Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Quick Tweet Analysis Using NLP

[1]Shivam Agarwal, [2]Anshika Jain, [3]Anshika Gupta, [4]Sejal Tyagi, [5]Dr. Mukesh Rawat

[1,2,3,4,5]Dept. of CSE, MIET, Meerut

Abstract : Twitter (currently known as X) is the most important platform and reference point where users can share news and express their opinions on events. This unique comparison of incident alerts, combined with its wealth of information and insight, highlights the importance of Twitter in incident alerts. As a result, Twitter content becomes an important tool that can provide a quick picture of any situation. But the abundance of Twitter content brings its own challenges, such as abbreviations, bad words and error messages. These nuances make extracting reliable and useful information from Twitter a difficult task, especially when short texts are used.

There is no doubt that recording events on Twitter is difficult and requires newer techniques than writing articles. Over the years, many studies have investigated different strategies for automatic Twitter content collection. This research aims to provide an overview of these commitments in the context of Twitter. Particular attention is paid to the evaluation of the collection process and the quality assessment of the state's evaluation process. The research concludes by presenting current and future research challenges in the field and provides a detailed overview.

Keywords: Automatic text summarization, Natural Language Processing, Sentiment analysis, Machine Learning, Clustering, Tokens, Latent Dirichlet Allocation (LDA).

1.Introduction

1.1 In the age of the internet and big data, people find themselves overwhelmed by too much data and information. This increase has led researchers to seek technological solutions, focusing on documentation, to reduce this data overload. Topics include compressing text, preserving important information, and reducing its size. The fact that the efforts required to write the article is time-consuming and laborious causes increased interest in the studies in the process and becomes the main driving force of learning.

Automatic writing of text works. The aim of the process is shortening the size of content while preserving important information. Its applications cover a wide range of areas, including social media, email, surveys and online research. The content of the text divides the long text into clear, accurate and coherent sentences and presents the necessary information to the reader in a few words. The two main methods of writing essays are summary and abstract writing. Inferential summarization selects important passages from the original text based on statistical data; Abstract summarization creates a summary by understanding important ideas and expressing them in plain English, including beautiful words.

Descriptive content and abstract content have their own advantages and disadvantages. Inferential summarization is good at choosing important and accurate words, but can experience inconsistencies. Abstract content, on the other hand, creates content that is clear and easy to read, but carries the risk of losing important information, especially large text.

The report focuses on microblogging services such as Twitter, where users use these services to express their views on events such as the US election. 280-character tweets posted on timelines help you stay up to date with your friends, and keyword searches keep tabs on events even by people outside your network. But the sheer size of the search results—spanning several weeks and containing millions of tweets—poses a challenge. Even if the filter is on, extracting the noise becomes a difficult

task. To solve this problem, a content-driven system that can filter the best tweets that suit the customer's needs. The bottleneck of social short text summarization is the lack of dataset methods for content evaluation. While the dataset, called the Document Understanding Conference (DUC) dataset, is freely available for data collection, authors typically create data for tweet summarization. Previous data, such as Shou L., Wang Z., Chen K., and Chen G. Continuous summarization of evolving tweet streams (2013) data, became inaccessible and hindered the evaluation process. Another document by Imran, M., Castillo, C., Lucas, J., Meier, P., and Vieweg, S. (2014) focuses on disaster response during the Joplin hurricane and contains more than 230,000 tweets. However, the lack of valid data makes it difficult to evaluate these data, indicating an urgent need for comprehensive and valid sources of good short writing in the field of communication.

1.2 Background History

Over the span of 2008 to 2019, text summarization research has encountered various challenges, with notable attempts made to address these issues. Extraction emerged as the most prevalent challenge during this period, involving the retrieval of data from diverse sources, structured or unstructured, for summary creation. The predominant technique employed to tackle extraction challenges was machine learning, aligning with the prevailing approach across the field during this timeframe.

