Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Analyzing Hybrid C4.5 Algorithm for Sentiment Extraction over Lexical and Semantic Interpretation

Kanchan Naithani*

*Corresponding Author, Research Scholar, Department of Computer Science and Engineering, Hemvati Nandan Bahuguna Garhwal University (A Central University), Srinagar Garhwal, Uttarakhand, India. E-mail: kanchannaithani696@gmail.com

Y. P. Raiwani

Professor, Head, Department of Computer Science and Engineering, Hemvati Nandan Bahuguna Garhwal University (A Central University), Srinagar Garhwal, Uttarakhand, India. E-mail: yp_raiwani@yahoo.com

Intyaz Alam

School of Computer and Systems Sciences, JNU New Delhi, India. E-mail: intyazcsejnu@gmail.com

Mohammad Aknan

Department of Computer Science and Engineering, Gaya College of Engineering, Gaya. Email: aknan.cse@gmail.com

Abstract

Internet-based social channels have turned into an important information repository for many people to get an idea about current trends and events happening around the world. As a result of Abundance of raw information on these social media platforms, it has become a crucial platform for businesses and individuals to make decisions based on social media analytics. The ever-expanding volume of online data available on the global network necessitates the use of specialized techniques and methods to effectively analyse and utilize this vast amount of information. This study's objective is to comprehend the textual information at the Lexical and Semantic level and to extract sentiments from this information in the most accurate way possible. To achieve this, the paper proposes to cluster semantically related words by evaluating their lexical similarity with respect to feature and sequence vectors. The proposed method utilizes Natural Language Processing, semantic and lexical clustering and hybrid C4.5

algorithm to extract six subcategories of emotions over three classes of sentiments based on word-based analysis of text. The proposed approach has yielded superior results with seven existing approaches in terms of parametric values, with an accuracy of 0.96, precision of 0.92, sensitivity of 0.94, and an f1-score of 0.92.

Keywords: Hybrid C4.5, Lexical Analysis, Machine Learning, Semantic Analysis, Sentiment Analysis, Social Media Data.

Journal of Information Technology Management, 2023, Vol. 15, Special Issue, pp. 57- 79
Published by University of Tehran, Faculty of Management
doi: https://doi.org/ 10.22059/jitm.2023.95246
Article Type: Research Paper

Received: July 06, 2023
Received in revised form: August 24, 2023
Accepted: November 09, 2023
Published online: December 24, 2023

Introduction

The human language is a distinct and impeccable tool of communication and cognition, characterized by phonetic, lexeme-based, and syntax-based features that allow individuals to express themselves (Shahi, Sitaula & Paudel, 2022). In contemporary times, online communication has turn out to be an integral part of people's lives due to advancements in web technologies. The text-based information generated by humans during informal or formal conversations is typically unstructured and highly noisy (Heras-Pedrosa et al., 2020). Nevertheless, it corresponds to an explicit language, adhering to specific syntax and semantics, thereby providing valuable information. Thus, providing a challenging objective that involves understanding the associations between the primary elements of the given textual information (Garcia & Berton, 2021).

The primary motivation behind this research work is derived from Jalil's (Jalil, Javed, et al., 2022). XGBoost - “eXtreme gradient boosting” - a tree-based model that uses a decision making algorithm that classifies data using a tree structure along with Garcia’s (Garcia & Berton, 2021). utilization of GSDMM –“Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture” for Twitter content topic recognition and sentiment analysis relating to COVID-19.While modern approaches have made significant strides throughout the previous few years, the rise of advanced deep learning methods and large-scale language models, there are still many open questions and limitations to be addressed (Snow et al., 2006). Specifically, these approaches often struggle to capture the nuanced relationships between words and their meanings, as well as the complex interplay between different concepts and ideas within a given text.

The foremost challenge lies in comprehending the lexical and semantic relationships among the fundamental constituents of the textual data (Hearst Marti, 1992). In a “multiaspect-specific position attention bidirectional long short-term memory" (MAPA BiLSTM) - BERTmodel (Wankhade, et al., 2023) proposed issues like the specific position context and the discrete aspect of an opinionated sentence are addressed. They showed how individual interference by multiple aspects in the same sentence can be evaluated while determining the current-aspect-attention-vector. Also, using a convolutional neural network (CNN) model by (Qorich & Ouazzani, 2023) made an effort to classify sentiments for text-based reviews as negative or positive. They made an analysis that involves the comparison over numerous models that depict word embedding. The outcomes of their study demonstrate the significance of incorporating stop words in sentiment analysis tasks. Specifically, they have shown that removing stop words may lead to an erroneous sentiment prediction, whereas including them can enhance the accuracy of the prediction by 2%, as opposed to the CNN model that disregarded them.

One important issue is the need to account for context and variability in language use. Words can have different implications in different circumstances and associations between words can vary depending on the domain, genre, and style of a given text. For instance, the association between the words "apple" and "computer" might differ depending on whether one is discussing fruit or technology (Bollegala et al., 2007). Another challenge is the need to capture the more abstract and nuanced aspects of language use, such as idiomatic expressions, metaphors, and humor. These can be difficult to model using traditional rule-based or statistical approaches, as they often rely on more subtle and context-dependent cues (Naseem, et al., 2021). As a result, there is a continued need for further research and development in this area to improve our ability to accurately model and understand language at a deeper level (Verma, 2017). With continued research and development, it is likely to see significant progress in our ability to model and understand the complex associations between words and sentences.

Another key challenge is the need to accurately capture the nuances of language use and the complexity of semantic and lexical associations between words. Traditional approaches to sentiment analysis often rely on simple keyword matching or rule-based methods, which can be limited in their ability to capture the full range of sentiments expressed in a text. The proposed work aims to address the above-mentioned issues by leveraging text similarity and clustering techniques to interpret semantic and lexical meaning at the word level. Specifically, the approach combines dictionary-based features with sequence vectors for text analysis, which are then utilized to classify sentiment for analyzing the corpus. The approach of merging dictionary-based features with sequence vectors for text analysis enables the capture of contextual and semantic relationships among words in a given text, which in turn can be leveraged to enhance the accuracy of sentiment classification. To achieve this goal, four major phases have been defined in the current paper after the basic introduction to the topic, as shown in Figure 1.

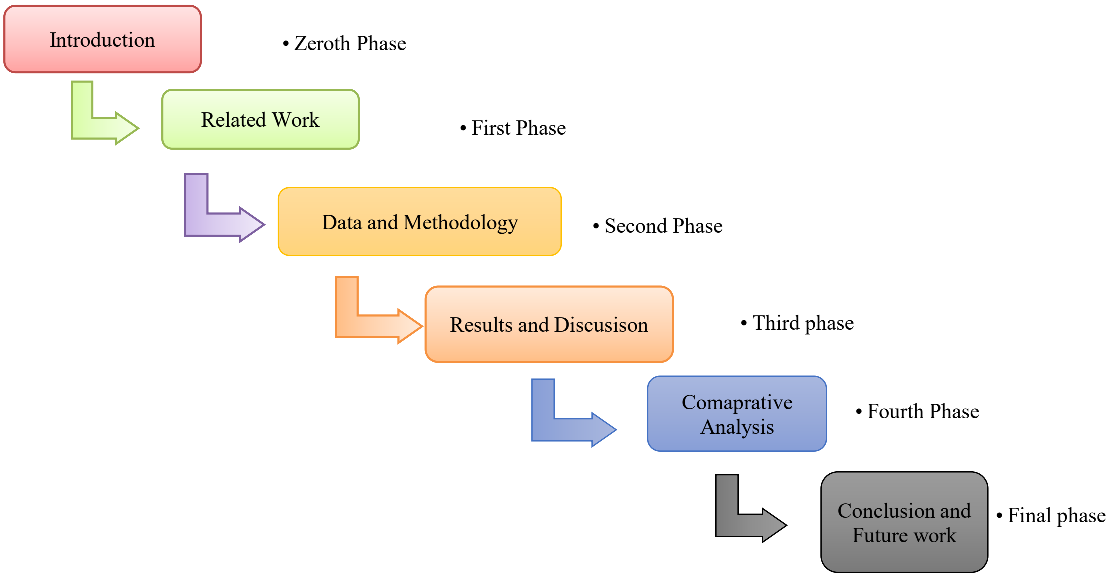


Figure 1. Workflow

In the first phase, related work is reviewed to observe the work that has been accomplished using multiple approaches by various academics in the field of Sentiment Analysis, Lexical Analysis, and Semantic Analysis. It also provides fundamental knowledge about Lexical and Semantic Interpretations, which are essential for understanding the proposed methodology.

