Machines and Algorithms
http://www.knovell.org/mna

Research Article,

Sentiment Analysis on Tweets using N-Grams and Lexicon

Muhammad Sanaullah [1, *], Rabea Saleem[2, *] and Fatima Riaz [3]

[1] Department of Computer Science, Bahauddin Zakariya University, Multan, 60800, Pakistan
[2] Department of Computer Science, Air University, Multan, 60000, Pakistan
[3] Lecturer Computer Science, Higher Education Department, Multan, 60000, Pakistan
[*]Corresponding Author: Muhammad Sanaullah. Email: drsanaullah@bzu.edu.pk

Received: 02 April 2025; Revised: 05 May 2025; Accepted: 04 July 2025; Published: 01 August 2025
AID: 004-02-000051

Abstract: Twitter is serving as micro-blogging platform where freedom is given to user to express their opinion and share information about any subject through short messages known as tweets. Tweet composed of textual data that can be classified into positive, negative or neutral sentiment. This classification is often based on the analysis of n-grams, which involves examining the frequency and combination of words used. This research article presents a technique which filters out the noise from tweets and apply the scoring mechanism to sentiments that assigns the score between -1 and +1. The proposed techniques results are validated by manually scored evaluations for same tweets. Additionally, the study compares the effectiveness of different n-gram techniques for sentiment analysis.

Keywords: Sentiment analysis; n-grams; twitter; tweets; lexicon;

1. Introduction

In this modern age of information, microblogging platforms and social media sites are the major source of information. These platforms allow user to express their feelings, emotions and their opinions. Various entities, including individuals, organizations, politicians, governments, and surveys, are increasingly utilizing these sites to fulfil their respective objectives. For instance, production companies may seek instant feedback on their products, political figures may engage in discussions on various issues, and governments may seek public opinions on policies and regulations.

In response to these objectives, a substantial volume of opinions is received from the respective users. Reviewing all of these opinions is a time-consuming and labor intensive task [1]. Consequently, there is a need for an automated tool that can classify these opinions, based on their nature, and allowing management to easily understand the behavior of their users. Sentiment Analysis, is a technique employed to analyze large amounts of data, serves the purpose of discerning user attitudes and opinions towards a given topic. Numerous machine learning techniques are utilized for opinion classification; however, due to inherent complexities in the text, such as unstructured content, negation detection, slang and abbreviation usage, irregular or incorrect wording, insufficient words, emoticons, symbols, tagging, and URLs, achieving the desired objective is not frequently attained [2].

To handle the inherent noise in such data and comprehend the underlying semantics of the text, an automated technique is essential. This technique should not only enable classification but also facilitate the measurement of the degree of emphasis in the text. To address this need, this paper proposes a comprehensive approach to sentiment analysis. Initially, the technique eliminates irrelevant information

and subsequently applies an n-grams (unigrams, bigrams, and trigrams) approach to analyze the sentiments expressed. The sentiment analysis is conducted utilizing SentiWordNet[3], a lexicon-based resource, to calculate the polarity score of the opinions, indicating the extent of positivity, negativity, or neutrality. These scores range from -1 (fully negative) to +1 (fully positive). These polarity scores are then employed to compute the sentiments and polarity of the opinions.

For the implementation of the proposed technique, Twitter was selected from among various microblogging sites, based on the following factors:

- The character limit of Twitter messages is restricted to 280 characters.
- Twitter has a larger user base consisting of more professional and engaged members.
- Many public figures utilize Twitter as a platform to discuss issues or policies.
- Twitter contains a significant volume of movie, event, and product reviews.

In the context of Twitter, user opinions are referred to as "tweets." To assess the reliability and accuracy of the proposed technique on these tweets, human experts were involved. These experts, unaware of the scores calculated by the technique, independently assigned similar scores to corresponding opinions, thereby confirming the technique's effectiveness.

The study contributes to present a novel technique for sentiment analysis on Twitter data, which filters out noise, assigns sentiment scores, and classifies tweets, with the reliability and accuracy of this approach validated through comparison with manually scored evaluations.

The content is the paper is structured as: the related work is presented in Section 2, the proposed technique is presented in Section 3, the implementation of the proposed technique is presented in Section 4, the results are shown in Section 5, and the last section consist of concluding remarks and future directions.

2. Related Work

In [4], authors gathered e-learning public opinions from twitter when the COVID-19 outbreak. People's opinions about the e-learning are classified into positive and negative polarities. The authors used the SVM model on the tweets and claimed that it performs even better than deep learning models. Most of the world agrees upon e-learning during the pandemic.

Another paper [5] worked on comparing the different methods of sentiment analysis on twitter gathered data. Authors also showed and compared the results. The proposed work showed the techniques of Bag of Words (BoW) and N-grams in a lexicon-based corpus. The work applied seven machine learning algorithms namely SVM, Naïve Bayes, Logistic Regression, Multi-layered perceptron, Best-first trees, functional trees and famous C4.5. Authors also tried combination of the algorithms and compared the results.

In another paper [6], the authors presented a hybrid approach for analysis of tweets. Authors searched the polarity of the words from Pool of Words in the lexicon-based corpus and trained the algorithms by polarity of the words.

In [7] article, the authors proposed a Naïve Bayes classifier that compares the 1-gram, 2-grams and 3-grams. The results showed that 2-grams work great and give best coverage of sentiments. Authors also used the negation attached with the words. This technique improved the classification process 2% on average. The authors distinguish the word occurrences that are ambiguous based on salience, entropy and sets. The two saliences showed good results but also created ambiguity in the system.

Another work [8] performed the sentiment analysis on microblogs and analyzed the role of word semantics. By focusing on semantics of the words give more accurate results and build good sentiment analysis models for twitter. Authors focus on two types of semantics: one based on context (also captured from words) and other concept-based semantics (gathered from other sources).

