE Wireless Communications*, vol. 29, no. 6, pp. 144–151, 2022.

[9] Y. Bi, J. Guan, and D. Bell, "The combination of multiple classifiers using an evidential reasoning approach," *Artificial Intelligence*, vol. 172, no. 15, pp. 1731–1751, 2008.

[10] Y. Bi, "The impact of diversity on the accuracy of evidential classifier ensembles," *International Journal of Approximate Reasoning*, vol. 53, no. 4, pp. 584–607, 2012.

[11] Y. Bi, "Sentiment classification in social media data by combining triplet belief functions," *Journal of the Association for Information Science and Technology*, vol. 73, no. 7, pp. 968–991, 2022.

[12] N. Amin, T. Yother, M. Johnson, and J. Rayz, "Exploration of natural language processing (NLP) applications in aviation," *Collegiate Aviation Review International*, vol. 40, no. 1, 2022.

[13] M. Nkongolo, "Classifying search results using neural networks and anomaly detection," *Educor Multidisciplinary Journal*, vol. 2, no. 1, pp. 102–127, 2018.

[14] Z. Liu, Y. Lin, M. Sun, Z. Liu, Y. Lin, and M. Sun, "Representation learning and nlp," *Representation Learning for Natural Language Processing*, vol. 21, pp. 1–11, 2020.

[15] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, New York, NY, USA, June 2003.

[16] S. Taj, B. B. Shaikh, and A. F. Meghji, "Sentiment analysis of news articles: a lexicon based approach," in *Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, July 2019.

[17] V. Dogra, S. Verma, and P. Chatterjee, "A complete process of text classification system using state-of-the-art NLP models," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1883698, 26 pages, 2022.

[18] H. Xu and Y. Lv, "Mining and application of tourism online review text based on natural language processing and text classification Technology," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 9905114, 13 pages, 2022.

[19] M. T. Ribeiro and S. Lundberg, "Adaptive testing and debugging of NLP models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3253–3267, Dublin, Ireland, July 2022.

[20] Z. Xu, B. Chen, S. Zhou et al., "A text-driven aircraft fault diagnosis model based on a word2vec and priori-knowledge convolutional neural network," *Aerospace*, vol. 8, no. 4, p. 112, 2021.

[21] Z. Jalil, A. Abbasi, A. R. Javed et al., "Covid-19 related sentiment analysis using state-of-the-art machine learning and deep learning techniques," *Frontiers in Public Health*, vol. 9, Article ID 812735, 2021.

[22] S. Reddy, R. Bhaskar, S. Padmanabhan et al., "Use and validation of text mining and cluster algorithms to derive insights from corona virus disease-2019 (COVID-19) medical literature," *Computer Methods and Programs in Biomedicine Update*, vol. 1, Article ID 100010, 2021.

[23] D. Yogesh Pawade, "Analyzing the impact of search engine optimization techniques on web development using experiential and collaborative learning techniques," *International Journal of Modern Education and Computer Science*, vol. 13, no. 2, pp. 1–10, 2021.

[24] Y. Li, "Toward a qualitative search engine," *IEEE Internet Computing*, vol. 2, no. 4, pp. 24–29, 1998.

[25] Y. Huang, X. Li, and G. Q. Zhang, "ELII: a novel inverted index for fast temporal query, with application to a large Covid-19 EHR dataset," *Journal of Biomedical Informatics*, vol. 117, Article ID 103744, 2021.

[26] I. Dilrukshi and K. De Zoysa, "Twitter news classification: theoretical and practical comparison of SVM against Naive Bayes algorithms," in *Proceedings of the 2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka, December 2013.

[27] K. K. Bun and M. Ishizuka, "Topic extraction from news archive using TF∗PDF algorithm," in *Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002*, pp. 73–82, Singapore, December 2002.

[28] J. Kapusta and J. Obonya, "Improvement of misleading and fake news classification for flective languages by morphological group analysis," *Informatics*, vol. 7, no. 1, p. 4, 2020.