In the second phase, the methodology with a Hybrid C.45 Algorithm for feature and space vectors to achieve Sentiment Analysis is discussed. This involves using text similarity and clustering techniques at initial stage to interpret the semantic and lexical meaning of text at the word level, which is then used to categorize the sentiment of the text.

Then, the results based on Text Similarity and clustering algorithm along with hybrid C4.5 algorithm show analysis of “The COVID-19” dataset. The outcomes are categorized into 6 emotions, viz. “Happy”, “Sad”, “Disgust”, “Fear”, “Anger”, and "Surprise”, along with a “neutral” variant from the three classes of sentiment.

Subsequently, the suggested method is evaluated against previously established approaches, and their comparative results are assessed based on metrics such as accuracy, precision, sensitivity, and f-score, before generating the final conclusion.

Overall, the proposed methodology contributes in providing a comprehensive and effective framework for sentiment analysis that leverages the strengths of text similarity and clustering techniques to interpret the semantic and lexical meaning of text at the word level, ultimately leading to more accurate and insightful results.

Literature Review

While various approaches have demonstrated multiple novel methods for establishing associations between words and sentences, such as through the use of neural networks and machine learning algorithms, they remain deficient in their ability to provide comprehensive understanding of lexical and semantic associations. The challenge of modelling lexical and semantic associations in language is a complex and cutting-edge focus of inquiry in the domain of NLP and computational linguistics (Caraballo, 1999). The process of understanding the semantics behind data extracted from social media platforms involves the use of various algorithms such as program comprehension, control and data flow analysis, program slicing, and pattern matching. To address the challenges associated with these approaches and to extract specific semantic spaces, researchers have proposed using local patterns to extract hyponymy (Caraballo, 1999; Snow et al., 2006), synonymy (Bollegala, 2007) or meronymy relationships (Naseem et al., 2021). One approach involves clustering noun vectors in a bottom-up fashion to form a nouns' hierarchy, as proposed in (Sharon Caraballo, 1999). Other studies have proposed using supervised learning to automatically obtain local patterns, as demonstrated by Naseem and Razzak and their colleagues in (Naseem et al., 2021). They studied manual patterns and proposed an approach to automatically obtain appropriate local patterns

Various tools for Machine Learning and Text Mining were used by (Verma, 2017) to elucidate the lexemes used in ten Holy Scriptures. - “the Holy Bible”, “the Bhagwad Gita”, “the Guru Granth Sahib”, “the Agama”, “the Quran”, “the Dhammapada”, the “Tao TeChing”, “the Rig Veda”, “the Sarbachan” and “the Torah.” The work utilized NLP techniques to observe the total words, nouns and verbs as lexical tokens.

People around the world are now indulged with various forums, blogs and popular social networking platforms Facebook, Instagram, WhatsApp, YouTube, Twitter, etc., to communicate their thoughts and ideas with other people (Scott & Dominic, 2003). One of the toughest tasks while extracting semantic relation from these sources is to face ambiguous patterns, only few of which represent the correct relations (Pang, Lee & Vaithyanathan, 2002; Naithani & Raiwani, 2022). Social media platforms have turned out to be the most active platforms for communication. As a result, massive amount of data is produced, for which sentiments were extracted to examine this big data at deeper levels (Mahmoud et al., 2016).

While existing approaches to natural language processing, including distributional approaches and knowledge mining solutions, have shown promise in a number of applications, they still face several critical challenges (Rana & Singh, 2016). One of the key issues is the need to train these models on copious amount of data with the motive to achieve accurate and reliable results (Jalil et al., 2021). In the case of knowledge mining solutions, these models often require extensive supervision in order to learn the underlying relationships between words and their meanings. This can limit the applicability of these models to individual languages and specific semantic relations, as it may be difficult to generalize these models to other domains or contexts (Wang et. al., 2020). In contrast, distributional approaches rely on statistical patterns in large datasets to learn the relationships amongst words and their meanings, and thus do not require the same level of supervision (Lyu et. al., 2020). However, they can also face limitations in their ability to capture more nuanced or abstract aspects of language use, such as metaphors or idiomatic expressions (Lyu et. al., 2020; Gencoglu, 2020).

Despite these challenges, there is much optimism about the potential of modern language models to improve our understanding of language at a deeper level for extracting sentiments (Babu & Eswari, 2020). There is ongoing research and advances in computational linguistics aimed at improving the accuracy and applicability of these models (Nagamanjula & Pethalakshmi, 2020). This includes the development of more sophisticated machine learning algorithms along with the integration of additional sources of knowledge, such as external knowledge graphs or ontologies (Naithani & Raiwani, 2022). The effectiveness of a sentiment analysis model is heavily influenced by multiple factors, including the extraction of relevant sentiment words, accurate sentiment classification, dataset quality, data cleansing, and more (Nabiul & Dhruba, 2021). Overall, the continued progress in this field is likely to have significant implications for a Broad spectrum of use cases, encompassing natural language understanding, information retrieval, and machine translation, among others (Nabiul & Dhruba, 2021; Khalid, et al., 2021).

Scott and Dominic explained how Lexical and semantic interpretation aims to extract the precise meaning or dictionary meaning from the textual content (Scott & Dominic, 2003). The building blocks for representing the meaning of the words are Entities (that denotes the individuals such as a particular person, location etc. such as Uttarakhand, Rohit), Concepts (that refers to the broad category of entities, such as persons, cities, etc.), Relations (that conveys the connection between entities and conceptual categories, as in "Rohit's categorization as a person.") and Predicates (that explain the patterns of verb usage such as semantic roles and case grammar).Lexical and Semantic Interpretations uses a simple grammar and lexicon to demonstrate the computation of logical forms with the help of features using predicate argument structure for parsing (Mahmoud et al., 2016; Rana & Singh, 2016). A “SEM-feature” is used as a chief extension for the description of a word in a dictionary or other lexicon and a rule or pattern that governs how words are combined to form phrases and sentences in a particular language respectively. A sample rule in the grammar is shown below:

(S_**SEM**_(semvp semnp)) → (NP_**SEM** semnp)_(VP_**SEM** semvp)

In view of the above given rule, the NP sub constituent with SEM (NAME c1 “Doctor” and the VP sub-constituent with SEM (ʎ a (SEES1 e8 a (Name p1 “patient”))) (Gencoglu, 2020; Babu & Eswari, 2020). Following figure 2 explains Parse Tree showing the SEM Features.