On the contrary, issues like word frequency, sentiment analysis, noise, and sentence ranking have received relatively less focus over the past decade. Word frequency and sentence ranking, perceived as less complex, were believed to be addressable through statistical approaches. However, sentiment analysis, crucial for understanding textual subtleties, has remained relatively unexplored despite being a potentially beneficial aspect of text summarization.

A challenging yet less explored aspect involved devising new or optimal feature combinations to significantly enhance summary quality. This necessitated an in-depth exploration to identify features that would substantially improve evaluation metrics for generating high-quality summaries.

Semantic comprehension emerged as a persistent challenge, particularly in deciphering contextual meaning within sentences. Ambiguity arising from multi-document scenarios, synonymous terms, or ambiguous phrases necessitated attention to ensure that the summary reflects the original document's intended meaning. Various methodologies like Abstract Meaning Representation (AMR), Semantic Link Network, Semantic Role Labelling (SRL), Maximal Marginal Relevance (MMR), Latent Semantic Analysis (LSA), and Non-Negative Matrix Factorization (NMF) were attempted to address semantic issues, but unresolved challenges persist, as indicated by multiple researchers in their conclusions or future work sections.

Addressing redundancy and similarity issues entailed identifying and removing duplicate or analogous sentences without altering the essence of the document. Techniques such as MMR, Maximum Relevance Minimum Redundancy (MRMR), text-rank, lex-rank, IntraLink, Bernoulli mode, similarity co-sin, Shark Smell Optimization, and others were employed to address these challenges, aiming to refine summary results while maintaining document integrity.

In summary, the era from 2008 to 2019 witnessed considerable progress in text summarization, yet challenges persist, notably in semantic comprehension, highlighting the need for continued research and innovation in this evolving domain.

Various approaches have been suggested to tackle the semantic difficulties encountered in text summarization, such as AMR (Bhargava et al., 2016) [1], Semantic Link Network (Sun and Zhuge, 2018) [2], SRL, MMR (Khan et al., 2015a)[3]. Despite these endeavors, numerous researchers have noted in their conclusions or future work sections that the semantic issue persists and remains unresolved (Patel et al., 2019; Binwahlan et al., 2009a; Khan et al., 2015a; Wu et al., 2017; Kacprzyk et al., 2008; Patel and Chhinkaniwala, 2018; Guo et al., 2019; Chen et al., 2015; Liu et al., 2015a,b)[5,7,8,9].This shared recognition emphasizes the ongoing difficulty in achieving semantic understanding in contemporary text summarization research.

1.3 Supported Techonologies

This research employs a sophisticated technological framework, leveraging a suite of Python libraries to conduct a comprehensive analysis of Twitter data. The initial phase involves data retrieval from the Twitter API using Twython, a Python wrapper for the Twitter API, and advertools, a library facilitating advanced analysis of online advertising data. Authentication parameters, including app keys and tokens, are configured for secure access to the Twitter API. The acquired tweets are subsequently stored in a structured format using the Pandas library, allowing for efficient data manipulation and analysis.

Data preprocessing is a critical step in ensuring the quality of the collected tweets. The script incorporates regular expressions (re) and the spaCy library for natural language processing to cleanse the tweet content. Unicode characters, newlines, and rawstring characters are systematically removed, contributing to the enhancement of data quality and consistency. To derive meaningful insights from the tweet corpus, a diverse set of features is extracted. The analysis encompasses the identification of mentions, hashtags, and title mentions (such as Mr., Mrs., Dr., and Miss) within the tweet text. Furthermore, the script computes metrics like word count, character count, average word length, stopwords frequency, and part-of-speech tag occurrences. This multifaceted feature extraction process enriches the dataset, enabling a nuanced understanding of the characteristics and linguistic nuances present in the Twitter data.

The exploration of latent topics within the tweet content is facilitated by Gensim, a Python library for topic modeling and document similarity analysis. The tweets are tokenized and lemmatized, and a bag of words (BoW) representation is created for subsequent application of Latent Dirichlet Allocation (LDA). The LDA model identifies underlying topics within the tweet corpus, contributing to a deeper comprehension of the thematic composition of the collected tweets.

