International Journal of Advanced Technology and Engineering Exploration, Vol 10(99)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.875886

Research Article

Improvisation in opinion mining using data preprocessing techniques based on consumer's review

Kartika Makkar[1*], Pardeep Kumar[1], Monika Poriye[1] and Shalini Aggarwal[2]
Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, India[1]
Department of Computer Science S.U.S. Govt. College, Matak Majri (Indri), Karnal, India[2]

Received: 30-April-2022; Revised: 05-February-2023; Accepted: 08-February-2023

Abstract
In today's digital age, an enormous volume of data is generated daily from various internet sources, including social media sites, emails, and consumer reviews. With competition on the rise, it has become essential for organizations to understand their customers' needs and preferences. To gain meaningful insights from human language data, such as reviews, and understand consumer perceptions, sentiment analysis is an effective method. This research article presents a text preprocessing approach consisting of three stages: data collection, cleaning, and transformation. The approach was applied to three datasets - restaurant, cell phone, and garments - and evaluated using various machine learning classifiers for sentiment prediction. A comparison was made between two sets of techniques: set1 employed data cleaning and transformation with stemming, while set2 used data cleaning and transformation with lemmatization. The results indicated that set2 (data cleaning, transformation with lemmatization) performed better during preprocessing when evaluated using various machine learning classifiers, such as support vector machine (SVM), logistic regression (LR), decision tree (DT), random forest (RF), and Naïve Bayes (NB). Specifically, SVM, LR, RF, and NB performed better for the restaurant dataset, while DT, LR, and RF outperformed for the cell phone dataset. In the garment's dataset, LR, DT, and RF outperformed for set2 compared to set1, making set2 the best preprocessing technique for subsequent comparison. Additionally, another comparison was made between two sets of techniques: set3 included text cleaning, transformation with lemmatization, and unigram features, while the other set included text cleaning, transformation with lemmatization, and bigram features. The sets were evaluated using machine learning classifiers, and the results revealed that set3 performed better with most classifiers.

Keywords
Support vector machine (SVM), Random forest (RF), Decision tree (DT), Logistic regression (LR), Naïve bayes (NB).

1.Introduction
The consumer is one of the assets of every organization. In order to survive in this competitive marketplace, it is mandatory for every organization to assess the sentiments, expectations, and feedback of consumers. For this assessment, it is required to perform the sentiment analysis of data collected from various internet sources. The data collected from various internet sources contain various types of anomalies such as stop words, hypertext markup language (HTML) tags, misspellings, abbreviations, special characters, uniform resource locator (URL), etc. Due to this, it becomes difficult for both humans and machines to get the exact meaning of the sentence.

Moreover, the presence of this unwanted noise in data increases the dimensions because each word in the text is considered a separate feature and it becomes challenging for classifiers to classify this noisy data. So, to get accurate sentiments it is mandatory to remove this unwanted noise from the data. In previous studies, various preprocessing techniques were used for different types of datasets and all these techniques were evaluated to find the best preprocessing techniques using various classifiers such as support vector machine (SVM), logistic regression (LR), decision tree (DT), random forest (RF), Naïve bayes (NB).

Some of the basic preprocessing approaches like removal of stop words, punctuation, tokenization, spell correction [1–5] stemming/lemmatization were used in various pieces of research. Similarly, various

*Author for correspondence

preprocessing techniques were also proposed and a comparison of stemming and lemmatization was also performed for various other languages such as Bengali [6], Hindi [7], Sinhala [8], Gujarati [9], Punjabi [10], Icelandic [11–13], Indonesian [14], etc. So, by getting the motivation from previous studies [1–5] a set of preprocessing techniques is also proposed for our research work. From literature, it is observed that there is a comparison between stemming and lemmatization for various languages [6–14] rather than the English language. So, the main objective of this research is to propose a set of preprocessing techniques for the English language by including the impact of stemming and lemmatisation. Even from the literature, there is no clear distinction between data cleaning, and data transformation techniques, and as per our literature study, there are few papers in which there is a comparison ofstemming and lemmatization with various preprocessing techniques. Further, in this research article data cleaning (punctuation, special characters, URLs, HTML tags removal, case normalization) and data transformation (tokenization, stop word removal, spelling correction, parts of speech (POS) tagging, stemming or lemmatization) steps are implemented in three datasets and evaluated using classifiers. Post that a comparison of set1 (data cleaning, data transformation with stemming) and set2 (data cleaning, data transformation with lemmatization) is also performed to find the best preprocessing approach. Moreover, after getting the best preprocessing technique i.e., set2 (data cleaning, transformation with lemmatization) a comparison is also done using n-gram techniques in the feature extraction step. Here, a comparison of set1, and set2 with unigram and bigram features are done and evaluated using ML classifiers to make the data noise free and to understand consumer's perception correctly. The contributions of this research are the following.

The contribution of this paper is as under:

- A text preprocessing approach consisting of three stages was proposed in this research article.
- The performance of two sets of preprocessing techniques was compared using various machine learning classifiers, namely set1 (data cleaning, data transformation with stemming) and set2 (data cleaning, data transformation with lemmatization).
- Another comparison was performed between set3 (data cleaning, data transformation with lemmatization with unigram features) and set4 (data cleaning, data transformation with lemmatization with bigram features).
- The proposed approach was evaluated using SVM, DT, RF, LR, and NB classifiers with unigram and bigram features.

The article is divided into six sections: introduction, related studies, proposed methodology, experimental setup and results, discussions, conclusion, and future work.

2.Related study

This section describes the literature related to text preprocessing and the impacts of various text preprocessing techniques on classification.

A set of preprocessing techniques for tweets was introduced in a research article and the whole preprocessing process was divided into four steps. In the first phase, tweets are extracted from Twitter using the Twitter application programming interface (API). The second phase includes the removal of URLs, punctuations, hashtags, symbols, and emoticons. In the third phase spell corrector, tokenization, internet slang identification, lemmatization/Stemming, expanding acronyms, and stop words removal. In phase four the sentiments of the whole sentence are calculated. Finally, the results conclude that preprocessing helps to reduce the dimensions of data and the accuracy of classifiers is improved when preprocessing techniques are applied [2].

Different preprocessing techniques provide different classification accuracy on different datasets. According to this observation, the author demonstrated that there should be a proper combination of preprocessing techniques for different datasets. In this study different preprocessing techniques are applied to four datasets, for the first three datasets, only stop words removal was applied with a unigram bag of 1000 words and it provides significantly better results. However, improvement was not so good when more preprocessing techniques are applied like removal of HTML tags, spelling correction, reduction of replicated characters, and punctuation removal. Moreover, for the fourth dataset spelling correction and conversion to lowercase provides significant improvements in results. So, from the above discussion, it can be concluded that there was always a combination of preprocessing techniques that proved best on different datasets [3].
In fake review detection, preprocessing is the initial step which has a huge impact on the accuracy of

classifiers. According to this research, high dimensional data with bigram or trigram representation gives better performance but stemming and stopwords removal have less impact on the classification of fake reviews [4].

In [5] various stemming techniques and existing stemming algorithms for Indian languages was given. It also presents the performance of the stemming algorithms in applications like spelling checkers in various languages. A simple stemming algorithm removes suffixes by using a frequent list of suffixes but a complex one uses morphological knowledge to derive the base word or stem from the words. In order to get an accurate text classification, it is mandatory to get accurate root words of text data. So, to get these accurate root words stemming and lemmatization are two transformation techniques. Between stemming and lemmatization, lemmatization produces more meaningful root words as compared to stemming.

Preprocessing techniques have a huge impact on the accuracy of models therefore a text preprocessing toolkit (txtprep) was purposed which consists of various preprocessing techniques. Further, various text preprocessing techniques were also purposed and compared to improve the accuracy of destructive message classification problem [6–9].

A stemming algorithm for the Punjabi language was introduced based on the Brute force technique. In this approach, there is no need to apply preprocessing of text data before applying stemming. Here, different groups of words are created for stemming some groups have simple words, and some groups have complex words, and the accuracy of stemming depends on words present in groups and the size of the database. The results depict that the Brute force technique gives better accuracy as compared to rule-based and hybrid approaches [10]. Further, the impact of various preprocessing techniques on text classification was also studied for Arabic and Gujarati language and it is revealed the accuracy of sentiment analysis depends on preprocessing, feature extraction and classification techniques [11–13].

Stemming is also one of the techniques to get the dictionary form of words. Porter stemmer is one of the stemming algorithms for English language but there is no standard technique to find the root words of Punjabi language. Therefore, Brute force technique was introduced to find the root words of Punjabi language. This technique also helps to resolve the issue of over-stemming and under-stemming of Punjabi words [14].