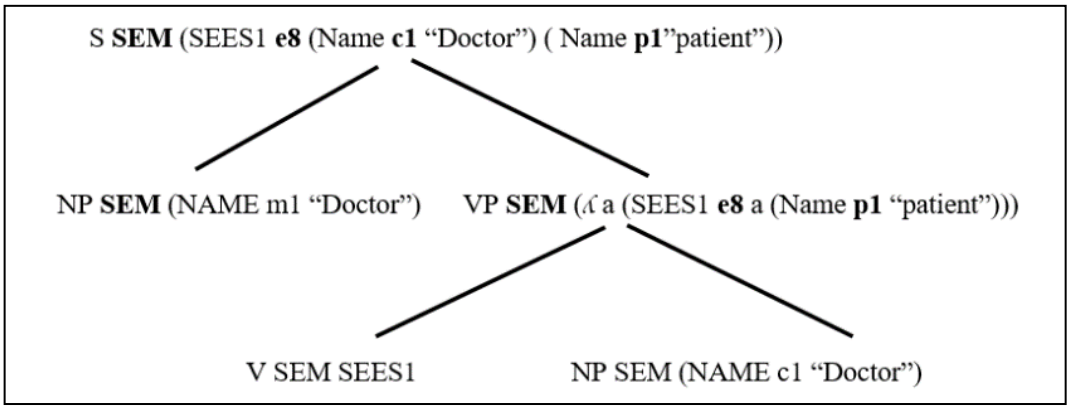

S **SEM** (SEES1 **e8** (Name **c1** “Doctor”) (Name **p1**”patient”))
NP **SEM** (NAME m1 “Doctor”)
VP **SEM** (ʎ a (SEES1 **e8** a (Name **p1** “patient”)))
V SEM SEES1
NP SEM (NAME c1 “Doctor”)

Figure 2. A Parse Tree showing the SEM Features

Lexical Semantics stands as the leading step for comprehending semantic context, where the examination of word meanings in isolation is studied (Qorich & Ouazzani, 2023). As explained by Mrtdaa &Salma, it comprises of words, sub-words, compound words, prefixes, suffices and phrases as well. All these components are as called lexical items. In simpler words, lexical and semantic interpretation is the association of lexical items, sentence meaning and its syntax (Hearst Marti, 1992). Figure 3, provides the primary steps of lexical semantics –

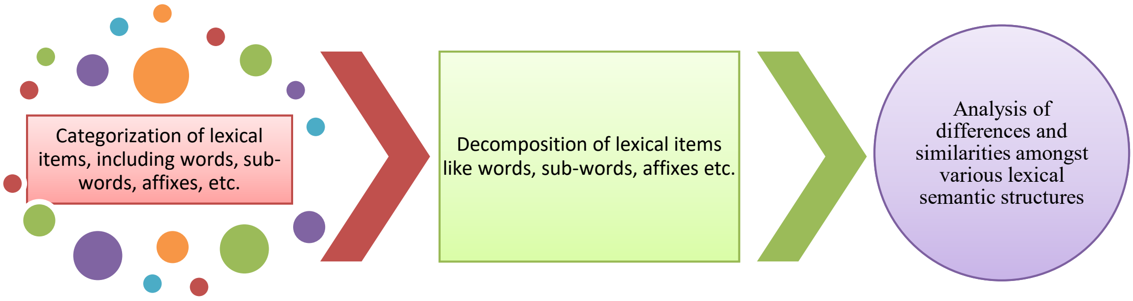

Categorization of lexical items, including words, sub-words, affixes, etc.
Decomposition of lexical items like words, sub-words, affixes etc.
Analysis of differences and similarities amongst various lexical semantic structures

Figure 3. Steps involved in lexical semantics (Hearst Marti, 1992; Naseem, et al., 2021; Naithani & Raiwani, 2022; Zhang et al., 2023)

Wankhade et. al., (2023) while working on an attention mechanism that is aware of multiple aspects, designed for sentiment analysis at the aspect level, explained the dynamic aspect of Lexical semantics in the field of NLP and NLU, as it deals with the meaning of individual words and how they combine to form larger units of meaning such as phrases, clauses, and sentences. The study of lexical semantics involves examining the relationships between words and their meanings, including how words are related to one another in terms of synonyms, antonyms, hyponyms, and hypernyms (Qorich & Ouazzani, 2023).

One important area of lexical semantics was introduced by Bollegala, Matsuo and Ishizuka in their recent study of polysemy, which refers to cases where a single word has multiple meanings (Bollegala et al., 2007). For example, the word "bank" can denote a financial body, the river-side, or the act of tilting or turning something (Snow et al., 2006). Understanding the different senses of a polysemous word is important for correctly interpreting text and avoiding ambiguity. Another key concept in lexical semantics is the notion of semantic features (Bollegala et al., 2007). Semantic features are the basic building blocks of word meaning, and they can be used to describe the features that differentiate one word from another.

For example,

the word "cat” → can be described in terms of its semantic features such as "feline," "four-legged," and "meows."

Gencoglu (2020) elaborated Lexical semantics as the study of idioms composed of fixed phrases that convey a figurative sense, departing from the literal interpretation of their individual words.

For example,

the phrase "*kick the bucket*" → means "*to die*"

Even though the actual or explicit meaning of words, as defined in dictionaries - "kick" and "bucket" do not convey this idea.

Thus, it can be concluded that lexical semantics is a critical component of semantic analysis, as it focuses on understanding the meaning of individual words and how they combine to form larger units of meaning. By studying the relationships between words and their meanings, we can gain a deeper understanding of how language works and how it is used to communicate meaning (Scott & Dominic, 2003; Babu & Eswari, 2020).

Methodology

Figure 4 illustrates the complete workflow of the proposed approach. The figure provides a comprehensive overview of the different stages and components of the proposed method. It showcases the flow of data and information, from the input stage where the text is pre-processed, to the final output stage where the sentiment classification is obtained. Additionally, the figure highlights the various sub-modules utilized in the approach, their connections, and the interplay between them. Overall, the diagram in Figure 4 serves as a valuable reference point for understanding the overall framework of the proposed approach.