In [9], authors discussed a model based on semantics. This model identifies the tweet as an entity like Person, Organization, etc. Authors did not remove the stop words in their model because removing stop words effect the classifier. Most of the techniques used in sentiment rely on syntactic structure of words like great, bad. However, these techniques are considered weak because they do no tell the semantics of the

words in text. This paper improves the results by using basic sentiment analysis approaches in combination with stemming, two step classification and negation detection. Moreover, negation detection is also very important technique used in sentiment analysis to detect the negative words like ‘not’, ‘no’, ‘never, etc. By this the sentiments of the nearby words changed.

Paper [10] used the linguistic features of the language to detected the sentiment from tweets. Author captured the information about creative and informal language usually found in microblogging sites. For this, author utilized the supervised algorithms to the problem and trained the model on existing hashtags in the Twitter data.

In another paper [11], the authors worked on a lexicon-based sentiment discovering technique [8]. The lexicon used in the technique is built from the taxonomy. The taxonomy used in the sentiment technique contains positive, negative, negation, stop words and phrases. The text in the tweet usually contains the hashtags, emoticons, word variations, etc. Preprocessing of the tweet involves stemming, emoticon detection, hashtag detection, word shortening and normalization. The lexicon-based technique uses normalized tweet and classifies the tweet as negative, positive based on the contextual orientation of the words. The proposed system shows the F-score of 0.8004 when tested on the unseen tweets.

In [12], authors presented a model named LeBERT that used social media reviews and created an embedded model using BERT and sentiment lexicon. The authors used n-grams and also applied the CNN layers to the model and predicted a sentiment class. Author applied the model on three datasets about hotel, movies or products reviews. The f-score of 88.73% shows that the model outperforms most of the state of art models.

In this study [13], author introduced a novel approach for Sentiment Analysis with Convolutional Neural Network Optimized via Arithmetic Optimization Algorithm (TSA-CNN-AOA). The research extracted and analyzed 173,638 tweets from July 25, 2020, to August 30, 2020, utilizing FastText Skip-gram for information extraction. The convolutional neural network was used to extract feature and optimize feature selection through an arithmetic optimization algorithm (AOA). Classification of tweets as positive, negative, or neutral was performed using K-nearest neighbors (KNN), support vector machine, and decision tree algorithms. The results showed that TSA-CNN-AOA (KNN) achieved an impressive accuracy rate of 95.098%.

In this research [14], author proposed a hybrid approach combining lexicon-based methods with deep learning models to improve sentiment accuracy, assessing the impact of TextBlob compared to other methods like Afinn and VADER. Results show that models perform better with TextBlob-assigned sentiments. Proposed model stands out with a high accuracy of 0.97, and support vector and extra tree classifiers achieve top accuracy scores of 0.92, using TF-IDF and BoW.

In [15], authors presented a new framework that uses n-grams with existing lexicons. The framework is applied on three lexicon-based datasets. The authors experiments show that the proposed work outperforms the existing lexicon-based approaches. The proposed work utilizes the attribute vector containing binary value polarity scores without disturbing the word order of text instead of leveraging the average polarity score of the words in the sentence.

3. Methodology

The proposed methodology is based on six core steps, as shown in Figure 1, the work performed at each step is explained in the following subsections.