Further, a new lemmatizer was developed for the Icelandic language. This lemmatizer gives the best performance by using IceTagger for doing tagging and for training the Icelandic frequency dictionary [15] was used. Here to maximize the performance, data-driven ML approach is combined with linguistic insights known as HOLI. In this approach, all the features of ML are organized using linguistic knowledge. Further, the accuracy of this model is improved to 99.55% by connecting it with the database of modern Icelandic inflection [16]. Before performing lemmatization, POS tagging is the initial step. For Icelandic text, there are various taggers available like trigrams‘n’tags (TNT), and maximum entropy POS tagger. All these taggers were trained on the corpus of the Icelandic frequency dictionary having 500,000 running words tagged with morphological tags and the results depict that the TNT tagger gives the best results with 90.36% accuracy [17].

Not only do text preprocessing techniques affect the accuracy of sentiment analysis, but misspelled words can also have an impact. To address this issue, the Levenshtein distance algorithm was proposed to detect and preprocess misspelled words in Indonesian text. The results showed an 8.2% increase in accuracy of sentiment analysis by using the Levenshtein distance algorithm for spelling correction [18, 19].

Nowadays big data and related technologies are emerging research areas. A huge amount of data generated from varied internet sources requires preprocessing steps before giving to ML classifiers. So, in this research article different preprocessing steps are applied to big data collected from twitter having 359 documents, and evaluated using NB, maximum entropy, and SVM. Here, firstly emoticons were removed then bigram techniques were applied. Post that, some more preprocessing techniques were applied like stopwords removal, stemming, and word vector. Among all these algorithms NB gives higher accuracy with an improvement of 8.12% [20].

A domain specific preprocessing technique was introduced to filter the irrelevant text. This preprocessing was introduced by defining multiple patterns of dependency trees and applied to retirement documents of United States. This technique helps to increase the f1-score by 16% [21].

The impact of various preprocessing techniques was also assessed on deep learning-based applications. In this research all the results are evaluated with or without preprocessing techniques. The results revealed that all the deep learning applications gives better results with preprocessing techniques [22].

The use of various preprocessing techniques also depends upon the data. Sometimes, even without applying any text preprocessing technique, the data obtained is accurate, and sometimes even after applying preprocessing techniques, the data is not so accurate. So, the selection of data preprocessing techniques also depends upon the type of data and should be used with caution [23].

Various ML algorithms were also used to classify various Arabic languages documents with or without various preprocessing techniques and it were revealed that classification accuracy gives better results with preprocessing techniques [24].

NLP is a branch of artificial intelligence (AI) that provides the ability for machines to understand and draw meaningful insights from human languages. NLP converts unstructured data into machine-readable form by using the attributes of natural language [25]. NLP follows various grammatical rules using different algorithms to derive meaning from text data. Lemmatization, tokenization, POS tagging, stemming, etc. are some of the most used algorithms in this context. NLP is identified as a challenging field reason that understanding the natural language not only requires the words to be understood but also how these words are connected to each other and produce a precise meaning. To handle the preprocessing of this text data NLP supports natural language toolkit (NLTK) which is a collection of various libraries, modules, etc. created in python. NLTK toolbox deals with tokenization [26], stemming, labeling and parsing, etc.

Spelling correction is one of the preprocessing techniques that are used to correct misspelled words in the text. One such approach was applied for the correction of misspelled words in the Indonesian language using the Levenshtein algorithm. This algorithm was applied to detect and preprocess misspelled words. Thereafter, preprocessed data is classified using multinomial NB and the results depict that the accuracy was increased by 8.2% after the application of this preprocessing technique [27]. Another research also demonstrates the evaluation of various preprocessing techniques on text classification. In this study, some preprocessing techniques are used on text data like tokenization, stopwords removal, and stemming. Moreover, in this study term frequency and inverse document frequency (TFIDF) with cosine similarity and chi-square are used for feature extraction. The results depict that the preprocessing techniques affect feature extraction and enhanced the classification accuracy using the TFIDF technique [28].

Combinations of various preprocessing techniques were also implemented on English news, English email and Turkish news, and Turkish email datasets. In this research combination of tokenization, stopwords removal, lower case conversion, and stemming were applied, and results are assessed by activating and deactivating various techniques. From the results, it was concluded that for different datasets various combinations should be tested before classification because there is no perfect combination of preprocessing techniques for every dataset [28–32]. One more combination that proved perfect for the Reuters dataset is stopwords removal, stemming, and TFIDF [33]. Another preprocessing set is applied to the Twitter dataset. Here, various combinations are applied like the basic (removal of hashtags, URLs, mentions, misspelled, more than three repeated vowels, blank space, and lower case)+ stemming, basic+ stopwords, basic+ negation, basic+ emoticon, basic+ dictionary, and all techniques revealed that basic + stemming steps gives the highest accuracy among all combinations [34].

A three-stage framework for sentiment analysis was developed that contains transformation, filtering, and classification. Initially, data transformation was performed which includes stopwords removal, abbreviation expansion, HTML tags removal, negation handling, and stemming. In data filtering chi-square is used for feature extraction then classification was performed using SVM. Finally, the results are evaluated with or without applying filtering and transformation techniques. It was revealed that the accuracy of SVM is more with transformation and filtering techniques [35–38].

Nowadays, fake review detection is also one of the crucial research areas due to the rapid increase of e-commerce websites. In order to correctly classify these fake reviews, it is mandatory to give preprocessed data to classifiers. So, in a research article, this task was accomplished using some preprocessing techniques such as tokenization, stopwords removal, stemming, feature

dimensionality, and different weighting scheme. In this research, it was revealed that the performance of classifiers is better when the data was represented in high dimensions using the bigram or trigram approach. Moreover, it was also depicted that stopwords removal and stemming are not so important to improve the accuracy of classifiers [39].
In another research, sentence retrieval was performed using the term TFIDF, and different preprocessing techniques were applied like stopwords removal, stemming, or lemmatization. And the results revealed that lemmatization provides better results for larger queries and stemming gives poor performance for larger queries [40].

A rule-based stemmer was also introduced for the Sinhala language using prefix and suffix rule. There are a number of preexisting stemmers but this rule-based stemmer is capable to generate correct root words [41].

A language-independent lemmatization algorithm was introduced in research that works with a RF classifier. This model was open source supervised ML model constructed using DT and grammatical features of the language. The advantage of this algorithm is that it can work with many languages and it can be extendable to some other languages [42].

A comparison of stemming and lemmatization was done for document retrieval and the result depicts that lemmatization gives better accuracy as compared to stemming [43]. Further, a comparison of three lemmatization approaches was performed for the Turkish language. The first lemmatization approach is based on a morphological analyzer and uses finite-state language processing. The second one is a dictionary-based lemmatizer that uses the radix-trie data structure. Another is a dictionary-based top-down parser and the last is a truncation of words at a fixed length. The results of this comparison shows that the dictionary-based Turkish lemmatizer which uses a radix-trie data structure gives better performance as compared to other lemmatizer for information retrieval system [44].

A comparison of stemmer and lemmatizer using handcrafted rules for Norwegian, Swedish and Danish languages were performed and the performance is same with 10% errors. The handcrafted rule based stemming approach is easy if developer has proper linguistic knowledge on the hand the lemmatization rule can be easily produced without linguistic knowledge provided the given training data is correct [45]. Further, a comparison of lemmatization and stemming was performed in the information retrieval of documents using clustering and the result depicts that lemmatization gives best performance as compared to stemming [46].

In order to overcome the over-stemming problem in stemming a partial lemmatization is hybridized with an unsupervised stemming algorithm that does not need any word class information. As per literature this approach has not been explored thus far and worked with the Hindi language. It uses a cluster-based approach and this algorithm overcomes the problem of determining the number of clusters. The results demonstrated that this approach proved significant in handling morphology [47].

In order to convert unstructured data to structured data various preprocessing techniques were used with bidirectional encoder representation from transformers (BERT) model with various deep neural networks (DNN) such as multilayer perceptron, long and short-term memory (LSTM), bidirectional LSTM (Bi-LSTM), gated recurrent unit (GRU), and convolutional neural network (CNN). From all these experiments it was revealed that BERT and CNN gave best classification accuracy [48].

From the above discussion, it can be concluded that there is very little discussion about text cleaning, text transformation, and comparison of these techniques, but these techniques have a huge impact on opinion mining. So, for efficient opinion mining, in this research article, the focus is on text cleaning, text transformation, and comparison of text transformation techniques with n-gram features to select the best technique for further research.

3.Methods

The proposed methodology starts with data collection of three datasets to find the best preprocessing technique. Initially, the data is collected from various sources as discussed in section 3.2. On this collected data, data cleaning is applied to remove all the anomalies present in data. Then, data transformation with stemming (set1) and lemmatisation (set2) is applied on cleaned data to get root words. Post that all these root words are converted to vector form using TFID technique and train test split is performed using different split ratios 70:30, 80:20. On this data different ML algorithms are applied as demonstrated in section 3.6. Further, set1, set2 of preprocessing are evaluated using various metrics as discussed in

section 3.7. In next step implementation and results shows that set2 of preprocessing gives best performance as compared to set1. Here, set2 is selected for further comparison using unigram, bigram features i.e., set3 (feature extraction using TFIDF with unigram) and set4 (feature extraction using TFIDF with bigram features). Then, train test split is performed on set3, 4 using different split ratios such as 80:20, 70:30. On this data classification is performed using various ML classifiers then the performance of these classifiers is evaluated. In final step, a comparison of results between set3, 4 is performed and from the results of comparison it is revealed that results of set3outperforms set4 therefore, we will use set3 for further research (*Figure 1*).