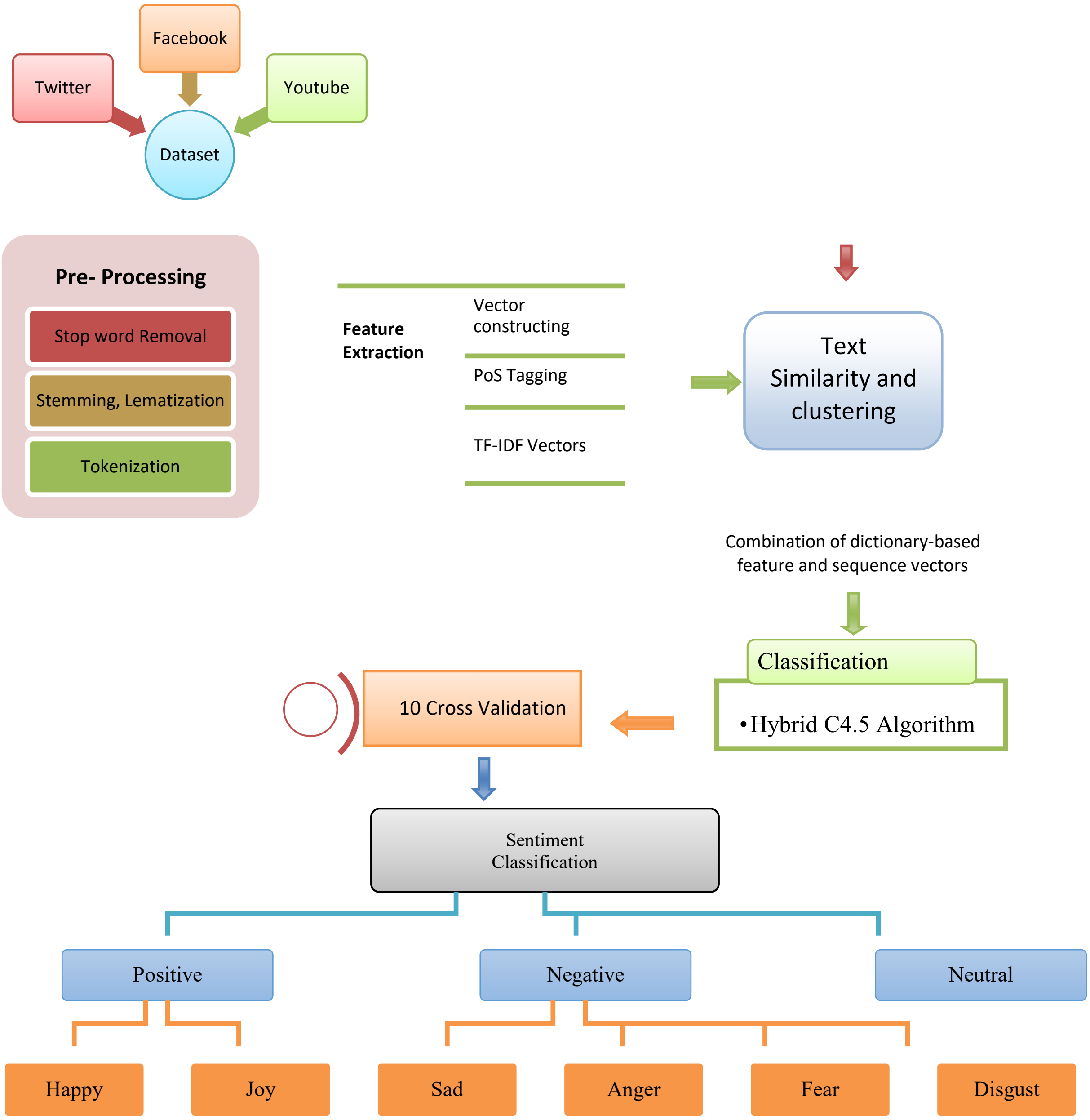


Figure 4. Work flow for Proposed Approached

Data Collection

In the current study, the collection of the corpus was carried out from three distinct platforms for online interaction, namely Twitter, Facebook and YouTube. The authors were able to retrieve a large dataset of 1, 06,700 comments from these platforms. The process of retrieving this raw data was likely a complex one, involving different platforms. The primary dataset is further elaborated in Table 1. This dataset is collected from March 2020 to October 2021.

Table 1. Data Statistics for Facebook, YouTube and twitter

Facebook	46,569	Total 1, 06,700
Twitter	36,536	
Youtube	23,595	

Once the raw data was collected, the primary dataset was manually congregated. This manual process likely involved going through each comment individually to determine whether it met the inclusion criteria for the study. This type of manual data collection can be time-consuming and resource-intensive, but it allows for a more accurate and targeted dataset that is better suited for the research at hand. Table 2 provides attribute information for the congregated data.

Table 2. Attribute Information of congregated dataset

Attributes	Attribute Description
id	Order of comment data frame
txt	Facebook comment/YouTube comment/tweet
crtd_at	Information about the content's date and time
rply	rerun status (Boolean value)
usr_lctn	User's geographical location
Hshtg	Comments with "#"
S_Class	Sentiment Class
S_Score	Sentiment Score

Data Mining

In order to extract a relevant set of text, preprocessing was conducted as outlined in (Mahmoud et al., 2016), which involved eliminating symbols and punctuations, converting characters to lowercase and truncating words to their root form. It entails converting unstructured textual material into a form that machine learning models can easily comprehend. In the context of the text mentioned, the preprocessing steps are as follows:

1. *Eliminating symbols and punctuations*: This step involves removing any non-alphanumeric characters, such as punctuation marks and special symbols, from the text. The objective of this step is to simplify the text and remove any noise that may interfere with the analysis.

2. *Converting characters to lowercase:* All uppercase letters must now be converted to lowercase. This prevents the model from treating terms with capital letters and those with lowercase letters as different words and ensures consistency in the text data.

3. *Truncating words to their root form:* This step involves reducing words to their Primitive Lemma. For instance, the word “running” would be reduced to "run". This is done to simplify the text and reduce the number of unique words that the model needs to analyze.

These operations were implemented using the R library. The R library is a frequently used instrument for analyzing data and manipulation. It provides functions for text preprocessing, such as removing symbols and punctuations, changing character cases, and stemming. R function gsub to remove specific symbols or punctuation marks from the comments and the tolower function to convert all characters in the comments to lowercase, making it easier to compare and analyze them are used. Additionally, the stem function to reduce words to their base or root form, which can help to standardize the language used in the comments is also used. By implementing these operations using the R library, the text data can be prepared for further analysis, such as feature extraction and model training. Subsequently, extraction was performed using TF-IDF.

TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF score is obtained in terms of TF (t,d)* IDF(t), where t = Term, d =Document.

The TF can be obtained using Eq. (1) that computes term weights in a document.

$$\text{TF}\ (t, d) = \frac{term\ t\ frequency\ in\ document\ d}{total\ words\ in\ document\ d} \tag{1}$$

The IDF is a metric that helps identify words that are important to a particular document but not necessarily important across the entire corpus. IDF is calculated based on the number of documents in which a word appears, and is defined using the formula shown in Eq. (2). By comparing a word's IDF value to a predefined threshold, it can be determined whether the word is a term (i.e., a significant word that should be included in analysis) or a stop word (i.e., a word that can be excluded from analysis without significant impact on the results).

$$\text{IDF}\ (t) = \log 2 \left(\frac{(total\ number\ of\ documents)}{(number\ of\ documents\ with\ term\ t)} \right) \tag{2}$$

In addition to identifying important words, IDF can also be used to distinguish between terms and stop words. Comparing a word's IDF value to a predefined threshold, it can be determined whether the word is a term (i.e., a significant word that should be included in analysis) or a stop word (i.e., a word that can be excluded from analysis without significant impact on the results).

Text Similarity and clustering

Text similarity is an important concept in various fields such as NLP, IR and ML that are used for tasks such as document clustering, and recommendation systems. Semantic similarity, on the other hand, refers to the degree of relatedness between the meanings of two text entities. This can be achieved through techniques such as NLP, which helps to identify and analyze the context of the text.

Term similarity is the measurement of similarity between individual terms or words used in a piece of text. This can be achieved through techniques like Jaccard similarity. The Algorithm 1 makes a higher cosine similarity score denotes a greater degree of similarity between the two documents being compared in this study's usage of cosine similarity as a matrix of document similarity within the text corpus that is transformed into a document term matrix (dtm).