The integration of NLTK (Natural Language Toolkit) further enhances the text processing capabilities of the framework. Stemming and lemmatization, crucial techniques in text normalization, are employed to reduce words to their root forms, thereby improving the accuracy and efficiency of subsequent analyses.

This technological amalgamation provides a robust foundation for a holistic analysis of Twitter data, combining the strengths of data retrieval, preprocessing, feature extraction, and text analysis. The resulting insights contribute to a nuanced understanding of the dynamics and content characteristics within the examined Twitter dataset. The synergistic integration of these technologies underscores the versatility and effectiveness of the implemented framework, positioning it as a comprehensive solution for researchers seeking to extract meaningful information from the vast and dynamic landscape of social media data.

2. Proposed Work Plan

This research presents a comprehensive methodologyfor Twitter data analysis, encompassing data retrieval, preprocessing, feature extraction, and machine learning. The integrated framework reveals nuanced insights into content characteristics and prevalent topics.

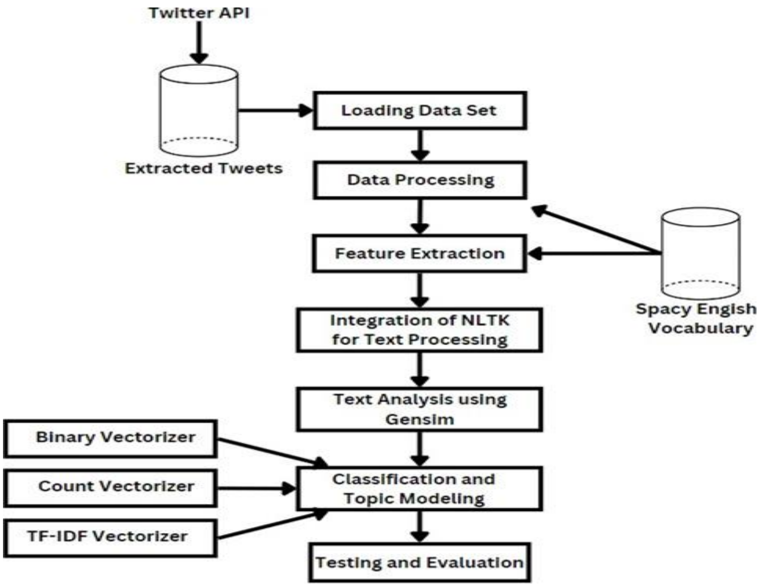


Fig 1. Flow chart of proposed methodology

Data Retrieval form Twitter API: In the initial phase of our research, we establish a connection with the Twitter API using Twython and advertools. Through this interface, we curate tweets based on a specific query, focusing on the topic with the help of keyword- based searching. The count of real time tweets fetched from the Twitter API is set beforehand with the help of Twython. Advertools is used to convert the tweets into a CSV file for easier processing. The authentication parameters, including app keys and tokens, are accurately configured to ensure secure access to the API. This stage sets the groundwork for obtaining a targeted and relevant dataset for subsequent analyses, enabling us to capture real-time insights from the dynamic Twitter platform.

Data Pre-Processing: After the acquisition of tweets, a pivotal stage involves meticulous data preprocessing for heightened quality. Regular expressions serve as a versatile tool for pattern matching, enabling the systematic removal of undesirable elements such as unicode characters, newlines, and rawstring characters. Simultaneously, spaCy enhances this cleansing process by providing a linguistic lens to understand the intricate structures within the text. SpaCy offers an advanced English vocabulary model, facilitating not only the removal of artifacts but also intricate linguistic analyses. The synergy of regular expressions and spaCy not only enhances the accuracy of data refinement but also lays a robust foundation for subsequent analyses, elevating our capability to derive meaningful insights from the diverse landscape of Twitter data.