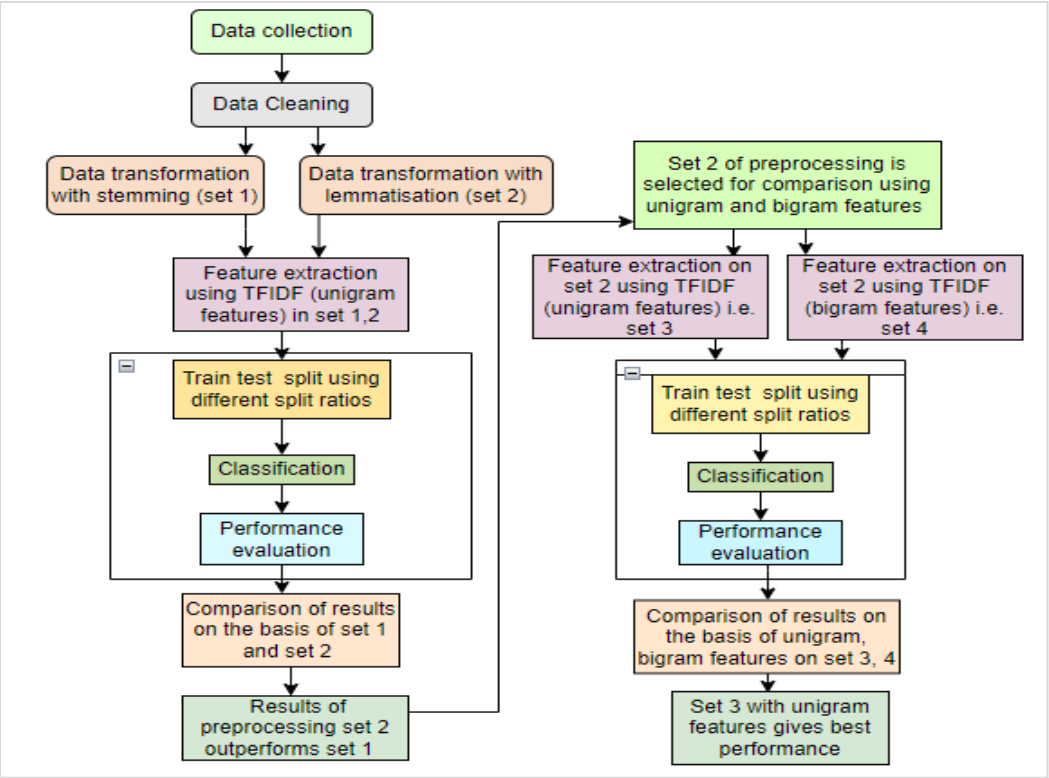

Figure 1 Proposed methodology for preprocessing

3.1 Data collection

Here, three datasets are used: the first is related to restaurant reviews, the second is related to garments, and the third is the cell phone reviews dataset collected from Kaggle, Amazon. Restaurant and garments datasets contain two variables i.e. review and polarity [48, 49]. Whereas the cell phone dataset contains various other attributes like reviews, overall ratings, polarity, helpfulness, timestamp, etc. Among all these attributes we have selected two attributes i.e., reviews and polarity. Here, both variables are used for sentiment prediction using various ML classifiers such as SVM, LR, DT, RF, and NB. Further, the reviews (x) attribute acts as an independent variable and the polarity attribute acts as the dependent (y) variable. The restaurant and garments datasets are bipolar datasets having 0, 1 polarity and the cell phone dataset is having three polarities i.e. -1, 0, 1. The restaurant dataset has 765 reviews, the garments dataset has 3000 reviews and the cell phone dataset has 2000 reviews. Here, all the implementation is performed by dividing datasets into 70:30, and 80:20 ratios. In case of 70:30 ratio, 70 is used for training, and 30 as test data. Whereas in case of 80:20 ratio, 80 is used for training, and 20 is used as test data. The collected raw data is having various types of noise and needs to be removed for accurate information retrieval from reviews.

3.2 Data cleaning

In this module, unwanted noise like punctuation marks, special characters, etc. are removed to ensure that the performance of a classifier is not degraded.

3.2.1 Punctuation removal

Punctuations present in the document are not helpful to perform text classification and these punctuations

can be removed using regular expressions. Some of the punctuations are '!"#$%&\'() * +,-./:;<=> ?@{|}[\\]~_'.

3.2.2 Case normalization

Case normalization is also an important phase in preprocessing because python is case sensitive language. So, all the text should be either in upper case or lower case and this text data can be converted to lower case or upper case by using str. lower() or str. upper() methods.

3.2.3 Removal of numbers

All the non-alphabetic characters don't have any contribution for opinion mining and are removed using data cleaning process.

3.2.4 HTML tags and URL removal

In this step all the HTML tags and URLs present in the data are removed.

3.3 Data transformation

After the removal of noise, the cleaned data needs to be converted according to some standards to apply feature engineering and classification. In this phase tokenization, stop words removal, POS tagging, and lemmatization are applied and results are shown in section 6, before and after the transformation of the dataset. Now, this data is ready for feature engineering and classification tasks to get consumers sentiments.

3.3.1Tokenization

It is defined as the process of splitting text data into small chunks or units known as tokens. Tokenization is important to step in NLP because it is necessary to determine those words that are having a string of characters and it also becomes easy to analyze the words in the text.

For example, the sentence

"This is a laptop" after tokenization becomes ['This', 'is', 'laptop'].

The NLTK contains a module called tokenize () which can be further categorized into two categories.

Word tokenization

A separate token is generated for each word in a document. It uses a method called word_tokenize () for tokenization.

Sentence tokenization

The whole document and paragraph is divided into sentences by using sent_tokenize () method.

3.3.2 Stop-word removal

These are commonly used words that do not provide any meaning to the text. For example, the, in, an, what is, with, etc. are commonly used stop words. These stop words reduce the accuracy of the classifiers while performing classification, sentiment analysis, etc. In the sentence "What is Machine learning", "machine learning " is more important as compared to "what is", and it also increases the dimensions of data. Thus, in order to overcome these flaws, there is a need to remove all these articles, pronouns, prepositions, etc. from the text data.

3.3.3 Spelling correction

Spelling mistake is a very common anomaly present in most of the datasets. These wrong spellings result into the misclassification of text data and degrade the performance of classifiers for opinion mining. So, to overcome this problem spelling correction has been performed.

Review in dataset with incorrect spelling

'honeslty taste fresh'

Review in dataset after applying spelling correction

'honestly taste fresh'

3.3.4POS tagging

It is a technique that assigns tags to the words in the text. It assigns the POS tags to various words according to context and particular POS tag [5] of a word shown in the sentence (*Figure 2*).

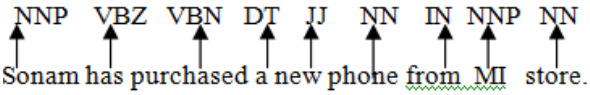

Figure 2 POS tags corresponding to each word

In the above sentence, stands for proper noun singular (NNP), verb 3rd person singular present (VBZ), verb past participle (VBN), determiner (DT), Adjectives (JJ), noun, singular or mass (NN), preposition (IN).

3.3.5 Normalization

Normalization is a technique that is used to convert various words into their root or base word called morpheme. This technique helps to reduce redundancy in the text, which also reduce the dimensions. Normalization can be done using following techniques.

(a)Stemming

Stemming is one of the normalization techniques that help to convert the tokens into their root or base form. For example, the token 'troubled' is converted into the base word 'trouble' after applying stemming. There are different stemming algorithms to perform normalization, but Porter's stemmer algorithm is one of the famous algorithms to perform stemming in English. The algorithm for stemming works on the basis of a crude heuristic approach that chop off the end parts of words for transforming into root words. But, sometimes it generates meaningless words due to chopping, for example, the word "increasing" is

converted into “increas” and lost the meaning of the word.

(b)Lemmatization

Lemmatization is another normalization technique that is different from stemming because performance of lemmatization ismore accurate by using vocabulary and morphological analysis of words. It chops off the inflections from the words and returns the base or dictionary form of the word called a lemma. It performs the full morphological analysis of the word to correctly identify the lemma and it also uses some rule-based approaches and a dictionary for mapping known as Wordnet. From the above discussion, it is clear that lemmatization and stemming are used for the normalization of dataset and lemmatization is a more appropriate technique for the normalization of text data. In this research article Wordnet, a lemmatizer has been used which is one of the most common and earliest lemmatizer present in the NLTK python library. It is a lexical database of over 200 languages and provides a semantic relationship between its words. *Table 1* show that words generated by stemming and lemmatization.