[29] S. Daud, M. Ullah, A. Rehman, T. Saba, R. Damaševičius, and A. Sattar, "Topic classification of online news articles using optimized machine learning models," *Computers*, vol. 12, no. 1, p. 16, 2023.

[30] M. Yousef and D. Voskergian, "TextNetTopics: text classification based word grouping as topics and topics' scoring," *Frontiers in Genetics*, vol. 13, Article ID 893378, 2022.

[31] T. K. N. Dang, D. Bucur, B. Atil et al., "Look back, look around: a systematic analysis of effective predictors for new outlinks in focused Web crawling," *Knowledge-Based Systems*, vol. 260, Article ID 110126, 2023.

[32] J. Reyes-Magaña, G. Bel-Enguix, G. Sierra, and H. Gómez-Adorno, "Designing an electronic reverse dictionary based on two word association norms of English language," in *Proceedings of the electronic lexicography in the 21st century conference*, Sintra, Portugal, October 2019.

[33] R. Kaur, A. Majumdar, P. Sharma, and B. Tiple, *Analysis of Tweets with Emoticons for Sentiment Detection Using Classification Techniques*, Springer, Berlin, Germany, 2023.

[34] W. Shahid, Y. Li, D. Staples, G. Amin, S. Hakak, and A. Ghorbani, "Are you a cyborg, bot or human?—a survey on detecting fake news spreaders," *IEEE Access*, vol. 10, pp. 27069–27083, 2022.

[35] G. Singh, A. Nagpal, and V. Singh, "Text classification using improved IWO-han," *Procedia Computer Science*, vol. 218, pp. 1184–1195, 2023.

[36] S. Sharma and A. Jain, "Hybrid ensemble learning with feature selection for sentiment classification in social media," in *Research Anthology on Applying Social Networking Strategies to Classrooms and Libraries*, pp. 1183–1203, IGI Global, Hershey, PA, USA, 2023.

[37] M. Dominic, A. P. Raj, S. Francis, and X. Pakkiam, "Runtime environment for java technologies using google app engine," in *International Journal of Engineering Research Technology*, pp. 2278–0181, IJERT, Gandhinagar, Gujarat, 2013.

[38] G. Salton and C. Buckley, *Term Weighting Approaches in Automatic Text Retrieval*, Cornell University, Ithaca, NY, USA, 1987.

[39] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[40] D. M. Powers, "Applications and explanations of Zipf's law," 1998.

[41] Q. Chen, X. Wang, X. Ye, G. Durrett, and I. Dillig, "Multimodal synthesis of regular expressions," in *PLDI 2020*, Association for Computing Machinery, New York, NY, USA, 2020.

[42] M. Al-Shabi, "Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining," *IJCSNS*, vol. 20, no. 1, p. 1, 2020.

[43] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: an enhanced lexical Resource for sentiment analysis and opinion mining," in *European Language Resources Association (ELRA)*, Valletta, Malta, Eurpoe, 2010.

[44] M. Thelwall, "Heart and soul: sentiment strength detection in the social web with sentistrength," *Cyberemotions: Collective Emotions in Cyberspace 2014*, vol. 12, 2017.

[45] X. Hu, G. Li, X. Xia, D. Lo, and Z. Jin, "Deep code comment generation with hybrid lexical and syntactical information," *Empirical Software Engineering*, vol. 25, no. 3, pp. 2179–2217, 2020.

[46] N. Kansal, L. Goel, and S. Gupta, "A literature review on cross domain sentiment analysis using machine learning," *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*, vol. 52, pp. 1871–1886, 2022.

[47] C. Hutto and E. Gilbert, "VADER: a parsimonious rule-based model for sentiment analysis of social media text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, pp. 216–225, 2014.

[48] D. Iosub, D. Laniado, C. Castillo, M. Fuster Morell, and A. Kaltenbrunner, "Emotions under discussion: gender, status and communication in online collaboration," *PLoS One*, vol. 9, no. 8, Article ID e104880, 2014.