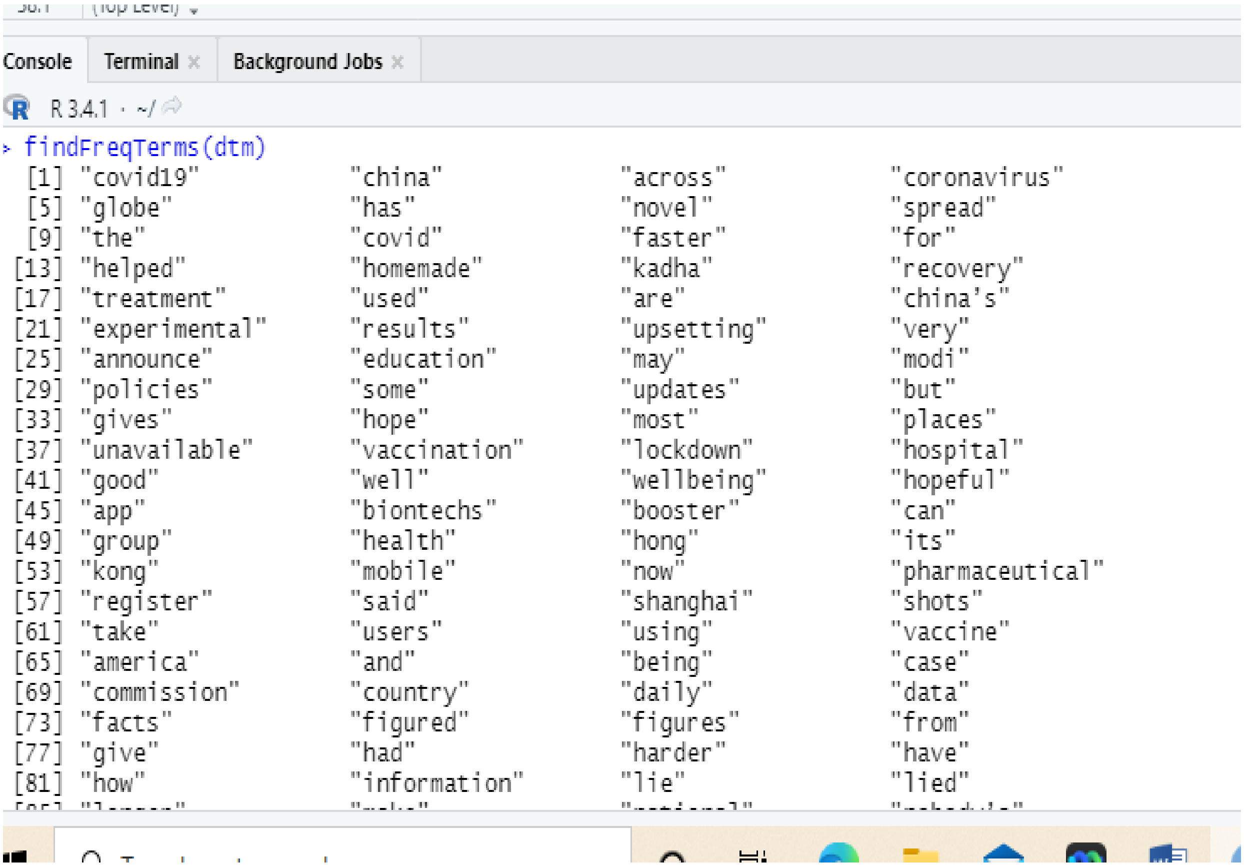
Algorithm 1: Text Similarity and clustering

Input:
- documents: a list of strings, representing the comments to cluster
- threshold: a numeric value between 0 and 1, representing the minimum Jaccard similarity threshold for clustering

Output:
- clusters: a list of lists, where each inner list represents a cluster of documents

```
Begin
1.  For each document d in documents do
        a) Tokenize d into individual words using a tokenizer function
        b) Store the tokenized document as a list of tokens T_d
2.  Create a feature matrix M, where each row represents a document (comment) and each column represents a token
3.  For each document d in documents do
4.      Initialize a row vector v_d with length equal to the number of unique tokens
5.      For each token t in T_d do
6.          Get the index i of t in the set of unique tokens
7.          Increment the i-th entry of v_d by 1
8.          Set the row of M corresponding to d to v_d
9.  Initialize a list of clusters C, where each cluster initially contains only one document
10. For each pair of documents (d_i, d_j) such that i < j do
        a) Calculate the Jaccard similarity S between T_d_i and T_d_j using the formula:
           $S = |T\_d\_i \cap T\_d\_j| / |T\_d\_i \cup T\_d\_j|$
        b) If S is above the threshold, merge the two clusters containing d_i and d_j into a single cluster in C
11. For each cluster c in C do
12.     Get the set of documents D_c in cluster c
13.     Initialize a document similarity matrix S_c, where each row and column represent a document in D_c
14.     For each pair of documents (d_i, d_j) in D_c such that i < j do
            a) Calculate the cosine similarity between the feature vectors of d_i and d_j using the formula:
               $\cos(theta) = dot\_product(v\_i, v\_j) / (norm(v\_i) * norm(v\_j))$
            b) Set the (i,j)-th and (j,i)-th entries of S_c to the cosine similarity between d_i and d_j
            c) Output S_c as the pairwise similarities between documents within cluster c
End
```

Following the combination of dictionary-based feature and sequence vectors, a text-bag is constructed, with each word represented as a weighted vector from the list obtained as a result of Algorithm 1. It initializes the clusters list with each document in its own cluster. The output on the basis of frequency of terms for the dtm as weighted vectors is shown in Figure 5.



```
> findFreqTerms(dtm)
 [1] "covid19"        "china"          "across"         "coronavirus"
 [5] "globe"          "has"            "novel"          "spread"
 [9] "the"            "covid"          "faster"         "for"
[13] "helped"         "homemade"       "kadha"          "recovery"
[17] "treatment"      "used"           "are"            "china's"
[21] "experimental"   "results"        "upsetting"      "very"
[25] "announce"       "education"      "may"            "modi"
[29] "policies"       "some"           "updates"        "but"
[33] "gives"          "hope"           "most"           "places"
[37] "unavailable"    "vaccination"    "lockdown"       "hospital"
[41] "good"           "well"           "wellbeing"      "hopeful"
[45] "app"            "biontechs"      "booster"        "can"
[49] "group"          "health"         "hong"           "its"
[53] "kong"           "mobile"         "now"            "pharmaceutical"
[57] "register"       "said"           "shanghai"       "shots"
[61] "take"           "users"          "using"          "vaccine"
[65] "america"        "and"            "being"          "case"
[69] "commission"     "country"        "daily"          "data"
[73] "facts"          "figured"        "figures"        "from"
[77] "give"           "had"            "harder"         "have"
[81] "how"            "information"    "lie"            "lied"
[85] "longer"         "make"           "national"       "schedule"
```

Figure 5. Frequently used words w.r.t. feature and sequence vectors

Tokens are generated to replace the correct form of the words stored in a knowledge base. This knowledge base is utilized to arrange the corpus, where sentence chaining is accomplished using three attributes: id, txt, and hshtg. The corpus is then clustered into a database for COVID-related text, and a distinct feature set is obtained, enabling accurate classification and tagging of lexicons in a semantically organized corpus. The word cloud for this clustered text bag is illustrated in Figure 6.