Feature Extraction: Advancing to the feature extraction stage, our attention converges on a comprehensive set of features derived from the pre- processed tweet corpus. In addition to conventional metrics such as mentions and hashtags, spaCy's integration continues to play a pivotal role in extracting intricate linguistic features. The English vocabulary model facilitates not only accurate tokenization but also the extraction of part-of-speech tag occurrences, contributing to a more nuanced understanding of syntactic structures within tweets. Regular expressions remain instrumental in capturing specific patterns, enhancing the robustness of feature extraction. The amalgamation of these approaches yields a diverse feature set, encompassing title mentions, word count, character count, average word length, stopwords frequency, and linguistic nuances. Each feature encapsulates a unique aspect of the tweet content, contributing to the development of a rich and multidimensional dataset.

Integration of NLTK for Text Processing: To fortify our text processing capabilities, we seamlessly incorporate the Natural Language Toolkit (NLTK). NLTK facilitates a spectrum of advanced techniques, including stemming and lemmatization, which are pivotal in refining the textual data. Stemming involves reducing words to their root forms, ensuring a standardized and concise representation. Simultaneously, lemmatization refines words to their base or dictionary forms, capturing nuanced meanings. The NLTK integration extends beyond these techniques, encompassing functionalities for part-of-speech tagging, named entity recognition, and sentiment analysis. This comprehensive suite of tools empowers our analysis pipeline with a deeper understanding of linguistic nuances, enabling us to uncover semantic relationships and contextual intricacies within the Twitter data.

Text Analysis using Gensim: Gensim takes center stage in our advanced text analysis, demonstrating its prowess in linguistic processing. This includes intricate tasks such as tokenization, where text is broken down into meaningful units, and lemmatization, which refines words to their base forms, ensuring a more refined understanding of semantic relationships. The creation of a Bag of Words (BoW) representation through Gensim not only captures word frequency but also lays the groundwork for more sophisticated analyses.

One essential technology utilized in this phase involves Latent Dirichlet Allocation (LDA), a robust probabilistic model used for topic modeling. LDA surpasses conventional content extraction methods by uncovering hidden themes within the tweet corpus. This model operates under the assumption that each document comprises a blend of topics, and the presence of each word can be attributed to one of these topics within the document. Through the application of LDA, we probe into the underlying thematic structure of the dataset, unveiling concealed patterns and prevalent subjects. This methodological decision extends our analysis beyond surface-level observations, providing a nuanced comprehension of the diverse topics circulating in the Twitter data. The incorporation of Gensim and LDA underscores our dedication to harnessing state-of- the-art technologies for a more comprehensive exploration of content dynamics within the realm of social media.

Classification and Topic Modelling: Moving beyond data preparation, our focus shifts to implementing machine learning models with a strategic integration of advanced vectorization techniques. This transformative phase encompasses binary vectorization, count vectorization, and TF-IDF vectorization, each playing a distinct role in converting tweet data into feature vectors. Binary vectorization simplifies representation, count vectorization provides a detailed view of word distribution, and TF-IDF captures word significance.

Binary vectorization involves representing each tweet as a binary vector, indicating the presence or absence of specific words. This approach simplifies the representation, capturing the essence of word occurrence without considering frequency. The formula for binary vectorization can be expressed as:

Binary Vector = $[n_1, n_2, \ldots, n_N]$

Here, each element in the vector n_i corresponds to the presence (1) or absence (0) of a specific word.

Count vectorization operates by enumerating the frequency of each word in the tweet corpus. The formula for count vectorization is:

Count Vector = $[f_1, f_2, \ldots, f_N]$

In this formula, each element f_i represents the count of a specific word in the tweet corpus. This technique results in a count-based representation, providing a more detailed view of word distribution across tweets.

TF-IDF Vectorization combines Term Frequency (TF) and Inverse Document Frequency (IDF) to assign weights to words based on their importance in individual tweets and the entire corpus. The formula for TF-IDF vectorization is:

Term Frequency (TF) refers to the count of occurrences of a term within a specific document.

$$\mathrm{TF}(t, d) = \text{Number of times term } t \text{ appears in document } d$$

Inverse Document Frequency (IDF) evaluates the significance of a term within a document in comparison to its occurrence frequency across all documents.

$$\mathrm{IDF}(t, D) = \log\left(\frac{\text{Total number of documents in corpus } D}{\text{Number of documents containing term } t{+}1}\right) + 1$$

The "+1" in the denominator is to avoid division by zero.

$$\text{TF-IDF Vector} = [tf_1 \times idf_1, tf_2 \times idf_2, \ldots, tf_n \times idf_n]$$

In this equation, 'tf' signifies the term frequency of a word within a particular tweet, while 'idf' stands for the inverse document frequency, reflecting the significance of a word across the entire corpus. This approach effectively gauges the importance of words in differentiating one tweet from another.