[49] X. Li, M. Xu, W. Zeng, Y. K. Tse, and H. K. Chan, "Exploring customer concerns on service quality under the COVID-19 crisis: a social media analytics study from the retail industry," *Journal of Retailing and Consumer Services*, vol. 70, Article ID 103157, 2023.

[50] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, New York, NY, USA, June 2004.

[51] F. A Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs," 2011, https://arxiv.org/abs/1103.2903.

[52] M. Nkongolo, J. P. Van Deventer, and S. M. Kasongo, "Ugransome1819: a novel dataset for anomaly detection and zero-day threats," *Information*, vol. 12, no. 10, p. 405, 2021.

[53] M. Nkongolo, D. vJP, and S. M. Kasongo, "The application of cyclostationary malware detection using boruta and PCA," in *Computer Networks and Inventive Communication TechnologiesSpringer Nature Singapore*, S. Smys, P. Lafata, R. Palanisamy, and K. A. Kamel, Eds., pp. 547–562, Springer, Singapore, 2023.

[54] M. Nkongolo, D. vJP, S. M. Kasongo, and W. Walt, "Classifying social media using deep packet inspection data," in *Inventive Communication and Computational Technologies-Springer Nature Singapore*, G. Ranganathan, X. Fernando, and Á Rocha, Eds., pp. 543–557, Springer, Singapore, 2023.

[55] M. Nkongolo, J. P. Van Deventer, S. M. Kasongo, S. R. Zahra, and J. Kipongo, "A cloud based optimization method for zero-day threats detection using genetic algorithm and ensemble learning," *Electronics*, vol. 11, no. 11, p. 1749, 2022.

[56] M. Nkongolo and M. Tokmak, "Zero-Day threats detection for critical infrastructures," Edited by A. Gerber and M. Coetzee, Eds., 2023, https://arxiv.org/abs/2306.06366.

[57] M. Nkongolo, J. P. van Deventer, and S. M. Kasongo, "Using deep packet inspection data to examine subscribers on the net-work," *Procedia Computer Science*, vol. 215, pp. 182–191, 2022.

[58] M. Nkongolo, J. P. Van Deventer, S. M. Kasongo, W. Van Der Walt, R. Kalonji, and M. Pungwe, "Network policy enforcement: an intrusion prevention approach for critical infrastructures," in *Proceedings of the 2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pp. 686–692, Coimbatore, India, December 2022.

[59] M. Nkongolo, "Game theory based artificial player for morabaraba game," in *Proceedings of the 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1210–1218, Tirunelveli, India, January 2023.

[60] M. Nkongolo, "Using ARIMA to predict the growth in the subscriber data usage," *Engage*, vol. 4, no. 1, pp. 92–120, 2023.

[61] M. A. Ayyoub, S. B. Essa, and I. Alsmadi, "Lexicon-based sentiment analysis of Arabic tweets," *International Journal of Social Network Mining*, vol. 2, no. 2, pp. 101–114, 2015.

[62] S. Šćepanović, I. Mishkovski, B. Gonçalves, T. H. Nguyen, and P. Hui, "Semantic homophily in online communication: evidence from Twitter," *Online Social Networks and Media*, vol. 2, pp. 1–18, 2017.

[63] M. R. Islam and M. F. Zibran, "A comparison of dictionary building methods for sentiment analysis in software engineering text," in *Proceedings of the 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 478-479, Toronto, Canada, November 2017.

[64] T. Chalothorn and J. Ellman, "Sentiment analysis of web forums: comparison between sentiwordnet and sentistrength," in *Proceedings of the 4th International Conference on Computer Technology and Development*, Bangkok, Thailand, November 2012.

[65] B. Stein and S. M. Zu Eissen, "Topic identification: framework and application," *Journal of Universal Computer Science*, vol. 36, pp. 353–360, 2004.

[66] R. Ferreira, L. de Souza Cabral, R. D. Lins et al., "Assessing sentence scoring techniques for extractive text summarization," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5755–5764, 2013.