Table 1 Words generated by stemming and lemmatization

S. No.	Words	Stemming	Lemmatization
1	angry	angry	angry
2	honestly	honesti	honest
3	service	servic	service
4	tried	tri	try
5	pretty	pretty	pretty
6	cranberry	canberri	cranberry
7	overpriced	overpr	overprice
8	highly	highly	high
9	little	little	little
10	restaurant	restaur	restaurant

3.4 Feature extraction

Feature extraction is the transformation of text data into vector form, because ML classifiers can perform sentiment predictions only with numerical data. So, in this article feature extraction is applied using TFIDF technique. TFIDF is the combination of two terms TF and IDF here, TF represents number of times term t is present in the document against the number of times all the words appear in a document and IDF represent the weight of a word in a document [48].

$$TF = \frac{Total\ number\ of\ time\ a\ word\ present\ in\ document}{total\ terms\ in\ the\ document}$$

$$IDF = \log\left(\frac{number\ of\ documents\ in\ the\ corpus}{number\ of\ documents\ in\ the\ corpus\ contain\ the\ term}\right)$$

$$TF\text{-}IDF = TF \times IDF$$

Feature can be represented with n-gram (n=1, 2, 3, …) while using TFIDF

- **Unigram**: In unigram sentence is represented like ‘This’, ‘car’, ‘is’, ‘costly’.
- **Bigram**: In bigram sentence is represented like ‘This car’, ‘is costly’.

3.5 Classification

Here, various ML classifiers with k-fold cross validation technique are used for performance evaluation and are discussed below.

(a)Support vector machine

SVM is based on the principle of generating a hyperplane that separates the data points into different classes. It works with both linear and non-linear problems. It is a linear model which works with both regression and classification problems. Here, various hyperparameters used in SVM are C=1, Kernel =rbf, gamma= scale, C stands for penalty parameter of the error term, RBF stands for radial basis kernel, and gamma represents the decision region of the kernel [50].

(b)Logistic regression

LR works with sigmoid function and uses to solve classification problems. It predicts the dependent variable that can be either 0 or 1, yes or no, etc. Parameters used in LR are random_state=0, solver=‘liblinear’, multi class=’auto’, here the random state is the random state instance used np. random, multi-class is auto because data is binary, solver is liblinear use for small datasets [50].

(c)Decision tree

It also works with both regression and classification problems. It performs splitting based on some parameters here two entities are used decision nodes and leaves. The split operation is performed in

decision nodes and leaves represent the decision. Parameters used in DT are criteria = gini, splitter = best, max_depth = none here, criteria represents how to measure the impurity of split, splitter represents how DT searches splitting feature and max_depth represents maximum depth of the tree [50].

(d)Random forest

RF is also used for classification and regression problems it acts as an ensemble classifier that works with various DTs. All these DTs are trained on various subsets of datasets and the final output is taken as the majority vote of each DT [50].

(e)Naive bayes

NB is used for classification problems and predicts the output based on the probability of each element. Here, Bernoulli NB is used because data is binary and for binary text classification Bernoulli NB is best [50].

3.6 Evaluation

For evaluating the results of all the classifiers following parameters are used.

(a)Confusion matrix

A confusion matrix is used to measure the performance of ML classifiers. A confusion matrix is just like a table having two dimensions named actual and predicted and it consists of four values i.e., True Positive (TP),True Negative (TN), False Positive (FP), and False Negative (FN) displayed in *Figure 3*.

	Positive (1)	Negative (0)
Positive (1)	**TP**	**FP**
Negative (0)	**FN**	**TN**

Figure 3 Confusion matrix

True Positive

It is the case when the actual and predicted value is 1 or yes.

True Negative

It is the case when the actual and predicted value is 0 or no.

False Positive

It is the case when the predicted value is yes or 1 and actual value is no or 0. It is also known as type I error.

False Negative

It is the case when the predicted value is no or 0 and actual value is yes or 1. It is also known as type II error.

(b)Accuracy

Accuracy is defined as the observations predicted correctly divided by a total number of observations as shown in Equation 1.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

(b)Precision

Precision is just like a metric that is used to measure the exactness of a classifier as shown in Equation 2.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

(d)Recall

The recall is also just like a metric that is used to measure the sensitivity or completeness of a classifier as shown in Equation 3.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

(c)F-measure

F-measure is the combination of recall and precision and has a range of 0.0-1.0 where 1.0 is the perfect value. F-measure is the harmonic mean of recall and precision of a particular model as shown in Equation 4.

$$F - measure = \frac{2TP}{2TP+FP+FN} \quad (4)$$

4.Results

4.1Experimental setup

In this research whole experiment has been done on Google Colab (RAM 12 GB, disk space 25 GB, CPU Model Intel (R) Xeon (R), No. CPU Cores 2) using python 3.7, on three datasets as mentioned in *Figure 1*. Here, initially preprocessing is performed in three stages and various experiments are performed with different sets of preprocessing techniques on three datasets. Then feature extraction is performed using TFIDF with unigram, bigram features. Post that classification is performed using various ML classifiers mentioned in *Figure 1*. Further, these classifiers are evaluated using various metrics also mentioned in *Figure 1*. Here, k-fold cross validation technique is used for validation, with k=10. Among these 10 folds, 9 folds are used for training and 1-fold is used for performance evaluation. Here, all the results are displayed after performing various experiments. *Table 2* depicts that the raw data collected from the dataset consists of various anomalies highlighted in green and yellow color in the raw data column. On this raw data, various data cleaning techniques are applied as shown in *Table 2*, and obtained data is in data after the cleaning column. *Table 3* shows that even after applying data cleaning techniques some anomalies are still left and removed after applying data transformation techniques mentioned in *Figure 1*. In *Table 3* data after the cleaning column acts as input data and the

preprocessed data is obtained in data after the transformation column shown in *Table 3* after applying transformation techniques as shown in *Figure 1*.

4.2Data after cleaning

In *Table 2*, raw data is given from datasets contains various anomalies highlighted in green and yellow color. All these anomalies are removed after applying various data cleaning techniques such as uppercase, punctuations, etc. So, cleaned data is obtained in data after cleaning column.

4.3Data after transformation

In *Table 3* data after cleaning acts as input and data after transformation acts as output after applying stopwords removal, POS tagging, lemmatization, etc.

Table 2 Raw data before and after cleaning

Raw data	Data after cleaning
Crustis not good.	crust is not good
Not tasty and the texture was just nasty.	not tasty and the texture was just nasty
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	stopped by during the late may bank holiday off rick steve recommendation and loved it
The selection on the menu was great and so were the prices	the selection on the menu was great and so were the prices
Now I am getting angry and I want my damn pho.	now i am getting angry and i want my damn pho
Honestly it didn't taste THAT fresh.)	honestly it didn't taste that fresh
The potatoes were like rubber and you could tell they have been made up ahead of time being kept under a warmer.	the potatoes were like rubberand you could tell they had been made up ahead of time being kept under a warmer
I was disgusted because I was pretty sure that was human hair.	i was disgusted because i was pretty sure that was human hair
I was shocked because no signs indicate cash only.	i was shocked because no signs indicate cash only

Table 3 Cleaned data after transformation

Data after cleaning	Data after transformation
crust is not good	crust good
not tasty and the texture was just nasty	tasty texture nasty
stopped by during the late may bank holiday off rick steve recommendation and loved it	stop late may bank holiday off rick steve recommend love
the selection on the menu was great and so were the prices	select menu great price
now i am getting angry and i want my damn pho	get angry want damn pho
honestly it didn't taste that fresh	honest taste fresh
the potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer	potato like rubber could tell make ahead time kept warmer
i was disgusted because i was pretty sure that was human hair	disgust pretty sure human hair
i was shocked because no signs indicate cash only	shock signs indicate cash

Stemming and lemmatization are quite similar but there are some differences between these two approaches, stemming is a rule-based approach and it generates the 'stem' of a particular word by applying a set of rules and morphological analysis without considering the context and POS tags of the words.

However, lemmatization is a dictionary based approach that generates the lemma of a particular word by referring to vocabulary and morphological analysis of words. Initially, lemmatization understands the context of a word thereafter, determines the POS tag, and then generates the 'lemma' of a word *Table 4* depicts the consumer reviews after applying stemming and lemmatization. From *Table 4*, it can be clearly inferred that lemmatization generates the dictionary form of words that are morphologically correct. In contrast, stemming sometimes generates words that are not morphologically correct and less meaningful as compared to lemmatization.