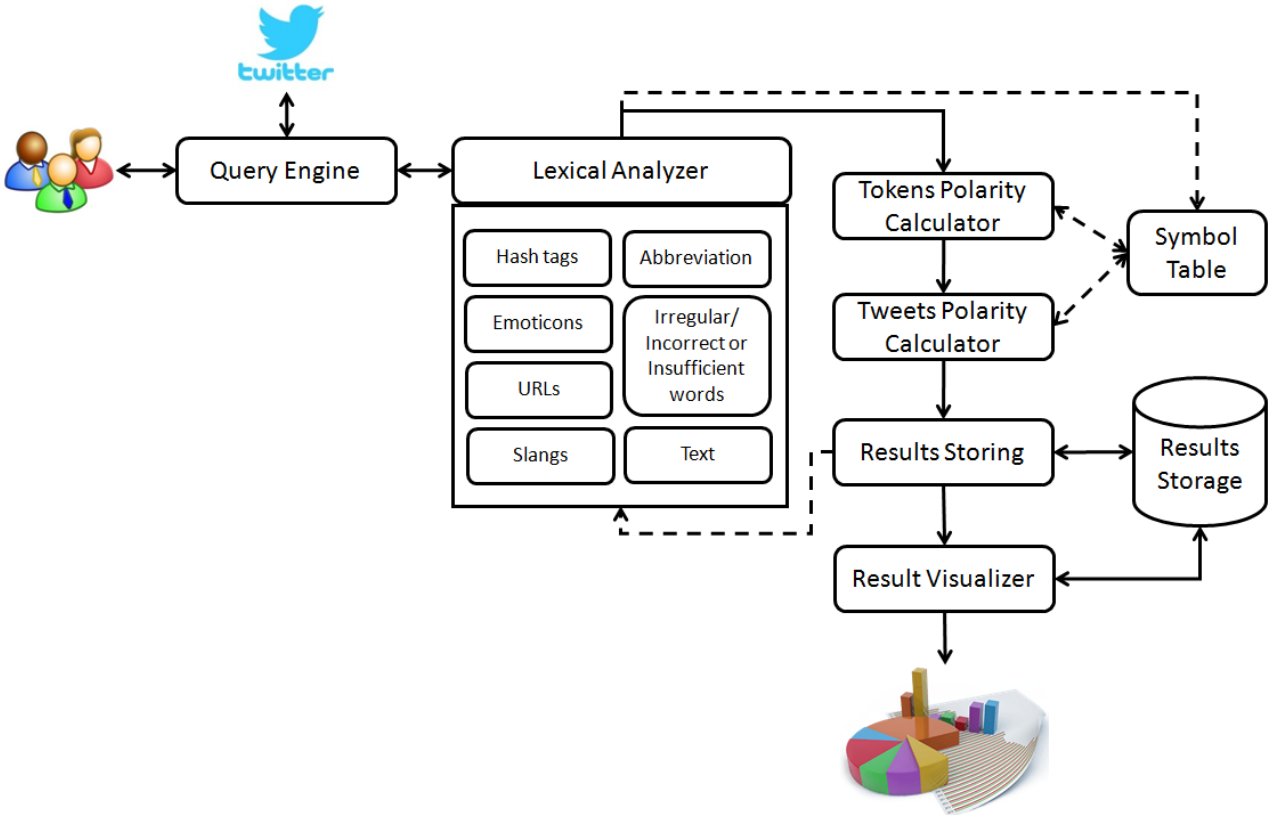

Figure 1: Methodology of Proposed Framework

3.1. Data Collection

In proposed system data set is collected from twitter in real-time. Users engage with the proposed system through a sophisticated interface. This interface enables users to submit a string or relevant keywords about the topic for which they seek sentiment analysis. For example, “Donald Trump”. The submitted keyword, denoted as Q (Quey), is then transmitted to the Query Engine (QE). The QE leverages the Twitter API to extract a set of real-time tweets, denoted as T, from twitter. Total number of tweets extracted from twitter is 100. The extraction time for tweets is a few seconds. The mathematical representation of the process can be expressed by the following equations.

- Q = Query (Keyword that is used to search tweets.)
- T = Extracted Tweets based on keyword.
- Q➔T
- T= [T1, T2, T3, … T_N]

The data extraction process relies on the utilization of the Twitter API. This API facilitates the collection of data, specifically tweets, from twitter platform. By creating a Twitter API, users gain access to and can store the desired data. Users initiate requests to the API to obtain from twitter data, and the API responds by returning data that aligns with the user's query.

3.2. Lexical Analyzer

After getting tweets from Twitter API, preprocessing is performed on tweets to make them more refined. Predefined python libraries Tweepy, NumPy, NLTK, pandas, csv, re, matplotlib and TextBlob are used in preprocessing. Preprocessing tasks involved in Lexical Analyzer are given below.

1. **Tokenization:** The process of splitting the input text into individual words is tokenization. For example, the sentence "I love this Car" would be tokenized into three tokens: ["I", "love", "this”,

"Car"]. Proposed system performs n-grams (unigram, bigram, trigram) tokenization one by one to perform more accurate sentiment analysis

2. **Stop Words Removal:** The words that do not has useful meanings and used as supporting words in sentences, known as stop words (e.g., "the," "and," "in"). These stop words increase the noise in our text. So, we have to remove these words.
3. **Lowercasing:** The process of conversion of capital letters to small letters is known as Lowercasing. In computer capital letter is different from lower case letter. The letter "M" is different from "m" in computer understanding. That's why the text is converted into lowercase to remove the uncertainty.
4. **Punctuation and Special Characters:** Punctuation marks and special characters can have sentiments. For example, "I love this!" and "I love this." might have different sentiment due to the presence of exclamation marks and periods.
5. **Lemmatization:** The process of reducing words to its root form is known as Lemmatization. Lemmatization is applied to capture the accurate semantics of a word. For instance, "running" and "ran" may be reduced to "run."

Our Lexical analyzer performs the tasks that are given below.

1. **URLs:** Twitter users engage in the platform not only to express their views but also to disseminate valuable information among others. One common method of sharing information involves the inclusion of links within tweets. These links, often in the form of URLs such as http://plurk.com/p/116r50, do not contribute to determining the sentiment of the tweet. Consequently, during the pre-processing stage, these URLs are removed to ensure they do not influence the sentiment analysis process.
2. **Usernames:** Within tweets, the "@" symbol is employed as a reference to mention or direct a tweet to other individuals. However, these references do not contribute to the sentiment analysis process. Hence, as part of the pre-processing stage, these references are removed to ensure they do not affect the sentiment analysis.
3. **Duplicates or repeated characters:** Twitter users often employ casual language and may use words in altered forms. For instance, the word "happy" might appear as "haaaaaaappy." Despite the variation in spelling, the underlying sentiment remains the same. To address this issue, the pre-processing stage involves the removal of duplicates and repeated words. These instances are then replaced with the correct form of the word to ensure accurate sentiment analysis, as shown in Table 1.
4. **Emotions:** Within tweets, various emoticons and emotion symbols such as 😊 and ☹ are frequently utilized. However, these symbols do not directly contribute to sentiment analysis. As a result, during the pre-processing stage, they are replaced with corresponding sentiment labels to ensure their influence on sentiment determination is accounted for. For example, "😊" replaced with "Happy," while "☹" replaced with "Sad." This substitution allows for a more accurate assessment of sentiment within the tweet content, as shown in Table 1.
5. **Stop-words removal:** In the context of information retrieval, numerous words serve as conjunctions within sentences. The stop-words examples include "the", "and", "before" and "while" etc. These conjunctive words don't have the significant impact on the sentiments of the tweets. Also, the stop-words don't help in the classification of all the classes of tweets.
6. **Spellchecker:** In case of misspelled words found in tweets. To address this issue pre-processing stage involves the correction of spelling for a more accurate assessment of sentiment within the tweets content.

3.3. Use of Dictionaries

Preprocessing procedures also make use of dictionaries in text documents.

1. Slang.txt: Used to detect and replace slangs.
2. Affin.txt: Used to detect and replace emotion with word.

3. Wordlist.txt: Used to detect hashtags and arrange in sequence with words space.

Table 1: Tweets preprocessing

Tweets Containing Words	Replaced By
Coooooooooool	Cool
Baaaaaad	Bad
😊	Happy
☹	Sad
#ILovePakistan	I love Pakistan
Lol	Lot of laughs
Apparent	Apparent

3.4. Tokens Priority Calculator

Tweets are tokenized, converted into n-grams including unigrams, bigrams, and trigrams. Words have the capacity to possess the different meanings according to their position in the sentence. In certain cases, combination of words offers more precise meanings as compared to individual words meaning. By using n-gram deeper understanding of sentence can be achieved. N-grams has numerous applications in text mining and natural language processing tasks, enabling more comprehensive analysis and interpretation of textual data.