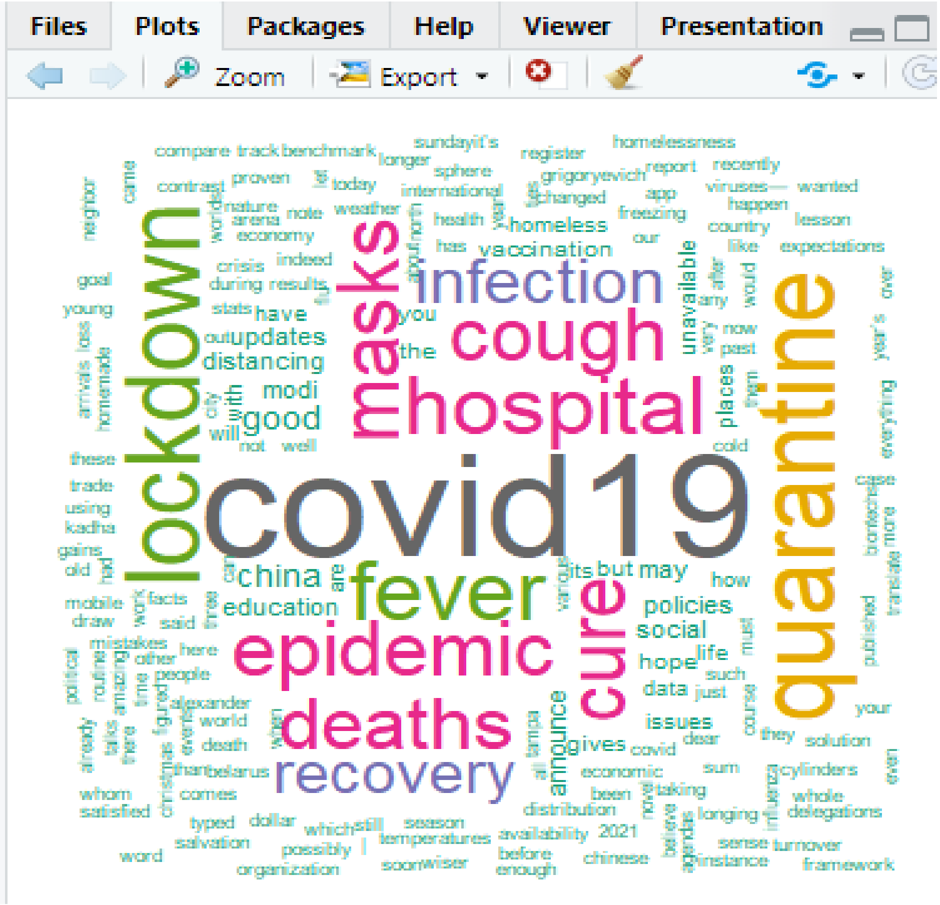


Figure 6. Visualization of Word Cloud for Clustered Corpus

Classification

Decision Trees. Decision Trees are used to generate data-based models that help in differentiating labels for diverse classes or differentiating values for the decision-making problems (Scott & Dominic, 2003). Tasks that require classification and regression models, make use of decision trees where the training dataset is fed to the system (supervised learning) (Jalil et al., 2021; Khalid et. al., 2021) for building models. The traditional decision tree based C4.5 algorithm assigns a class label to each leaf node based on the majority class in that node. The hybrid C4.5 algorithm calculates a sentiment score for each record by adding up the percentages of each sentiment class encountered in the leaf nodes. This allows the Algorithm 2 to provide more nuanced predictions than a simple class label. It also includes cost-sensitive learning, which allocates distinct misclassification costs to diverse classes. This can help improve the performance of the model on imbalanced datasets, where one class is significantly underrepresented compared to the others. In such cases, misclassifying examples from the sub class can have a much greater impact on the overall performance of the model than misclassifying examples from the super class.

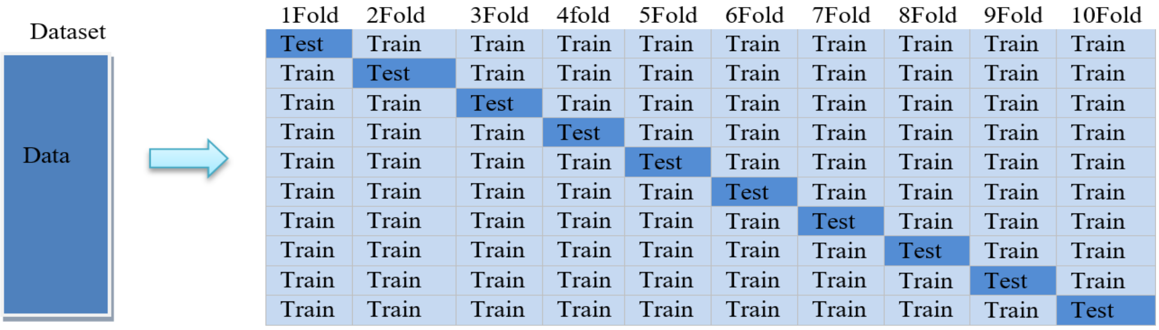
```
Algorithm 2: Hybrid C4.5 Algorithm for classification
Input:
    - Training dataset D
    - Threshold value T
    - Maximum tree depth L
    - Ensemble size E
    - Cost matrix C with misclassification costs for each class
Output:
    - Ensemble of decision trees

Begin
1. Initialize ensemble to empty set
2. For i = 1 to E do
3.     Sample a training subset D' from D
4.     Train a decision tree T_i with C4.5 algorithm using D' and the cost matrix C
5.     While T_i has depth > L do
6.         Prune the subtree with the lowest decrease in accuracy using reduced error pruning
7.     Compute the accuracy of T_i on the validation set
8.     If accuracy of T_i on validation set > T, add T_i to the ensemble
9. Return the ensemble of decision trees
10. For each decision tree T in the ensemble:
11.    For each comment in the test set:
12.        Classify the comment using T, taking into account the misclassification costs
13.    For each subclass:
        a.  Compute the frequency of the subclass in the set of comments classified as positive
        b.  Compute the frequency of the subclass in the set of comments classified as negative
        c.  Compute the frequency of the subclass in the set of comments classified as neutral
14.    Assign the comment the S_Class and S_Score with the highest frequency
15. Return the classifications for all examples in the test set
16. For each attribute A:
        a. Compute the information gain w.r.t. to S_Score
        b. Compute the weighted information gain w.r.t. to S_score
                weighted_info_gain(A, S_Score) = info_gain(A) * cost(S_Score)
17. For each leaf node:
        a.  Compute the expected cost of misclassification for each S_class, using the formula:
                expected_cost(class) = cost(S_Class) * P(comment is misclassified as S_Class | reaches the leaf node)
        b.  Assign the comment to the S_Class and respective S_Score with the lowest expected cost
End
```

Steps for classification

Step 1: A training dataset and a testing dataset are created from the dataset in this step. To facilitate this procedure, the widely utilized Python library scikit-learn (sklearn) has a method named "train_test_split". Using a common split of 80% training data and 20% test data, this programme randomly divides the dataset into sections

Step 2: In this step, test_size parameter value is set as 0.2 and a random_state variable is taken as a pseudo-random number generator state for random sampling.

Step 3: The classifier method is then configured to use the test and train datasets to determine how well the test and train models performed. In this study, k-fold cross validation with k=10 was used. Figure 7 below gives the idea for cross-validation and cluster separation for the test and train sets.



1Fold	2Fold	3Fold	4fold	5Fold	6Fold	7Fold	8Fold	9Fold	10Fold
Test	Train	Train	Train	Train	Train	Train	Train	Train	Train
Train	Test	Train	Train	Train	Train	Train	Train	Train	Train
Train	Train	Test	Train	Train	Train	Train	Train	Train	Train
Train	Train	Train	Test	Train	Train	Train	Train	Train	Train
Train	Train	Train	Train	Test	Train	Train	Train	Train	Train
Train	Train	Train	Train	Train	Test	Train	Train	Train	Train
Train	Train	Train	Train	Train	Train	Test	Train	Train	Train
Train	Train	Train	Train	Train	Train	Train	Test	Train	Train
Train	Train	Train	Train	Train	Train	Train	Train	Test	Train
Train	Train	Train	Train	Train	Train	Train	Train	Train	Test

Figure 7. Cross-validation with separation for the test and train sets

Step 4: In the final step the sentiment classification for the test set is achieved on the basis of S_class and S_score, for which a Hybrid C4.5 algorithm is implemented. For S_class the classification algorithm used three sets of words that are positive, negative and neutral and tagged them along the weighted vectors observed during text clustering. For each decision tree in the ensemble, we classify each example in the test set using the tree, taking into account the misclassification costs. This means that each example will be assigned a sentiment label of positive, negative, or neutral, based on which leaf node it reaches in the decision tree.

The S_class was classified by sentiment score for the three classes of sentiments and it is achieved using equation (3) for three social media platforms.

$$S\_Class = \frac{no.of\ positive\ words - no.of\ negative\ words}{total\ no.of\ words} \quad (3)$$

For each subcategory, we compute the frequency of that subcategory in the set of examples that were classified under S-Class (as positive, negative and neutral respectively). This is done by counting the number of examples in each set that are associated with the subcategory. The S_score is calculated for each word as weighted vector where sentiment class for each three major categories of sentiment "S_class" is further exploited using decision tree. The S_Score was derived over the S_class and is achieved using equation (4) where the score was achieved (out of 5).

$$S\_Score = G(S) - \sum\nolimits_{t \in T} p(t),\ G(t),\ S\_class \quad (4)$$

Finally, for each comment in the test set, we assign it to the S_Class and generate S_Score with the highest frequency. This means that each example will be assigned a sentiment label (positive, negative, or neutral) as well as a subclass label (happy, joy, sad, disgust, fear, or anger) based on the subcategory that has the highest frequency in the set of examples with the same sentiment label.