Table 4 Reviews after stemming and lemmatization

Reviews after preprocessing using stemming	Reviews after preprocessing using lemmatization
fri great	fry great
get angri want damn pho	get angry want damn pho
cashier cave eve say still end wayyy overpr	cashier cave ever say still end wayyy overprice
shock sign indic cash	shock sign indicate cash
redeem qualiti restaur inexpens	redeem quality restaurant inexpense
heart attack grill downtown vega absolut flat line excus restaur	heart attack grill downtown vega absolute flat line excuse restaurant
frozen puck disgust worst peopl behind regist	frozen puck disgust worst people behind register
side greek salad greek dress tasti pita hummu refresh	side greek salad greek dress taste pita hummus refresh
heart attack grill downtown vega absolut flat line excus restaur	heart attack grill downtown vegas absolute flat line excuse restaurant

However, both of these algorithms have the time complexities i.e., O(N) but the lemmatization is more complex in use as compared to stemming. As in lemmatization, there is a need to do POS tagging before applying lemmatization. POS tagging is used to assign various tags to words such as nouns, verbs, adverbs, adjectives, etc., after applying POS tagging, the lemmatizer generates the "Lemma" of the words by referring to these tags. If POS tagging is not applied in lemmatization then it takes its default tag as a noun. Due to this, the generated "Lemma" is not so accurate. From the above discussion and results, it can be concluded that lemmatization is more accurate as compared to stemming. It states that there are many words like indic, fri, restaur etc. generated by stemming are not accurate and leads to misclassification whereas fry, Indicate, and restaurant are generated by lemmatization and these words are more accurate and dictionary form of words.

4.4Comparison of results using two sets i.e. data cleaning, data transformation with stemming (set1) and data cleaning, data transformation with lemmatisation (set2)

Here, *Tables 5 and 6* shows the confusion matrix for set1, set 2. It shows the actual and predicted value of various ML classifiers.

Comparison of results using 80:20 split ratio

Table 5 Confusion matrix for cell phone dataset (set1)

Confusion matrix for cell phone dataset (set1)				
SVM	**NB**	**DT**	**RF**	**LR**
{{2,1,0}, {0, 0, 0}, {43, 26, 328}}	{{17, 2, 15}, {0, 2, 7}, {28, 27, 306}}	{{15, 8, 39}, {8, 3, 22}, {22, 16, 267}}	{{3,1,2}, {0,0,0}, {42, 26, 326}}	{{9,2,2}, {1,0,0}, {35,25,326}}

Table 6 Confusion matrix for cell phone dataset (set2)

Confusion matrix for cell phone dataset (set2)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0,0,0}, {0, 0, 0}, {45, 27, 328}}	{{3, 1, 10}, {0, 0, 0}, {42, 26, 318}}	{{11,1,27}, {4,3,11}, {30, 23, 290}}	{{3,0,0}, {0,0,0}, {42,27,328}}	{{9,2,1}, {0,0,0}, {36,25,327}}

In literature various researchers performed the comparison of stemming and lemmatization for other languages and in [40, 43] comparison of stemming and lemmatization is performed but not with preprocessing techniques and n-gram features. But these techniques impact the classification process as shown in *Table 7*.

All the results in *Table 7* are calculated from confusion matrices given above. *Table 7* depicts that in case of DT, RF accuracy, precision, recall and f-measure outperform for set2 as compared to set1. For LR precision, recall and f-measure outperform for set2. *Table 8* and *9* shows the confusion matrix for restaurant dataset for set1 and set2. From *Table 10* it is clear that in case of SVM accuracy, precision, f-measure outperform for set2, for NB, LR all the parameters for set2 outperform set1. For RF except accuracy all the parameters outperform for set2. However, for DT all the parameters outperform for set1. *Table 11 and 12* shows the confusion matrix for garments dataset for set1 and set2.

Table 7 Comparison of set1 and set2 on cell phone dataset

Cell phone dataset with split ratio 80:20												
Data cleaning+ data transformation+ stemming (set1)							Data cleaning+ data transformation+ lemmatization (set1)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.820	.825	.825	.825	.39	0:0:58	.820	.820	.820	.820	.38	0:0:59
NB	.812	.812	.812	.812	.42	0:1:15	.810	.810	.810	.810	.41	0:1:10
DT	.730	.712	.712	.712	.46	0:1:17	.762	.755	.755	.755	.44	0:1:00
RF	.830	.822	.822	.822	.40	0:1:56	.835	.839	.840	.839	.38	0:1:23
LR	.840	.837	.837	.837	.38	0:0:41	.840	.840	.840	.840	.37	0:0:50

Table 8 Confusion matrix for restaurant dataset (set1)

Confusion matrix for restaurant dataset (set1)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0,0}, {66, 87}}	{{40,12}, {26, 75}}	{{49,24}, {17, 63}}	{{52,27}, {14, 60}}	{{39,14}, {27, 73}}

Table 9 Confusion matrix for restaurant dataset (set2)

Confusion matrix for restaurant dataset (set2)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0,0}, {61, 92}}	{{34, 5}, {27, 87}}	{{46, 33}, {15, 59}}	{{46, 26}, {15, 66}}	{{32, 7}, {29, 85}}

Table 10 Comparison of set1 and set2 on restaurant dataset

Restaurant dataset with split ratio 80:20												
Data cleaning+ data transformation+ Stemming (set1)							Data cleaning+ data transformation+ Lemmatization(set2)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.570	.568	1.0	.725	.46	0:0:57	.600	.601	1.0	.751	.42	0:0:25
NB	.750	.742	.862	.797	.34	0:1:1	.790	.763	.945	.844	.44	0:1:09
DT	.730	.787	.724	.754	.42	0:1:12	.692	.768	.684	.724	.36	0:0:56
RF	.730	.810	.689	.745	.21	0:1:28	.730	.814	.717	.763	.39	0:1:12
LR	.730	.732	.732	.732	.29	0:0:50	.764	.745	.923	.825	.41	0:0:57

Table 11 Confusion matrix for garments dataset (set1)

Confusion matrix for garments dataset (set1)				
SVM	**NB**	**DT**	**RF**	**LR**
{{21,4}, {83,492}}	{{37, 18}, {67,478}}	{{37, 50}, {67,416}}	{{15, 1}, {89,495}}	{{22, 3}, {82,493}}

Table 12 Confusion matrix for garments dataset (set2)

Confusion matrix for garments dataset (set2)				
SVM	**NB**	**DT**	**RF**	**LR**
{{22, 8}, {88, 482}}	{{38, 15}, {72,475}}	{{41, 67}, {69, 423}}	{{9, 9}, {101,481}}	{{16, 5}, {94,485}}

Table 13 shows that in case of SVM, DT, RF accuracy, precision, recall and f-measure outperform for set2. For NB except recall all the parameters outperform for set1. However, for LR all the parameters outperform for set1 as compared to set2.

Table 13 Comparison of set1 and set2 on garments dataset

Garments dataset with split ratio 80:20												
Data cleaning+ data transformation with Stemming (set1)							Data cleaning+ data transformation with lemmatization (set2)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.840	.840	.840	.839	.18	0:1:00	.850	.855	.991	.918	.21	0:0:42
NB	.858	.877	.963	.918	.20	0:0:17	.855	.868	.969	.916	.21	0:1:00
DT	.776	.859	.863	.861	.27	0:1:12	.816	.869	.899	.884	.265	0:0:46
RF	.823	.826	.981	.897	.20	0:1:56	.853	.847	.997	.916	.20	0:1:23
LR	.860	.857	.993	.920	.21	0:1:00	.835	.837	.989	.907	.18	0:0:30

4.5Comparison of results using two sets i.e. data cleaning, data transformation with lemmatization using unigram features (set3) and data cleaning, data transformation with lemmatisation using bigram features (set4)

Table 14 and 15 shows the confusion matrix for cell phone dataset for set3 and set4.

Table 14 Confusion matrix for cell phone dataset (set3)

Confusion matrix for cell phone dataset (set3)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0,1, 0}, {0, 0, 0}, {45, 26, 328}}	{{16, 3, 17}, {0, 1, 4}, {29, 23, 307}}	{{17, 4, 34}, {5, 5, 14}, {23, 18, 280}}	{{9,1,1}, {0,0,0}, {36, 26, 327}}	{{9,2,1}, {0,0,0}, {36,25,327}}

Table 15 Confusion matrix for cell phone dataset (set4)

Confusion matrix for cell phone dataset (set4)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0, 0, 0}, {0, 0, 0}, {45, 27, 328}}	{{3, 1, 10}, {0, 0, 0}, {42, 26, 318}}	{{11, 1, 27}, {4, 3, 11}, {30, 23, 290}}	{{3,0,0}, {0,0,0}, {42, 27, 328}}	{{0, 0, 0}, {0, 0, 0}, {45,27,328}}

Table 16 shows that in case of LR, NB, RF accuracy, precision, recall and f-measure outperform for set3. For DT except accuracy all the parameters outperform for set4. However, SVM gives equal performance for all the parameters on set3 and set4. *Table 17 and 18* shows the confusion matrix for restaurant dataset for set3 and set4.