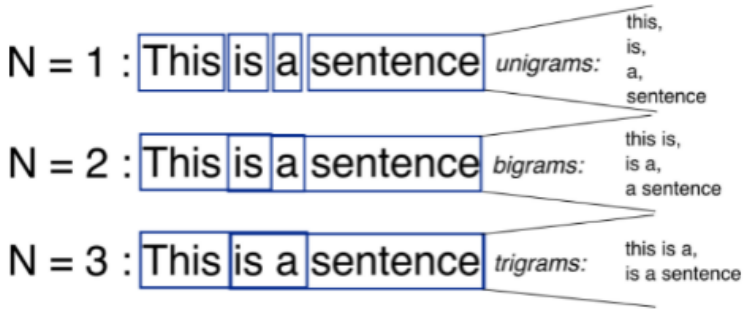

Figure 2: n-grams

The classification accuracy of tweets can be increased using n-grams as compare to single word feature. For the lexical analysis of the Tweets the three approaches can be adopted:

3.4.1. Manual Approach

The sentiment of words is based on the language understanding of a specific domain. This manual approach needs a human intervention that can be time-consuming. Typically, this approach is combined with automated approaches to streamline the sentiment analysis process. By combining manual and automated techniques, a more efficient and effective sentiment analysis methodology can be achieved.

3.4.2. Dictionary-Based Approach

The proposed technique of this paper uses the online dictionaries to determine the orientation of words. This approach does not require any specific domain knowledge in order to work. These dictionaries employ various techniques, such as synonyms, antonyms, and word hierarchies, to determine the sentiment of the word. However, context-specific sentiment can be challenging to identify using only a dictionary-based

approach. Popular online dictionaries utilized in this approach include WordNet, SentiWordNet, secticNet, and SentiFul, among others. These dictionaries serve as valuable resources in the process of sentiment analysis for determining the sentiment polarity of words.

3.4.3. Corpus-Based Approach

This approach takes into consideration the syntactic patterns and co-occurrence of words in the text along with their sentiment. This approach addresses some limitations of the dictionary-based approach. It relies on labelled data for training and analysis purposes. However, both the corpus-based approach and the manual approach are generally considered to be less efficient as compared to the dictionary-based approach. Nevertheless, the corpus-based approach offers more advantage of capturing contextual nuances and patterns that may not be readily apparent in dictionary-based methods.

4. SentiWordNet

Different tools TextBlob, Vader and SentiWordNet are used to perform lexicon-based analysis. In proposed work for n-gram analysis SentiWordNet is a lexical resource specifically designed to assist in Sentiment Analysis applications. It offers annotations in the form of three numerical sentiment scores (positivity, negativity, neutrality) for each synset within WordNet. These sentiment scores are the valuable indicators of the sentiment polarity associated with the corresponding synsets. This approach facilitates in getting the more accurate sentiment analysis of textual data. It has vast vocabulary that is easily adaptable to a variety of domain SentiWordnet's lexical relation makes n-gram analysis easier to handle. Words are mapped to synsets using Pos Tagging Integration, which increased the accuracy of the study.

Conversely, however TextBlob and Vader are two more lexicon-based programs that offer a single polarity score. Their vocabulary are limited, and they are unable to apply it for a variety of topics. These tools do not offer deeper analysis for n-grams since Vader is used for social media bias and produces good results for brief expressions, while TextBlob is the most basic tool that treats words in isolation.

In proposed work after converting the tweet to n-gram, the lexicon-based resource SentiWordNet is employed to determine the scores of the n-gram tokens. By using SentiWordNet, the sentiment scores n-gram tokens can be obtained that allowing for a more comprehensive analysis of the sentiment tweets.

5. Tweet Polarity Calculator

The tweet polarity is computed by using scores of unigrams, bigrams, and trigrams for each tweet. Additionally, assigned weights are applied to tweet objects such as favcount, likes, and hashtags.

The tweet polarity of each n-gram group is determined using the following methodology:

5.1. Unigrams

In Unigrams, the tweet polarity is calculated by adding the SentiWordNet scores for the unigram words. SentiWordNet gives us the (pos, neg) values for each phrase score. We must determine whether the sentence total emotion score is positive or negative. We utilized equation_1 and equation_2 to determine the *Average UniScore*. A CSV (Comma Separated Value) file is created by calculating and storing the average score. This makes it possible to store the sentence polarity scores of each unique unigram in an organized manner and analyze them further.

- Pos = positive value
- Neg = negative value
- Uni = unigrams
- Tokens = unigrams words
- Equation_1. Calculate the total score of sentences for both positive and negative.
- Equation_2. Used to determine the average score of unigram words on a positive-negative scale; sentence polarity is positive if the average score value is greater than 0 or negative otherwise.

$$Uni_{Sum(pos,neg)} = \sum_{n=1}^{n} Word_1\,(pos, neg) + Word_2(pos, neg) + \cdots + Word_n(pos, neg) \quad (1)$$

$$Average\ UniScore = \begin{cases} \frac{\sum_{n=1}^{n} Word_n(pos)}{Total\ no.\ of\ tokens}, & if\ Uni\_\mathrm{Sum}(pos) > 0 \\ \frac{\sum_{n=1}^{n} Word_n(neg)}{Total\ no.\ of\ tokens}, & otherwise \end{cases} \quad (2)$$

5.2. Bigrams

In Bigrams, the tweet text is transformed into groups of two co-occurring words. Consequently, the sentence polarity is computed by taking the product of the scores of the two words in each bigram. The scores of all the bigram groups with multiplied word scores are then summed. After summing the scores, the average of the resulting sum is calculated. The computed average value is then stored in a CSV file. This facilitates organized storage and subsequent analysis of the tweet polarity scores associated with the bigrams.