Results and Discussion

In this study, a total of 46,569 comments, 36,563 tweets and 23,568 and replies (total=1,06,700 comments) were obtained from Facebook, twitter and YouTube respectively. The classification algorithm based on decision trees was used for extracting sentiments into six categories and a neutral tag. The results primarily for sentiment class are shown in Table 3.

Table 3. Sentiment class categorization of the corpus

Corpus	S_Class			Total
	Positive	**Negative**	**Neutral**	
Facebook	19313	25011	2245	46,569
Twitter	13265	19653	3645	36,563
YouTube	12651	4563	6381	23,595

On Facebook, out of a total of 46,569 instances, the majority were classified as Negative (25,011), followed by Positive (19,313), and Neutral (2,245). On Twitter, out of a total of 36,563 instances, the majority were classified as Negative (19,653), followed by Positive (13,265), and Neutral (3,645). On YouTube, out of a total of 23,595 instances, the majority were classified as Positive (12,651), followed by Neutral (6,381), and Negative (4,563). The results indicate that Negative sentiments were more prevalent on Facebook and Twitter, while Positive sentiments were more prevalent on YouTube. This information can be useful for businesses and organizations to understand the sentiment patterns of their target audience on different social media platforms and develop targeted marketing and communication strategies accordingly.

After text mining, text similarity and clustering and vector-space-modelling, the separation for test and train datasets is obtained that utilized 10-fold cross-validation. In addition, the results for six sub categories of emotions of the given classification algorithms on three datasets, based on S_Score are given in Table 4.

The proposed approach performed text mining, text similarity and clustering, and vector-space-modelling techniques on COVI19 related data over social media platforms. After performing these techniques, the data is separated into test and train datasets using 10-fold cross-validation.

Table 4. S_Score (out of 5) for three Social Media Platforms

Sentiment subclass	S_Score (out of 5)		
	Facebook	twitter	YouTube
Happy	3.2	2	2
Sad	4.2	4.2	4.2
Disgust	0.2	0.2	0.2
Angry	2.1	2.1	2.1
Fear	2.6	2.6	2.6
Joy	3.4	3.4	3.4
Neutral	3.5	3.5	3.5

Overall, the results of sentiment classification algorithms on the three social media platforms suggest that Sad sentiment is the most prevalent emotion among users, while Disgust is the least prevalent emotion. The results can be useful for understanding the sentiment patterns of users on different social media platforms and for developing effective strategies for targeted marketing and communication. Table 4 displays the outcomes of sentiment categorization algorithms on Facebook, Twitter, and YouTube, three social media sites. The S_Score (out of 5) is provided for each of the six subcategories of emotions: "happy", "sad", "disgust", "angry", "fear", "joy" and "neutral". It is observed that the Sad sentiment had the highest S_Score (4.2) on all three platforms, while Disgust had the lowest S_Score (0.2) on all platforms. The Happy sentiment had the highest S_Score on Facebook (3.2) but had the lowest S_Score on Twitter and YouTube (2.0). The Anger and Fear sentiments had the same S_Score (2.1) on all three platforms, while Joy and Neutral had the same S_Score (3.4) on all three platforms.

Based on the data in the table 3, it seems that while discovering major classification the sentiments were overlapped and the overall emotional tone of the social media platforms is slightly negative, with a total of 44,727 negative instances, compared to 34,229 positive instances and 6,926 neutral instances. On the other hand, while considering table 4, overall emotional tone of the social media platforms is generally neutral, with a S_Score of 3.5 across all three platforms. However, there are some variations in the emotions expressed on each platform. Facebook seems to have the highest level of happiness, with a S_Score of 3.2, while YouTube and Twitter have relatively low levels of happiness, with a S_Score of 2 for both platforms. On the other hand, all three platforms have relatively high levels of sadness, with a S_Score of 4.2 across the board. The emotions of disgust, anger, and fear are relatively low across all three platforms, with a S_Score of 0.2 for disgust and 2.1 for both anger and fear. Overall, the emotional tone of these social media platforms is mixed, with a relatively high level of sadness and a relatively low level of happiness. The diagrammatic summary of emotions for the above-mentioned discussion is shown in figure 8.

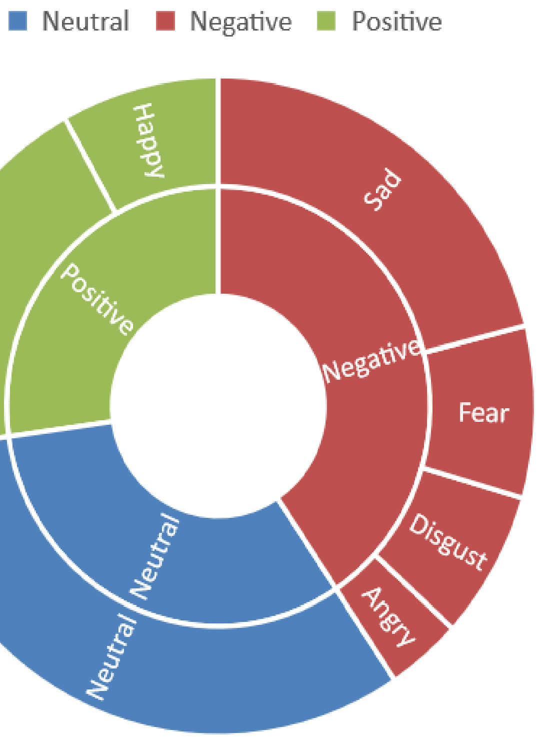


Figure 8. Sentiments based on S_Classes and sub-classes w.r.t. S_Score

Comparative Analysis

The performance of classification algorithms can be assessed by evaluating the test and train models using a confusion matrix. This matrix is used to calculate several performance metrics, including accuracy (ACC), precision (PRE), sensitivity (SENS), and F-measure (F). The F-measure is calculated as the harmonic mean of precision and sensitivity, as shown in equations (5-8). A perfect F-measure score of 1 indicates perfect specificity and sensitivity, while a score of 0 indicates poor performance.

$$Accuracy = \frac{T_p + T_N}{T_P + F_N + F_P + F_N} \tag{5}$$

$$Sensitivity = \frac{T_P}{T_P + F_N} \tag{6}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{7}$$

$$F - measure = \frac{2 * Precision * Sensitivity}{Precision + Sensitivity} \tag{8}$$

Table 5. Relative Comparison with existing work

Reference	Classification Approaches	ACC	PRE	SENS	F
(Babu & Eswari, 2020)	Ensemble of CT-BERT, RoBERTa, and SVM	n/a	0.89	0.87	0.88
(Nagamanjula & Pethalakshmi, 2020)	*LAN*^{2}*FIS* (Logistic Adaptive Network Based on Neuro-Fuzzy Inference System)	0.89	0.88	0.89	0.89
(Khan, Nabiul, & Dhruba, 2021)	Deep Learning-Based LSTM And Bidirectional LSTM (Bi-LSTM)	0.90	**0.92**	0.91	**0.91**
(Rustam et. al., 2021)	Naive Bayes, Multinomial Naive Bayes and Decision Trees	0.81	0.82	0.86	0.83
(Shahi, Sitaula & Paudel, 2022)	Extra Tree Classifier using concatenated features	0.93	0.90	0.89	0.89
(Jalil, Javed, Rehman, 2022)	XGBoost (eXtreme Gradient Boosting) - tree base model	**0.96**	0.89	0.86	0.88
(Kumar & Sharma, 2023)	IAOCOOT method	0.82	0.85	0.83	0.84
(Kukkar et. al., 2023)	A novel lexicon-based system for lengthy words	0.89	0.85	0.88	0.87
Proposed Decision tree-based Classification using Hybrid C4.5		**0.96**	**0.92**	**0.94**	**0.92**

***Note: The highlighted values have shown better results in the respective parametric column**

Overall, the results suggest that deep learning-based models, such as LSTM and Bi-LSTM and ensemble methods such as the combination of CT-BERT, RoBERTa and SVM, perform well in classification tasks, while tree-based models, such as Extra Tree Classifier and XGBoost and the proposed decision tree-based Hybrid C4.5 can also be used to achieve high accuracy in the field of sentiment analysis. The results are shown in Table 5.