Table 16 Comparison of set3 and set4 on cell phone dataset

Cell phone dataset with split ratio 80:20												
Data cleaning+ data transformation with lemmatisation and unigram features (set3)							**Data cleaning+ data transformation with lemmatisation and bigram features (set4)**					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.820	.820	.820	.820	.14	0:0:45	.820	.820	.820	.820	.14	0:0:54
NB	.810	.810	.810	.810	.18	0:0:10	.800	.802	.802	.802	.16	0:1:14
DT	.762	.755	.755	.755	.31	0:0:40	.760	.780	.780	.780	.25	0:1:0
RF	.835	.839	.840	.839	.26	0:1:28	.830	.827	.827	.827	.13	0:1:56
LR	.840	.840	.840	.840	.24	0:0:50	.820	.820	.820	.820	.14	0:0:26

Table 17 Confusion matrix for restaurant dataset (set3)

Confusion matrix for restaurant dataset (set3)				
SVM	**NB**	**DT**	**RF**	**LR**
{{31, 6}, {30,86}}	{{34, 5}, {27,87}}	{{45, 28}, {16, 64}}	{{42, 30}, {19, 62}}	{{32, 7}, {29, 85}}

Table 18 Confusion matrix for restaurant dataset (set4)

Confusion matrix for restaurant dataset (set4)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0, 0}, {61, 92}}	{{11, 4}, {50, 88}}	{{57,78}, {5,14}}	{{57, 75}, {4,17}}	{{5,1}, {56, 91}}

Table 19 shows that that in case of DT, RF accuracy, precision, recall and f-measure outperform for set3. For NB except recall all the parameters outperform for set3. In case of LR accuracy and precision outperform for set3 as compared to set4. However, for SVM only recall outperforms for set3 as compared to set4. *Table 20 and 21* shows the confusion matrix for garments dataset with set3 and set4.

Table 19 Comparison of set3 and set4 on restaurant dataset

Restaurant dataset with split ratio 80:20												
Data cleaning+ data transformation with lemmatization and unigram features (set3)							**Data cleaning+ data transformation with lemmatization and bigram features (set4)**					
Classifier	Accuracy	Precision	Recall	F-	TE	CT	Accuracy	Precision	Recall	F-measure	TE	CT

				measure								
SVM	.600	.601	1.0	.751	.42	0:1:1	.610	.613	.967	.751	.46	0:0:54
NB	.790	.763	.945	.844	.44	0:1:14	.650	.637	.956	.765	.40	0:1:14
DT	.692	.768	.684	.724	.36	0:1:20	.460	.736	.152	.252	.41	0:1:0
RF	.730	.814	.717	.763	.39	0:2:00	.483	.809	.184	.300	.39	0:1:56
LR	.764	.745	.923	.825	.41	0:0:21	.630	.619	.989	.761	.21	0:0:26

Table 20 Confusion matrix for garments dataset (set3)

Confusion matrix for garments dataset (set3)				
SVM	**NB**	**DT**	**RF**	**LR**
{{22,8}, {88,482}}	{{38,15}, {72,475}}	{{41,67}, {69,423}}	{{9,9}, {101,481}}	{{16,5}, {94,485}}

Table 21 Confusion matrix for garments dataset (set4)

Confusion matrix for garments dataset (set4)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0,0}, {110,490}}	{{0,0}, {110,490}}	{{35,38}, {73,452}}	{{9,4}, {101,486}}	{{0,0}, {110,490}}

Table 22 shows that that in case of RF accuracy, precision, recall and f-measure outperform for set3. For NB, LR except recall all the parameters outperform for set3. For DT except precision all the parameters outperform for set4. However, for SVM only recall outperform for set4 as compared to set3.

Table 22 Comparison of set3 and set4 on garments dataset

Garments dataset with split ratio 80:20												
Data cleaning+ data transformation with lemmatization and unigram features (set3)							**Data cleaning+ data transformation with lemmatization and bigram features (set4)**					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.850	.855	.991	.918	.21	0:0:17	.820	.816	1.0	.899	.17	0:1:10
NB	.855	.868	.969	.916	.21	0:0:48	.816	.816	1.0	.899	.17	0:0:14
DT	.816	.869	.899	.884	.26	0:1:12	.828	.862	.930	.894	.18	0:1:20
RF	.853	.847	.997	.916	.20	0:1:41	.825	.822	.991	.900	.19	0:1:03
LR	.835	.837	.989	.907	.18	0:0:22	.820	.816	1.0	.899	.17	0:0:24

4.6Comparison of results using two sets i.e. data cleaning, data transformation with stemming (set1) and data cleaning, data transformation with lemmatisation (set2)

Comparison of results using 70:30 split ratio

Table 23 and 24 shows the confusion matrix for cell phone dataset for set1 and set2. *Table 25* shows that that in case of DT accuracy, precision, recall and f-measure outperform for set1. For NB, RF, LR all the parameters outperform for set2. However, for SVM precision, recall and f-measure outperform for set1.

Table 23 Confusion matrix for cell phone dataset (set1)

Confusion matrix for cell phone dataset (set1)				
SVM	**NB**	**DT**	**RF**	**LR**
{{1,0,0}, {0,0,0}, {76,47,475}}	{{4,2,0}, {0,0,0}, {73,46,475}}	{{24,8,51}, {11,6,28}, {42,34,396}}	{{4,2,0}, {0,0,0}, {73,46,475}}	{{8,1,2}, {1,1,0}, {68,47,473}}

Table 24 Confusion matrix for cell phone dataset (set2)

Confusion matrix for cell phone dataset (set2)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0,1,0}, {0,0,0}, {45,26,328}}	{{8,0,1}, {0,0,0}, {69,48,474}}	{{23,12,49}, {6,2,35}, {48,34,391}}	{{8,0,1}, {0,0,0}, {69,48,474}}	{{8,1,1}, {0,0,0}, {69,47,474}}

Table 25 Comparison of set1 and set2 on cell phone dataset

Cell phone dataset for split ratio 70:30												
	Data cleaning+ data transformation with stemming (set1)						Data cleaning+ data transformation with lemmatisation (set2)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.790	.793	.793	.793	.39	0:0:11	.790	.791	.791	.791	.38	0:1:10
NB	.785	.798	.798	.798	.42	0:1:19	.803	.803	.803	.803	.41	0:0:14
DT	.716	.710	.710	.710	.46	0:0:06	.695	.693	.693	.693	.44	0:1:23
RF	.800	.798	.798	.798	.40	0:1:00	.803	.803	.803	.803	.38	0:1:06
LR	.800	.801	.801	.801	.38	0:1:29	.803	.803	.803	.803	.37	0:1:24

Table 26 and 27 shows the confusion matrix for restaurant dataset for set1 and set2. *Table 28* shows that that in case of DT precision, recall and f-measure outperform for set1. For NB all the parameters outperform for set2. However, for SVM, RF, LR accuracy, recall and f-measure outperform for set2.
Table 29 and *30* shows the confusion matrix for garments dataset for set1 and set2.

Table 31 shows that in case of NB precision, recall and f-measure outperform for set2. For DT accuracy, recall and f-measure outperform for set2, in case of RF all the parameters outperform for set2. However, for SVM accuracy, precision and F-measure outperform for set2. For LR only precision outperform for set2.

Table 26 Confusion matrix for restaurant dataset (set1)

Confusion matrix for restaurant dataset (set1)				
SVM	**SVM**	**SVM**	**SVM**	**SVM**
{{62, 27}, {35,106}}	{{62, 27}, {35,106}}	{{62, 27}, {35,106}}	{{62, 27}, {35,106}}	{{62, 27}, {35,106}}

Table 27 Confusion matrix for restaurant dataset (set2)

Confusion matrix for restaurant dataset (set2)				
SVM	**SVM**	**SVM**	**SVM**	**SVM**
{{53, 8}, {49,120}}	{{53, 8}, {49,120}}	{{53, 8}, {49,120}}	{{53, 8}, {49,120}}	{{53, 8}, {49,120}}

Table 28 Comparison of set1 and set2 on restaurant dataset

Restaurant dataset for split ratio 70:30												
	Data cleaning+ data transformation+ Stemming (set1)						Data cleaning+ data transformation+ Lemmatization (set2)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.730	.752	.797	.774	.46	0:0:03	.750	.710	.938	.808	.46	0:0:10
NB	.713	.713	.714	.742	.44	0:1:1	.774	.732	.875	.797	.44	0:0:18
DT	.704	.798	.654	.719	.42	0:1:0	.704	.761	.648	.700	.45	0:0:41
RF	.730	.772	.714	.742	.44	0:1:08	.739	.732	.875	.797	.46	0:2:00
LR	.709	.736	.774	.754	.43	0:1:07	.752	.710	.938	.808	.45	0:0:23

Table 29 Confusion matrix for garments dataset (set1)

Confusion matrix for garments (set1)				
SVM	**SVM**	**SVM**	**SVM**	**SVM**
{{0, 0}, {162,738}}	{{0, 0}, {162,738}}	{{0, 0}, {162,738}}	{{0, 0}, {162,738}}	{{0, 0}, {162,738}}

Table 30 Confusion matrix for garments dataset (set2)