- Equation_3. Multiply two-word groups that occur together, then execute the summation to determine the overall score of sentences for both positive and negative.
- Equation_4. Used to determine the average score of bigrams words on a positive-negative scale; sentence polarity is positive if the average score value is greater than 0 or negative otherwise.
- Tokens = Bigrams words
- Bi = Bigrams

$$Bi_{Sum(pos,neg)} = \sum_{n=1}^{n}\big(Word_1(pos, neg) * Word_2(pos, neg)\big) + \cdots + (Word_n(pos, neg) * Word_{n-1}(pos, neg)) \quad (3)$$

$$Average\ BiScore = \begin{cases} \frac{\sum_{n=1}^{n} Word_n(pos)}{Total\ no.\ of\ tokens}, & if\ Bi\_\mathrm{Sum}(pos) > 0 \\ \frac{\sum_{n=1}^{n} Word_n(neg)}{Total\ no.\ of\ tokens}, & otherwise \end{cases} \quad (4)$$

5.3. Trigrams

In the case of **Trigrams**, the tweet text is converted into groups of three co-occurring words. The sentence polarity in trigrams is determined by taking the product of the scores of the three words within each trigram. The scores of all the trigram groups with multiplied word scores are then summed. Once the scores are added together, the average of the resulting sum is calculated. This average value is then stored in a CSV (Comma Separated Value) file format, allowing for organized storage and subsequent analysis of the tweet polarity scores associated with the trigrams.

- Equation_5. Multiply three-word groups that occur together, then execute the summation to determine the overall score of sentences for both positive and negative.
- Equation_6. Used to determine the average score of trigrams words on a positive-negative scale; sentence polarity is positive if the average score value is greater than 0 or negative otherwise.
- Tokens = trigrams words
- Tri = Trigrams

$$Tri\_Sum(pos, neg) = \sum_{n=1}^{n}\big(Word_1(pos, neg) * Word_2(pos, neg) * Word_3(pos, neg)\big) + \cdots + (Word_n(pos, neg) * Word_{n-2}(pos, neg) * Word_{n-1}(pos, neg)) \quad (5)$$

$$Average\ BiScore = \begin{cases} \frac{\sum_{n=1}^{n} Word_n(pos)}{Total\ no.\ of\ tokens}, & if\ Tri\_\text{Sum}(pos) > 0 \\ \frac{\sum_{n=1}^{n} Word_n(neg)}{Total\ no.\ of\ tokens}, & otherwise \end{cases} \quad (6)$$

After calculating tweet polarity by using the above formulas stored them in a CSV file and showed the results in a graph and tabular form.

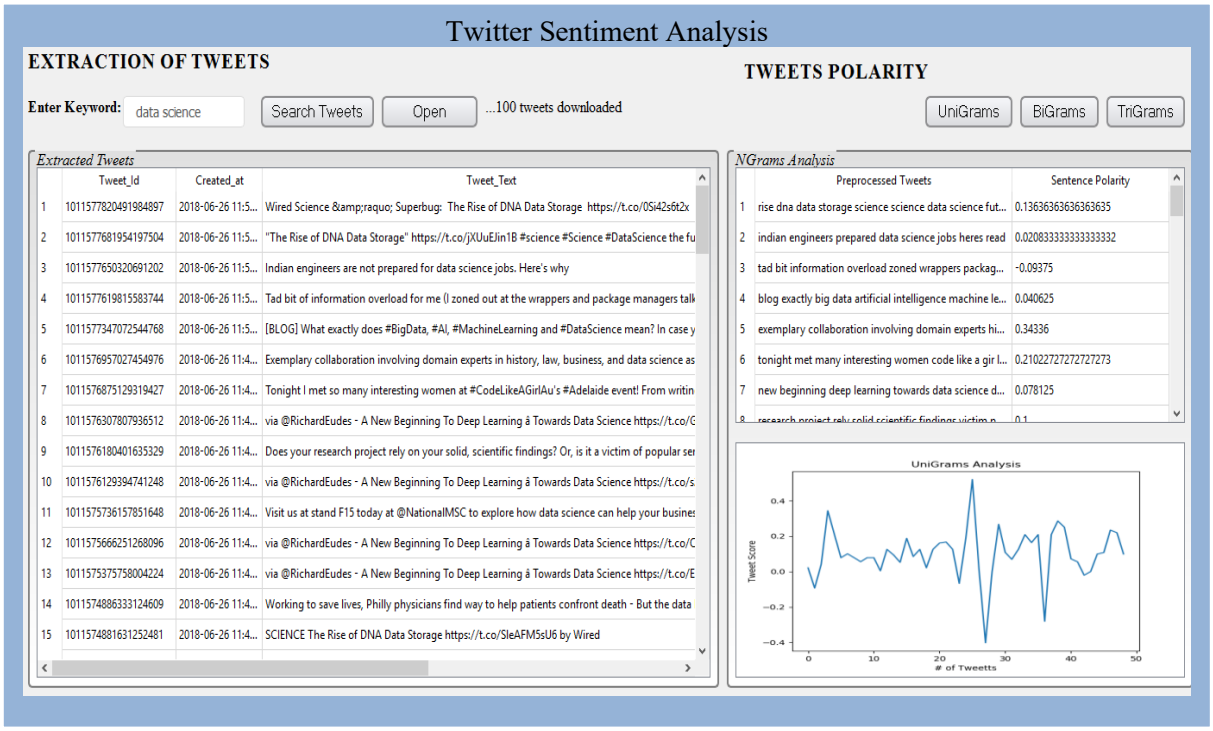

Figure 3: Interface of Proposed Work

6. Implementation

The proposed system requires the proper download and installation of the following components:

1. **Python 2.6 or above:** Python programming language should be installed and configured correctly in the desired location.
2. **Tweepy library:** This library is used to gather the tweets from Twitter by using Twitter API. It is a wrapper that provides easy retrieving and managing of the Twitter data [11].
3. **NumPy:** This library provides easy operations on multi-dimensional arrays and large mathematical functions [11].
4. **NLTK (Natural Language Toolkit):** This toolkit provides wide variety of tools to work with textual data. It also provides all the text processing libraires like classification, tokenization, stemming, parsing and semantic reasoning and Dictionary like WordNet.[11]

6.1. Pandas library

This library provides fast and flexible management of large labelled or unlabeled data in the form of a Dataframe. It is a versatile library for data analysis and manipulation of real-world data tasks.