With the vectorized data, our focus shifts to generating Multinomial Naïve Bayes classification models tailored for each representation - Binary Vectorizer, Count Vectorizer, and TF-IDF Vectorizer. These models stand as potent tools for classifying and categorizing tweets based on their content. The classification process structures the dataset, providing an organized and interpretable framework for understanding the diverse content within the Twitter dataset.

The Multinomial Naïve Bayes algorithm, chosen for its efficiency in handling sparse data, forms the backbone of our classification strategy. Its formula for text classification tasks involves calculating the probability as:

$$P(\text{Class}|\text{Document}) \propto P(\text{Class}) \times P(\text{Term}_1|\text{Class}) \times \ldots \times P(\text{Term}_n|\text{Class})$$

Here, P(Class) represents the prior probability of a tweet belonging to a specific class, and P(Term$_i$ |Class) denotes the likelihood of observing a specific term given the class. The Naïve Bayes algorithm assumes independence between terms, simplifying the calculation.

For the actual classification, the calculated probabilities are compared, and the class with the highest probability is assigned to the tweet. The vectors play a crucial role in this process, as the probabilities are calculated based on the presence or absence of specific terms in the vectorized representations. The choice of vectorization technique influences how these probabilities are computed, impacting the overall classification outcome.

This algorithm is well-suited for text classification tasks, making it an ideal choice for categorizing tweets. By applying these models to different vectorized representations, we gain insights into how the choice of vectorization technique influences the classification outcomes. This dual focus on vectorization and classification ensures a comprehensive exploration of the intricate patterns and thematic structures present in the dynamic Twitter data.

Testing and Evaluation: In the final evaluative phase, the classification models undergo rigorous testing against a designated set of test documents. Evaluation metrics, including accuracy, precision, recall, and F1 score, are computed. This comprehensive testing protocol provides insights into the models' performance, their robustness, and their capacity to generalize to unseen data, thereby validating the effectiveness of our implemented models.

3. Experimental Result Analysis

3.1 Description of data set used

In the exploration of our dataset, we strategically employed the en_core_web_sm model from the Spacy library for its notable efficiency and speed in analysing linguistic components. Despite its compact size, this model demonstrated remarkable proficiency in tasks such as tokenization, part-of-speech tagging, and entity recognition, playing a pivotal role in swiftly extracting meaningful insights from our data. Simultaneously, our dataset curation involved leveraging NLP techniques, utilizing libraries like NLTK and Gensim, and incorporating a Snowball Stemmer for English to enhance linguistic uniformity. Additional configurations for word lemmatization and stemming were implemented, and the dataset was augmented with the WordNet lexical database for linguistic richness. This thoughtful consideration of lexical resources strengthens the robustness of our tweet analysis model's foundation. The testing dataset, comprised of tweets obtained through the Twitter API, has been enriched with preprocessing, incorporating various key information columns and additional features like counts of mentions, hashtags, word and character counts, average word length, stopwords count, part-of-speech tagging, and named entity recognition. This multi- dimensional dataset serves as a solid foundation for detailed analyses, covering temporal trends, social interactions, linguistic characteristics, and syntactic features. The integration of the Spacy English model, regular expressions, NLTK, Gensim and WordNet lexical database ensures a comprehensive understanding of both content and structure, laying the groundwork for subsequent research investigations.

3.2 Results

In this section, the results obtained with different sources, from the basic process to the advanced level used in the project, are carefully examined. It emphasizes the importance of testing and its important role in developing an accurate model.