Confusion matrix for garments (set2)				
SVM	**SVM**	**SVM**	**SVM**	**SVM**
{{30, 4}, {132,734}}	{{30, 4}, {132,734}}	{{30, 4}, {132,734}}	{{30, 4}, {132,734}}	{{30, 4}, {132,734}}

Table 31 Comparison of set1 and set2 on garments dataset

Garments dataset for split ratio 70:30												
	Data cleaning+ data transformation+ Stemming (set1)						Data cleaning+ data transformation+ Lemmatization (set2)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.810	.813	1.0	.897	.18	0:0:19	.820	.820	1.0	.901	.18	0:0:17
NB	.840	.855	.962	.905	.23	0:1:18	.840	.856	.971	.908	.20	0:0:16
DT	.780	.874	.859	.866	.26	0:0:47	.790	.873	.876	.874	.26	0:0:52
RF	.820	.827	.990	.901	.19	0:1:05	.840	.837	.993	.908	.39	0:2:00
LR	.850	.846	.994	.914	.18	0:0:35	.850	.847	.993	.914	.18	0:0:51

4.7 Comparison of results using two sets i.e. data cleaning, data transformation, lemmatization, unigram features (set3) and data cleaning, data transformation, lemmatisation, bigram features (set4)

Table 32 and *33* shows the confusion matrix for cell phone dataset for set1 and set2. *Table 34* shows that that in case of NB, RF, LR accuracy, precision, recall and f-measure outperform for set3. However, for DT all parameters outperform for set4. For SVM accuracy outperform for set4 only. *Table 35 and 36* shows the confusion matrix for restaurant dataset with set3 and set4.

Table 32 Confusion matrix for cell phone dataset (set3)

Confusion matrix for cell phone dataset (set3)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0, 1, 0}, {0, 0, 0}, {45, 26, 328}}	{{8, 0, 1}, {0, 0, 0}, {69, 48, 474}}	{{23, 12, 49}, {6, 2, 35}, {48, 34, 391}}	{{8,0,1}, {0,0,0}, {69, 48, 474}}	{{8, 1, 1}, {0, 0, 0}, {69,47,474}}

Table 33 Confusion matrix for cell phone dataset (set4)

Confusion matrix for cell phone dataset (set4)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0, 0, 0}, {0, 0, 0}, {77, 48, 475}}	{{1, 0, 0}, {0, 0, 0}, {76, 48, 475}}	{{18, 2, 19}, {3, 1, 12}, {56, 45, 444}}	{{1, 0, 0}, {0, 0, 0}, {76, 48, 475}}	{{0, 0, 0}, {0, 0, 0}, {77, 48, 475}}

Table 34 Comparison of set3 and set4 on cell phone dataset

Cell phone dataset for split ratio 70:30												
	Data cleaning+ data transformation with lemmatization and unigram features (set3)						Data cleaning+ data transformation with lemmatization and bigram features(set4)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.790	.791	.791	.791	.43	0:1:17	.791	.791	.791	.791	.43	0:1:17
NB	.803	.803	.803	.803	.32	0:1:16	.785	.793	.793	.793	.42	0:0:59
DT	.695	.693	.693	.693	.48	0:0:51	.770	.771	.771	.771	.46	0:0:54
RF	.803	.803	.803	.803	.32	0:1:56	.793	.793	.793	.793	.42	0:1:13
LR	.803	.803	.803	.803	.32	0:0:58	.791	.791	.791	.719	.43	0:0:45

Table 35 and *36* shows the confusion matrix for restaurant dataset with set3 and set4. *Table 37* shows that that in case of NB, LR accuracy, precision and f-measure outperforms for set3. For RF, DT accuracy, recall and f-measure outperforms for set3. However, for SVM accuracy, precision and f-measure outperforms for set3only.

Table 35 Confusion matrix for restaurant dataset (set3)

Confusion matrix for restaurant dataset (set3)				
SVM	**NB**	**DT**	**RF**	**LR**
{{53, 8}, {49,120}}	{{61, 16}, {41,112}}	{{76, 45}, {26, 83}}	{{61, 16}, {41, 112}}	{{53, 8}, {49,120}}

Table 36 Confusion matrix for restaurant dataset (set4)

Confusion matrix for restaurant dataset (set4)				
SVM	**NB**	**DT**	**RF**	**LR**
{{0, 0}, {102,128}}	{{10, 4}, {92,124}}	{{98, 112}, {4, 16}}	{{100, 106}, {2, 22}}	{{8, 0}, {94,128}}

Table 37 Comparison of set3 and set4 on restaurant dataset

Restaurant dataset for split ratio 70:30												
	Data cleaning+ data transformation with lemmatization and unigram features (set3)						Data cleaning+ data transformation with lemmatization and bigram features (set4)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.750	.710	.938	.808	.46	0:1:47	.560	.560	1.0	.715	.43	0:1:17
NB	.774	.732	.875	.797	.44	0:0:54	.580	.574	.969	.721	.48	0:0:52
DT	.704	.761	.648	.700	.45	0:0:24	.500	.800	.125	.216	.39	0:0:47
RF	.739	.732	.875	.797	.46	0:1:00	.530	.917	.172	.289	.42	0:1:1
LR	.752	.710	.938	.808	.45	0:0:39	.590	.577	1.0	.731	.43	0:0:21

Table 38 and *39* shows the confusion matrix for restaurant dataset with set3 and set4. *Table 40* shows that that in case of NB, RF, LR accuracy, precision and f-measure outperforms for set3. In case of SVM accuracy, precision outperforms for set4. In case of DT accuracy, recall outperform for set4.

Table 38 Confusion matrix for garments dataset (set3)

Confusion matrix for garments dataset (set3)				
SVM	**NB**	**DT**	**RF**	**LR**
{{30, 4}, {132,734}}	{{48, 21}, {120,711}}	{{52,91}, {110, 647}}	{{19, 1}, {143, 737}}	{{37, 5}, {131,727}}

Table 39 Confusion matrix for garments dataset (set4)

Confusion matrix for garments dataset (set4)				
SVM	**NB**	**DT**	**RF**	**LR**
{{56, 60}, {112,672}}	{{4, 0}, {164,732}}	{{56, 60}, {112, 672}}	{{4, 0}, {164,732}}	{{0, 0}, {168,732}}

Table 40 Comparison of set3 and set4 on garments dataset

Garments dataset for split ratio 70:30												
	Data cleaning+ data transformation with lemmatization and unigram features (set3)						Data cleaning+ data transformation with lemmatization and bigram features (set4)					
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**TE**	**CT**
SVM	.820	.820	1.0	.901	.18	0:0:15	.887	.857	.837	.873	.18	0:1:18
NB	.840	.856	.971	.908	.20	0:0:36	.813	.817	1.0	.899	.48	0:1:13
DT	.790	.873	.876	.874	.26	0:0:27	.823	.857	.918	.817	.39	0:1:00
RF	.840	.837	.993	.908	.39	0:0:11	.818	.817	1.0	.899	.18	0:0:58
LR	.850	.847	.993	.914	.18	0:0:60	.813	.813	1.0	.897	.18	0:1:12

5.Discussions

Table 7, *10*, *13* states that between set1 and set2, set2 i.e. data cleaning and transformation with lemmatization gives best results for most of the classifiers. Therefore, in set2 one more feature is included i.e. unigram and bigram features, and named as set3, set4. Again a comparison is performed to find the best between set3 and set4 and it is revealed that set3 i.e. preprocessing with unigram features gives the best results. So, in our further research, we use set3 and it is a basic preprocessing set that can be used for most of the datasets.

Table 7, *10*, *13* gives the comparison between set1 and set2 for restaurant, cell phone and garments datasets with split ratio of 80:20, 80 is training data and 20 is test data. All these results are given in the form of accuracy, precision, recall, f-measure calculated from confusion matrix for various ML classifiers. Further, training error and computation time are also calculated for above mentioned classifiers. *Table 7* depicts that in case of DT, RF accuracy, precision, recall and f-measure outperform for set2 as compared to set1. For LR precision, recall and f-measure outperform for set2. In *Table 10*, majority of the classifiers i.e., SVM, NB, LR, RF

most of the metrics outperforms for lemmatization as compared to stemming. In *Table 13*, SVM, NB, RF and DT most of the metrics gives better performance for lemmatisation. *Table 16, 19, 22* gives the comparison between set3 and set4 for restaurant, cell phone and garments datasets with split ratio of 80:20, 80 is training data and 20 is test data. *Table 16, 19* shows that accuracy, precision, recall and f-measure outperform in case of set3 for majority of classifiers. However, for SVM only recall outperforms for set3 as compared to set4. In *Table 22* for DT, RF all the parameters outperforms for set3. However, in case of SVM, NB, LR majority of metrics outperforms for set3. *Table 25, 28, 31* gives the comparison between set1 and set2 for restaurant, cell phone and garments datasets with split ratio of 70:30, 70 is training data and 30 is test data. For NB, RF, LR all the parameters outperforms for set2. However, for SVM and DT majority of parameters outperforms for set1.