1. **CSV (Comma Separated Value) library:** This library helps in reading and writing the tabular data in csv files.
2. **RE:** This module provides the support for regular expression matching operations. It allows for the matching of patterns against Unicode strings and 8-bit strings.
3. **Extract Data:** Tweepy is the open-source library that is used to extract the tweets from twitter.
4. **Twitter Posts:** The tweets are extracted and stored in CSV file.

6.2. Pre-processing

The extracted tweets are initially in an unstructured form, referred to as noisy data, which contains irrelevant information that cannot be used in the Sentiment Analysis (SA) process. Therefore, pre-processing of the tweets is performed to clean and refine the data by removing unwanted elements such as common words, stop words, symbols, extra spaces, special characters, and numbers. Additionally, the uppercase letters are converted to lowercase to ensure consistency in the data.

In the proposed system, the Python standard library is utilized for performing the pre-processing steps. Some of the key steps involved are as follows:

- Converting uppercase letters to lowercase: This step ensures uniformity in the text data, treating uppercase and lowercase letters as the same.
- Filtering URLs: URLs can be filtered using regular expressions such as [a-zA-Z0-9\./] +, which matches the URL pattern. The URLs are then replaced with the term "URL" to eliminate their influence on sentiment analysis.
- Removing user references (@): User references in the form of "@username" can be removed using regular expressions, specifically @(\w+), which matches the "@" symbol followed by one or more-word characters.
- Removing hashtags (#): Hashtags, denoted by the "#" symbol, can be removed using regular expressions, such as #(\w+), which matches the "#" symbol followed by one or more-word characters.
- Removing repeated characters: In colloquial language, words with repeated characters, such as "I'm in a hurrryyyyy," can be cleaned by using regular expressions. For example, (.)\1Error! Bookmark not defined. matches any character followed by one or more repetitions of the same character, and \1\1 is used to replace the repeating characters with a single instance.

The pro-processing steps are performed to clean the tweets data and used this clean data for further analysis.

6.3. n-grams

In the proposed system, three approaches of n-gram (unigrams, bigrams, and trigrams) are used for experimentation. Each tweet text is converted into these n-gram groups to capture different levels of linguistic context.

- Unigrams: In this step, the tweet text is tokenized into individual words and forming the unigrams. Each word in the tweet represents a single unigram. This approach allows us to analyse the sentiment associated with each word in the tweet.
- Bigrams: The tweet text is further processed to identify co-occurring pairs of words, known as bigrams. By grouping words in pairs, we can capture more contextual information and understand the sentiment that arises from word combinations. For example, "good movie" or "happy birthday" would be considered as bigrams.
- Trigrams: To get deeper into the linguistic context, the tweet text is transformed into trigrams,

which consist of three consecutive words. Trigrams helps to capture more complex relationships between words in a sentence and provide a higher level of context for sentiment analysis. Examples of trigrams could include "sky is blue" or "great customer service."

After converting the tweet text into n-gram, the proposed system gets different levels of linguistic information to better analyze the sentiment expressed in the tweets. Each type of n-gram allows for a more understanding of the text, leading to improved sentiment classification and analysis results.

6.4. Tweet Polarity

After converting the tweet text into n-gram, the score is assigned to each word by using the SentiWordNet dictionary. After scoring proposed system calculates the polarity using the SentiWordNet dictionary. The polarity of the tweet depends on both positive and negative scores. The positive and negative scores are added up for all the n-grams tokens in the tweet. The final polarity score represents sentiment expressed in the tweet.

The final polarity score has the range of [-1, 1]. The polarity score after zero (inclusive) is considered positive. Conversely, if the score is negative, it is classified as negative sentiment. For example, tweet: "I absolutely loved the movie! The acting was phenomenal."

6.4.1. Unigrams

Tokenized unigrams: ["i", "absolutely", "loved", "the", "movie", "The", "acting", "was", "phenomenal"]

The positive and negative scores of each unigram are obtained from the SentiWordNet dictionary after applying the equation_1and equation_2.

6.4.2. Bigrams

Tokenized bigrams: ["i absolutely", "absolutely loved", "loved the", "the movie", "movie acting", "acting was", "was phenomenal"]. The positive and negative scores of each bigram are obtained from the SentiWordNet dictionary after applying the equation_3 and equation_4.

6.4.3. Trigrams

Tokenized trigrams: ["i absolutely loved", "absolutely loved the", "loved the movie", "the movie acting", “movie acting was”, "acting was phenomenal"]

The positive and negative scores of each trigram are obtained from the SentiWordNet dictionary after applying the equation_5 and equation_6.

6.5. Unigrams Scoring

- **Tweet:** I like Foxy car 😊….!!!! https://www.pakwheels.com.
- **Preprocessed:** detected emotion symbol and convert into text. Removed stop words and URL.
- **Unigrams Tokenization:** “i”, “like”, “foxy”, “car”, “happy”
- **Calculated Polarity:** I: positive :0.01 negative:0.0, like: positive :0.2 negative: 0.01, foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01, happy: positive:0.125 negative:0.01
- Unigrams Sentence Polarity=0.24
- **Added Assigned Weight:** In the proposed system, the polarity calculation for unigrams takes into account the occurrence of emotions, the tweet length, and the total number of likes. Emotions are weighted and added to the polarity score, the tweet length is multiplied by its weight and added, and the total number of likes is also incorporated. Hashtags, however, are not considered in the polarity calculation.

6.6. Bigrams Scoring

In the case of bigrams, the tweet text is divided into pairs of words. The polarity score is obtained for each word using SentiWordNet (SWN). Then, the scores of each word pair are multiplied together and added up to calculate the sentence polarity. If the final score is greater than '0', it is considered positive; otherwise, it is considered negative.