Evaluating data classification models involves various tests to evaluate their performance using specific metrics. The formula used to show the accuracy of classification is as follows: Accuracy = $(1 - \mu / N) * 100\%$

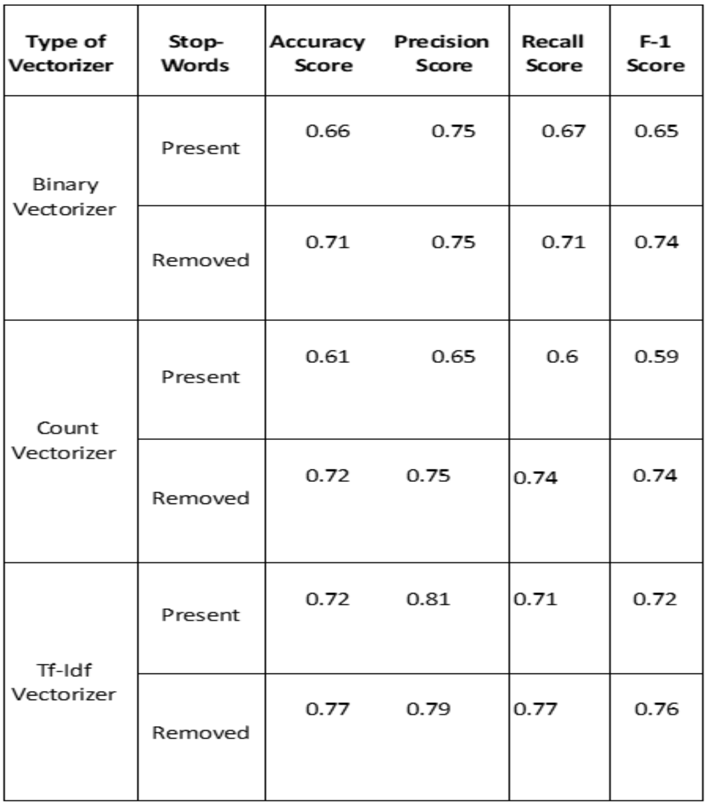
Here "μ" represents data that is not classified in the test with the value "N". Each result corresponds to one run of the classifier. In addition, the model is carefully evaluated against additional criteria such as precision and recall.

Precision (P) is the ratio of true quality (Tp) to the sum of negative (Fp) and positive (Tp).

Recall (R) is defined as the ratio of the positive (Tp) to the sum of the number of negatives (Fn) and positives (Tp).

These measurements are related to each other through the **(F1) score**, which represents the balance between true and its opposite.

All measurements are normalized on a scale from 0 to 1; where 1 indicates good performance and 0 indicates poor performance.



Type of Vectorizer	Stop-Words	Accuracy Score	Precision Score	Recall Score	F-1 Score
Binary Vectorizer	Present	0.66	0.75	0.67	0.65
	Removed	0.71	0.75	0.71	0.74
Count Vectorizer	Present	0.61	0.65	0.6	0.59
	Removed	0.72	0.75	0.74	0.74
Tf-Idf Vectorizer	Present	0.72	0.81	0.71	0.72
	Removed	0.77	0.79	0.77	0.76

Table 1. Comparing Results

Compared to count vectorizers and binary vectorizers, the TF-IDF vectorizer performs better, outperforming the count vectorizer by 4% and the binary vectorizer by 6%. It also achieved a higher accuracy score of 0.81 compared to the 0.75 accuracy score of the count vectorizer and binary vectorizer. In terms of recall, the TF-IDF vectorizer performs well with a score of 0.77, beating the count vectorizer's 0.74 and the binary vectorizer's 0.71. Therefore, due to higher sensitivity and recovery, the TF-IDF vectorizer achieved an F-1 score of 0.76; This is better than the binary vectorizer's
0.74 and the computational vectorizer's 0.74.