Conclusion

The enormous amount of data retrieved from multiple social media platforms encounters the challenge of making the information convenient in such a way that that operations like summarization, sentiment analysis, refutation, translation, information processing etc., can be acquired. The resolution to this challenge is to develop approaches, methods and applications that can understand the denotation of basic entities and are capable of performing a high-level semantic analysis on the available information.

The proposed study demonstrated higher accuracy in sentiment classification compared to seven other existing methods and equal to one XGBoost (eXtreme Gradient Boosting) - tree base model. According to the data presented in Table 5, XGBoost exhibited superior accuracy results amongst the preexisting approaches. The steps proposed for hybrid C4.5 algorithm has also demonstrated superior values of Precision, Sensitivity, and F-measure with the other state-of-the-art machine learning algorithms in the field of sentiment analysis. The majority of recent research on sentiment classification has utilized a diverse range of machine learning techniques. However, it has been observed that decision trees could potentially yield superior results if implemented with cost-sensitive learning. This involves conveying varying “misclassification costs” to multiple classes based on lexical and semantic features, as well as sequence vectors.

Future work aims to uncover new insights of emotions like sarcasm, irony and many more variations of lexical and semantic analysis through the exploration of diverse weighting approaches over feature vectors and sequence vectors conducting analysis on multiple social media datasets.

Conflict of interest

The authors of this paper state that they do not have any competing financial interests or personal relationships that could have influenced their work. We would like to verify that there are no conflicts of interest related to this publication and that no significant financial support has been received that could have influenced the outcome of the research. This statement indicates that the authors have taken steps to ensure that their work is unbiased and free from any undue influence.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Babu, Y. P., & Eswari, R. (2020). CIA_NITT at WNUT-0p-2020 task 2: classification of COVID-19 tweets using pre-trained language models. arXiv preprint arXiv:2009.05782.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. Proceedings of 16th World Wide Web Conference (WWW16), 757–766.

De Las Heras-Pedrosa, C., Sánchez-Núñez, P., & Peláez, J. I. (2020). Sentiment analysis and emotion understanding during the COVID-19 pandemic in Spain and its impact on digital ecosystems. *International Journal of Environmental Research and Public Health*, 17(15), 5542.

Elghazaly, T., Mahmoud, A., & Hefny, H. A. (2016). Political sentiment analysis using twitter data. In Proceedings of the International Conference on Internet of things and Cloud Computing (pp. 1-5).

Garcia, K., & Berton, L. (2021). Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Applied Soft Computing*, 101, 107057.

Gencoglu, O. (2020). Large-scale, language-agnostic discourse classification of tweets during COVID-19. *Machine Learning and Knowledge Extraction*, 2(4), 603-616.

Hearst Marti, A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics.

Jalil, Z., et al. (2021). Covid-19 related sentiment analysis using state-of-the-art machine learning and deep learning techniques. Public Health Frontiers, 9.

Jalil, Z., Javed, A., Rehman, M. A., et al. (2022). COVID-19 Related Sentiment Analysis Using State-of-the-Art Machine Learning and Deep Learning Techniques. Frontiers in Public Health.

Jelodar, H., Wang, Y., Orji, R., & Huang, S. (2020). Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. *IEEE Journal of Biomedical and Health Informatics*, 24(10), 2733-2742.

Khan, M., Nabiul, A. K., & Dhruba, A. (2021). Deep learning-based sentiment analysis of COVID-19 vaccination responses from Twitter data. Computational and Mathematical Methods in Medicine.

Kukkar, A., Mohana, R., Sharma, A., Nayyar, A., & Shah, M. (2023). Improving sentiment analysis in social media by handling lengthened words. *IEEE Access*, 11, 9775-9788. https://doi.org/10.1109/ACCESS.2023.3238366

Kumar, R., & Sharma, S. C. (2023). Hybrid optimization and ontology-based semantic model for efficient text-based information retrieval. *Journal of Supercomputing*, 79(3), 2251-2280. https://doi.org/10.1007/s11227-022-04708-9

Nagamanjula, R., & Pethalakshmi, A. (2020). A novel framework based on bi-objective optimization and LAN2FIS for Twitter sentiment analysis. *Social Network Analysis and Mining*, 10(1). doi:10.1007/s13278-020-00648-5.

Naithani, K., & Raiwani, Y. P. (2022). Novel ABC: Aspect Based Classification of Sentiments Using Text Mining for COVID-19 Comments. In Machine Learning, Image Processing, Network Security and Data Sciences (pp. 199-208). Springer

Naseem, U., Razzak, I., Khushi, M., Eklund, P. W., & Kim, J. (2021). COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis. *IEEE Transactions on Computational Social Systems*, 8(4), 1003-1015.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. arXiv preprint cs/0205070.

Qorich, M., & El Ouazzani, R. (2023). Text Sentiment Classification of Amazon Reviews Using Word Embeddings and Convolutional Neural Networks. *Journal of Supercomputing*.

Rana, S., & Singh, A. (2016). Comparative analysis of sentiment orientation using SVM and Naïve Bayes techniques. In 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun (pp. 106-111). doi: 10.1109/NGCT.2016.7877399.

Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. S. (2021). A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. Plos one, 16(2), e0245909.

Safder, Z., Mahmood, R., Sarwar, S., Hassan, S., et al. (2021). Sentiment analysis for Urdu online reviews using deep learning models. Expert Systems, e12751. https://doi.org/10.1111/exsy.12751.

Scott, C., & Dominic, W. (2003). Using LSA and Noun Coordination Information to Improve the Recall and Precision of Automatic Hyponymy Extraction. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 (pp. 111-118).

Shahi, T., Sitaula, C., & Paudel, N. (2022). A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification. Computational Intelligence and Neuroscience.

Sharon Caraballo, A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (pp. 120-126). Association for Computational Linguistics.

Snow, R., Jurafsky, D., & Ng, A. Y. (2006). Semantic taxonomy induction from heterogeneous evidence. Proceedings of the ACL-COLING, 801-808.

Vashisht, G., & Jaillia, M. (2021). Enhanced lexicon E-SLIDE framework for efficient sentiment analysis. *International Journal of Information Technology*, 13, 2169-2174. doi:10.100

Verma, M. (2017). Lexical Analysis of Religious Texts Using Text Mining and Machine Learning Tools. International Journal of Computer Applications, 168(8).

Wankhade, M., Annavarapu, C. S. R., & Abraham, A. (2023). MAPA BiLSTM-BERT: Multi-Aspects Position Aware Attention for Aspect Level Sentiment Analysis. *Journal of Supercomputing*.

Zhang, M., Liu, L., Mi, J., Li, Q., & Zhang, L. (2023). Enhanced dual-level dependency parsing for aspect-based sentiment analysis. *Journal of Supercomputing*, 79(1), 6290-6308. https://doi.org/10.1007/s11227-022-04898-2

Zhang, Y., Lyu, H., Liu, Y., Zhang, X., Wang, Y., & Luo, J. (2020). Monitoring depression trend on Twitter during the COVID-19 pandemic. arXiv preprint arXiv:2007.00228. [Original source: https://studycrumb.com/alphabetizer.

Bibliographic information of this paper for citing:

Naithani, Kanchan; Raiwani, Y. P.; Alam, Intyaz & Aknan, Mohammad (2023). Analyzing Hybrid C4.5 Algorithm for Sentiment Extraction over Lexical and Semantic Interpretation. *Journal of Information Technology Management,* 15 (Special Issue), 57-79. https://doi.org/10.22059/jitm.2023.95246