- **Tweet:** I like Foxy car 😊….!!!! https://www.pakwheels.com
- **Preprocessed:** detected emotion symbol and convert into text. Removed stop words and URL.
- **Bigrams Tokenization:** “i like", "like foxy", "foxy car", "car happy"
- **Calculated Polarity:** 1st step (i: positive :0.01 negative:0.0, like: positive:0.2 negative: 0.01), (like: positive:0.2 negative: 0.01, foxy: positive:0.25 negative:0.0) (foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01) (car: positive:0.1 negative:0.01 happy: positive:0.125 negative:0.01)
- 2nd Step Multiply the positive score with a positive and negative score with a negative in every bigram group. It shows the relation between two co-occurring words.
- (I like: positive :0.03, negative:0.0), (like foxy: positive:0.05, negative: 0.0) (foxy car: positive:0.025, negative:0.0) (car happy: positive:0.0125, negative:0.0001)
- Add all of them and take the average for sentence polarity.
- Bigrams Sentence Polarity=0.145
- **Added Assigned Weight:** Emotion count, tweet length, and total number of likes are taken into account. These counts are multiplied by their respective weights and added to the polarity score of bigrams. If there are no hashtags in the example, the hashtag count is considered as '0'.

6.7. Trigrams Scoring

In the case of trigrams, there are three groups of words. The polarity calculation involves multiplying the scores of each word group obtained from SentiWordNet and then adding them together to determine the sentence polarity. If the final score is greater than '0', it is considered positive; otherwise, it is considered negative.

- **Tweet:** I like Foxy car 😊….!!!! https://www.pakwheels.com.
- **Preprocessed:** detected emotion symbol and convert into text. Removed stop words and URL.
- **Trigrams Tokenization:** “i like foxy", " like foxy car", "foxy car happy"
- **Calculated Polarity:** Step 1 (i: positive :0.01 negative:0.0, like: positive:0.2 negative: 0.01, foxy: positive:0.25 negative:0.0), (like: positive:0.2 negative: 0.01, foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01) (foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01 happy: positive:0.125 negative:0.01)
- **Step 2:** Multiply the positive score with the positive and negative score with the negative in every trigram group. It shows the relation of three co-occurring words.
  (i like foxy: pos :0.03, neg:0.0), (like foxy car: pos:0.05, neg: 0.0) (foxy car happy: pos:0.025, neg:0.0)
- After adding a score, calculated the average of tweet polarity.
- Trigrams Tweet Polarity=0.002
- **Added Assigned Weight:** The proposed system uses emotions, tweet’s length, and the number of likes to compute the polarity score. The system counts the number of emotions in each tweet, multiplies it by the assigned weight, and adds it to the trigrams polarity score. Similarly, tweet’s length and the number of likes is multiplied by their respective weights and added to the trigram’s polarity score. In the given example, since there are no hashtags, their count is considered as '0'.

7. Results

7.1. Survey Results

To evaluate the efficiency of the proposed work, a survey was conducted on Twitter datasets. Dataset was consisting on 50 tweets. A random tweets list was generated, no. of 100 graduate college students and teachers' opinion are taken regarding the tweets were positive, negative or neutral. Tweets were presented them in printed document. Performa was design that was consist on 3 columns. *Sr no.*, *Tweet*, *Sentiment Score* (Pos, Neg, Neu). Participant tick on one option (pos, neg, neu) after reading the tweets. After getting result we performed n-gram (unigram, bigram, trigram) analysis of those tweets in proposed system and compared it with survey result.

Table 2: Survey Results

Tweets	Peoples Opinion (Survey Result)	Unigrams	Bigrams	Trigrams
Tweet_1	Pos (75%), Neg (0%), Neu (25%)	Pos	Pos	Pos
Tweet_2	Pos (60%), Neg (0%), Neu (40%)	Neg	Pos	Neu
Tweet_3	Pos (40%), Neg (50%), Neu (10%)	Neg	Neg	Neu
Tweet_4	Pos (70%), Neg (5%), Neu (25%)	Pos	Pos	Pos
…				
Tweet_50	Pos (70%), Neg (5%), Neu (25%)	Pos	Pos	Pos

7.2. Comparison of Results

From the survey results, it was found that 75% of the participants expressed a positive opinion about tweet_1, while 25% considered it to be neutral. None of the participants had a negative opinion about the tweet. In the proposed work, the sentiment analysis results for tweet_1 were as follows: the unigrams analysis indicated a positive sentiment, while both the bigrams and trigrams analyses yielded neutral sentiments. In rest of tweets 41% results match with unigram, 44% results match with bigrams and 36% results match with trigrams out of 50 tweets.

7.3. Accuracy of Results

From the survey results, the accuracy of the proposed work is calculated by using an Accuracy formula.

$$Accuracy = Total\ no.of\ correct\ queries / Total\ no.of\ queries \quad (7)$$

Unigrams = 80% Bigrams = 88% Trigrams= 72%

Comparison of survey results with proposed system bigrams > unigrams and trigrams. It means bigrams produce more accurate result.

8. Discussion

The sentiment analysis results using unigrams (single words), bigrams (pairs of words), and trigrams (three words) show some interesting findings about how well these different methods work. The results suggest that bigrams are more effective than unigrams and trigrams. This is because bigrams look at pairs

of words together, which helps to better understand the context and handle things like negations. By considering two words at a time, the meaning of the sentence is clearer, leading to more accurate sentiment analysis.

The trigram analysis, which looks at groups of three words, didn't show clear results because all the sentiment scores were '0'. This suggests that using three-word combinations might not be as helpful in this case. Trigrams add more complexity but don't improve sentiment analysis much in this context.

From these results, we can conclude that both unigrams (single words) and bigrams (pairs of words) are good choices for analyzing opinions in text. Using both methods together is especially effective for understanding sentiments, as it improves the accuracy and efficiency of the analysis in the system being studied.

To check efficiency different twitter datasets were used and found good results.

Table 3: Comparison with different twitter datasets.