TF-IDF vectorizer has the advantage that it differs from count vectorizers that focus only on frequency expressions or binary vectorizers that only record content. This approach ensures that the TF-IDF vectorizer accurately represents the data. Interestingly, when stop words are not removed, the binary vectorizer outperforms the count vectorizer, achieving 66% accuracy compared to the count vectorizer's 61% accuracy. This is because most files contain many stop words. While frequency can cause stop words to become unimportant in the vectorizer calculation, the binary vectorizer treats stop words as words in the document, preventing them from subsuming other important words.

Even without removing stop words, TF-IDF vectorizer performs well as it includes both time frequency and IDF value. The IDF value reduces the effect of high frequency of stop words, maintaining the balance of the TF-IDF score. However, the performance of the TF-IDF vectorizer decreases due to relationships between groups such as religion and politics, resulting in the emergence of information that can affect both groups.

In summary, the superiority of the TF-IDF vectorizer comes from its subtle weighting method, which allows it to represent data more accurately, despite the difficulties it faces in inherently relevant groups such as religion and politics.

4. Conclusion

Our analysis indicates that while both binary and count vectorizers demonstrate strong performance, the TF- IDF vectorizer surpasses them in terms of data fit and classification outcomes. Despite the count vectorizer's commendable performance, it outperforms the binary vectorizer only when stop words are eliminated.

From a group perspective, most groups achieved classification accuracy, recovery rate, and F-1 scores above 70%. More importantly, the information is fragmented into religious, political, and other important aspects that have been shown to be less effective. Middle Eastern Law has the highest sensitivity, recall and F-1 scores of 0.90.

It is useful to use the p-value test to evaluate the distribution of the null hypothesis. A P value of less than 0.05 indicates significant evidence against the null hypothesis and confirms that the alternative hypothesis is accepted.

Our research only uses the Naive Bayes classification method, which is based on our main goal of searching for different information. As a suggestion for future research, we encourage researchers and students to explore and analyze various models using other machine learning algorithms such as SVM, neural networks, maximum search, and decision trees.

Additionally, while our report doesn't include details about the hybrid vectorizer we experimented with, it's worth noting that although it doesn't match the efficiency of the TF-IDF vectorizer, it operates significantly faster than our mentioned representative vectorizer. Not requiring constant retention in memory saves space, which could be advantageous. For researchers dealing with distributed data instantiation, exploring hybrid vectorizers could be beneficial.

References

[1] Bhargava, R., Sharma, Y., & Sharma, G. (2016). *ATSSI: Abstractive Text Summarization Using Sentiment Infusion. Procedia Computer Science, 89, 404–411.*

[2] Sun, X., & Zhuge, H. (2018). *Summarization of Scientific Paper Through Reinforcement Ranking on Semantic Link Network. IEEE Access, 6, 40611–40625.*

[3] Khan, A., Salim, N., & Jaya Kumar, Y. (2015). *A framework for multi-document abstractive summarization based on semantic role labelling. Applied Soft Computing, 30, 737–747.*

[4] Patel, A., & Jain, S. (2019). *Present and future of semantic web technologies: a research statement. International Journal of Computers and Applications, 1–10.*

[5] Patel, D., & Chhinkaniwala, H. (2018). Fuzzy logic- based single document summarisation with improved sentence scoring technique. International Journal of Knowledge Engineering and Data Mining, 5(1/2), 125.

[6] Gao, S., Chen, X., Li, P., Ren, Z., Bing, L., Zhao, D.,& Yan, R. (2019). Abstractive Text Summarization by Incorporating Reader Comments. Proceedings of the AAAI Conference on Artificial Intelligence, 33, 6399– 6406.

[7] K. -Y. Chen et al., "Extractive Broadcast News Summarization Leveraging Recurrent Neural Network Language Modeling Techniques," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 1322-1334, Aug. 2015.

[8] Kacprzyk, J., Wilbik, A., & Zadrożny, S. (2008). Linguistic summarization of time series using a fuzzy quantifier driven aggregation. Fuzzy Sets and Systems, 159(12), 1485–1499.

[9] Binwahlan, M. S., Salim, N., & Suanmali, L. (2010). *Fuzzy swarm diversity hybrid model for text summarization. Information Processing & Management, 46(5), 571–588.*