Ref	Technique	Analysis Feature	Accuracy
[7]	Supervised and Unsupervised Learning	Evaluate the performance of existing technique	ML=86.40 DeepLearning=80.70 LexiconBased=74.00
[8]	Machine Learning	Unigrams, Bigrams, Trigrams	Bigrams>Unigrams and Trigrams
[16]	Lexicon Based and Machine Learning	Bag of Words, N-grams	83.15 on n-grams with combinatory approach
[17]	Hybrid	Unigrams, Bigrams, Trigrams	Unigrams = 63.23 Bigrams= 62.33 Trigrams=59.98
[18]	Lexicon Based	Stemming, Emoticon Detection and Normalization, Exaggerated word shortening and Hashtags Detection	0.800 F-Score
Proposed System	Lexicon Based (Used SentiWordNet)	Emotion Detection, HT's Detection, Slangs Detection Spelling Correction Uni, Bi, Tri	Unigrams= 80 Bigrams= 88 Trigrams= 72

9. Future work

Sentiment Analysis on Twitter data is very tricky task. In our work, we discovered that how to understand the semantic of meaning and improve the accuracy of polarity on unlabeled data. Preprocessing phase is playing very vital role in existing work because, if preprocessing done good then calculation of polarity will give good results.

There are many factors that can be participate for future work like detection of Sarcasm, ruled based negation, extraction of useful video links and stemming can be improve the sentiment results more refine and useful. However, SentiWordNet has a limited ability to identify sarcasm. In the future, we can combine Vadar and SentiWordNet in a hybrid mode to deal with sarcasm and rule-based negation.

Author Contributions

Muhammad Sanaullah examined and proposed the experiments, conducted it, designed proposed work interface, evaluated the data, designed the mathematical computations, formatted the figures and/or tables, composed or revised article drafts and approved the final draft.

Rabea Saleem examined and proposed the experiments, conducted it, performed the mathematical computations, developed interface of proposed work, composed or revised article drafts.

Fatima Riaz examined and proposed the experiments, conducted it, evaluated the data, evaluated the mathematical computations, performed testing, formatted the figures and/or tables, composed or revised article drafts.

Funding Statement: The authors received no funding to conduct this study.

Conflicts of Interest: Authors declare that no conflicts of interest exist regarding this study.

Data Availability: The data supporting this study were collected in real-time from Twitter using the Twitter API and are available upon reasonable request, subject to Twitter's data sharing policies.

References

[1] B. Shelke, Mahesh, Daivat D. Sawant, Chatrabhuj B. Kadam, Kailas Ambhure, and Sachin N. Deshmukh. "Marathi SentiWordNet: A lexical resource for sentiment analysis of Marathi." *Concurrency and Computation: Practice and Experience* 35, no. 2 (2023): e7497.

[2] Kaur, Ravneet, Ayush Majumdar, Priya Sharma, and Bhavana Tiple. "Analysis of tweets with emoticons for sentiment detection using classification techniques." In *International Conference on Distributed Computing and Intelligent Technology*, pp. 208-223. Cham: Springer Nature Switzerland, 2023.

[3] Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In *Lrec*, vol. 10, no. 2010, pp. 2200-2204. 2010.

[4] Kristiyanti, Dinar Ajeng, Dwi Andini Putri, Elly Indrayuni, Acmad Nurhadi, and Akhmad Hairul Umam. "Twitter sentiment analysis using support vector machine and deep learning model in e-learning implementation during the Covid-19 outbreak." In *2ND INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SCIENTIFIC DEVELOPMENT (ICAISD) 2021: Innovating Scientific Learning for Deep Communication*, vol. 2714, no. 1, p. 020033. AIP Publishing LLC, 2023.

[5] Psomakelis, Evangelos, Konstantinos Tserpes, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. "Comparing methods for twitter sentiment analysis." *arXiv preprint arXiv:1505.02973* (2015).

[6] Lalji, T., and Sachin Deshmukh. "Twitter sentiment analysis using hybrid approach." *International Research Journal of Engineering and Technology* 3, no. 6 (2016): 2887-2890.

[7] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In *LREc*, vol. 10, no. 2010, pp. 1320-1326. 2010.

[8] Saif, Hassan. *Semantic sentiment analysis of microblogs*. Open University (United Kingdom), 2015.

[9] Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." In *International semantic web conference*, pp. 508-524. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[10] Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!." In *Proceedings of the international AAAI conference on web and social media*, vol. 5, no. 1, pp. 538-541. 2011.

[11] Palanisamy, Prabu, Vineet Yadav, and Harsha Elchuri. "Serendio: Simple and Practical lexicon based approach to Sentiment Analysis." In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 543-548. 2013.

[12] Mutinda, James, Waweru Mwangi, and George Okeyo. "Sentiment analysis of text reviews using lexicon-enhanced bert embedding (LeBERT) model with convolutional neural network." *Applied Sciences* 13, no. 3 (2023): 1445.

[13] Aslan, Serpil, Soner Kızıloluk, and Eser Sert. "TSA-CNN-AOA: Twitter sentiment analysis using CNN optimized via arithmetic optimization algorithm." *Neural Computing and Applications* 35, no. 14 (2023): 10311-10328.

[14] Aljedaani, Wajdi, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry." *Knowledge-Based Systems* 255 (2022): 109780.

[15] Farah, Hassan Abdirahman, and Arzu Gorgulu Kakisim. "Enhancing lexicon based sentiment analysis using n-gram approach." In *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*, pp. 213-221. Cham: Springer International Publishing, 2021.

[16] Kumar, Shachi H. "Twitter Sentiment Analysis‖." *medium. com, https://medium. com/analytics-vidhya/twitter-sentiment-analysisb9a12dbb2043 (Last access Nov. 22, 2022)* (2014).

[17] Psomakelis, Evangelos, Konstantinos Tserpes, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. "Comparing methods for twitter sentiment analysis." *arXiv preprint arXiv:1505.02973* (2015).

[18] Lalji, T., and Sachin Deshmukh. "Twitter sentiment analysis using hybrid approach." *International Research Journal of Engineering and Technology* 3, no. 6 (2016): 2887-2890.