Table 34, 37, 40 gives the comparison between set3 and set4 for restaurant, cell phone and garments datasets with split ratio of 70:30, 70 is training data and 30 is test data. In Table34 for NB, RF, LR all the parameters outperforms for set3. However, for DT majority of parameters outperforms for set4. *Table 37, 40* also shows that majority of classifiers outperforms for set3.

5.1Limitation

Based on our review of literature and experimental results, there is no universal best set of preprocessing techniques for all datasets. Rather, the optimal approach must be determined through careful experimentation. In this study, we propose a basic set of preprocessing techniques that can be applied to a wide range of datasets. However, we also acknowledge the challenge of negation handling, which is a linguistic phenomenon that can alter the polarity of words or sentences in text and thus disrupt sentiment analysis. To address this challenge, we plan to investigate additional preprocessing techniques in our future work, with the aim of improving the accuracy of sentiment prediction.

A complete list of abbreviations is shown in *Appendix I.*

6.Conclusion and future work

After analyzing the results and comparisons of various preprocessing techniques, it can be concluded that there is no one-size-fits-all solution. However, a basic set of preprocessing techniques has been proposed that can be applied to different datasets. The effectiveness of different classifiers varies depending on the dataset and preprocessing techniques used. Our research has compared various preprocessing techniques and found that data cleaning, lemmatization, and unigram features perform best across most classifiers. Nevertheless, we also identified negation handling as a challenge that needs to be addressed to further improve preprocessing and classifier performance. As such, we plan to focus on this challenge in future research.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author's contribution statement

Kartika Makkar: Conceptualization, data curation, formal analysis, investigation, methodology, software, visualization, validation, writing - original draft, writing - review & editing. **Pardeep Kumar:** Writing - original draft, conceptualization, data curation, methodology, software, validation, supervision. **Monika Poriye:** Writing - original draft, conceptualization, data curation, methodology, software, validation, supervision. **Shalini Aggarwal:** Writing - original draft, conceptualization, data curation, methodology, software, validation, supervision.

References

[1] Rosid MA, Fitrani AS, Astutik IR, Mulloh NI, Gozali HA. Improving text preprocessing for student complaint document classification using sastrawi. In IOP conference series: materials science and engineering 2020 (pp. 1-7). IOP Publishing.

[2] Pavan KCS, Dhinesh BLD. Novel text preprocessing framework for sentiment analysis. In smart intelligent computing and applications: proceedings of the second international conference on SCI 2018, 2019 (pp. 309-17). Springer Singapore.

[3] Hacohen-kerner Y, Miller D, Yigal Y. The influence of preprocessing on text classification using a bag-of-words representation. PloS one. 2020; 15(5):1-20.

[4] Barushka A, Hajek P. The effect of text preprocessing strategies on detecting fake consumer reviews. In proceedings of the 3rd international conference on e-business and internet 2019 (pp. 13-7).

[5] Khyani D, Siddhartha BS, Niveditha NM, Divya BM. An interpretation of lemmatization and stemming in natural language processing. Journal of University of Shanghai for Science and Technology. 2021; 22(10):350-7.

[6] Muaad AY, Davanagere HJ, Guru DS, Benifa JB, Chola C, Alsalman H, et al. Arabic document classification: performance investigation of preprocessing and representation techniques. Mathematical Problems in Engineering. 2022; 2022:1-6.

[7] Ali MA, Kulkarni SB. Preprocessing of text for emotion detection and sentiment analysis of Hindi

movie reviews. International conference on IoT based control networks and intelligent systems 2020 (pp. 848-56).
[8] Pandya SS, Kalani NB. Preprocessing phase of text sequence generation for Gujarati language. In 5th international conference on computing methodologies and communication 2021 (pp. 749-52). IEEE.
[9] Kumar D, Rana P. Stemming of punjabi words by using brute force technique. International Journal of Engineering Science and Technology. 2011; 3:1351-7.
[10] Pind J, Magnússon F, Briem S. The icelandic frequency dictionary. The Institute of Lexicography, University of Iceland, Reykjavik, Iceland. 1991.
[11] Ingason AK, Helgadóttir S, Loftsson H, Rögnvaldsson E. A mixed method lemmatization algorithm using a hierarchy of linguistic identities (HOLI). In advances in natural language processing: 6th international conference, GoTAL 2008 Gothenburg, Sweden, 2008 (pp. 205-16). Springer Berlin Heidelberg.
[12] Helgadóttir S. Testing data-driven learning algorithms for POS tagging of icelandic. Nordisk Sprogteknologi. 2004:257-65.
[13] Setiabudi R, Iswari NM, Rusli A. Enhancing text classification performance by preprocessing misspelled words in Indonesian language. Telecommunication Computing Electronics and Control. 2021; 19(4):1234-41.
[14] Alam S, Yao N. The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. Computational and Mathematical Organization Theory. 2019; 25:319-35.
[15] Churchill R, Singh L. Textprep: a text preprocessing toolkit for topic modeling on social media data. In proceedings of the 10th international conference on data science, technology and applications 2021 (pp. 60-70).
[16] Orlovskyi O, Ostapov S. Analysis of the text preprocessing methods influence on the destructive messages classifier. Advanced Information Systems. 2020; 4(3):104-8.
[17] Babanejad N, Agrawal A, An A, Papagelis M. A comprehensive analysis of preprocessing for word representation learning in affective tasks. In proceedings of the 58th annual meeting of the association for computational linguistics 2020 (pp. 5799-810).
[18] Kunilovskaya M, Plum A. Text preprocessing and its implications in a digital humanities project. In proceedings of the student research workshop associated with RANLP 2021 (pp. 85-93).
[19] Dash NS, Dash NS. Lemmatization of inflected nouns. Language Corpora Annotation and Processing. 2021:165-94.
[20] Prakash C, Chittimalli PK, Naik R. Domain specific text preprocessing for open information extraction. In 15th innovations in software engineering conference 2022 (pp. 1-5).
[21] Ranganathan G. A study to find facts behind preprocessing on deep learning algorithms. Journal of Innovative Image Processing (JIIP). 2021; 3(1):66-74.
[22] Mohammad F. Is preprocessing of text really worth your time for online comment classification? Proceedings on the international conference on artificial intelligence 2018 (pp.1-7).
[23] El KA, Zeroual I. The effects of pre-processing techniques on Arabic text classification. International Journal of Advanced Trends in Computer Science and Engineering. 2021; 10(1):41-8.
[24] Yogish D, Manjunath TN, Hegadi RS. Review on natural language processing trends and techniques using NLTK. In recent trends in image processing and pattern recognition: second international conference, RTIP2R 2018, Solapur, India, Revised Selected Papers, Part III 2019 (pp. 589-606). Springer Singapore.
[25] A ML, Benoit K, Keyes O, Selivanov D, Arnold J. Fast, consistent tokenization of natural language text. Journal of Open Source Software. 2018; 3(23):1-3.
[26] Orellana G, Arias B, Orellana M, Saquicela V, Baculima F, Piedra N. A study on the impact of pre-processing techniques in Spanish and English text classification over short and large text documents. In international conference on information systems and computer science 2018 (pp. 277-83). IEEE.
[27] Uysal AK, Gunal S. The impact of preprocessing on text classification. Information Processing & Management. 2014; 50(1):104-12.
[28] Méndez JR, Iglesias EL, Fdez-riverola F, Díaz F, Corchado JM. Tokenising, stemming and stopword removal on anti-spam filtering domain. In current topics in artificial intelligence: 11th conference of the Spanish association for artificial intelligence, CAEPIA 2005, Santiago de Compostela, Spain, 2006 (pp. 449-58). Springer Berlin Heidelberg.
[29] Kotsiantis SB, Kanellopoulos D, Pintelas PE. Data preprocessing for supervised leaning. International Journal of Computer Science. 2006; 1(2):111-7.
[30] Hickman L, Thapa S, Tay L, Cao M, Srinivasan P. Text preprocessing for text mining in organizational research: review and recommendations. Organizational Research Methods. 2022; 25(1):114-46.
[31] Saif H, Fernandez M, He Y, Alani H. On stopwords, filtering and data sparsity for sentiment analysis of twitter. Ninth international conference on language resources and evaluation. 2014 (pp.810-17).
[32] Srividhya V, Anitha R. Evaluating preprocessing techniques in text categorization. International Journal of Computer Science and Application. 2010; 47(11):49-51.
[33] Angiani G, Ferrari L, Fontanini T, Fornacciari P, Iotti E, Magliani F, et al. A comparison between preprocessing techniques for sentiment analysis in twitter. KDWeb. 2016:1-11.
[34] Haddi E, Liu X, Shi Y. The role of text pre-processing in sentiment analysis. Procedia Computer Science. 2013; 17:26-32.
[35] Dos SFL, Ladeira M. The role of text pre-processing in opinion mining on a social media language dataset.

In Brazilian conference on intelligent systems 2014 (pp. 50-4). IEEE.
[36] Hemalatha I, Varma GS, Govardhan A. Preprocessing the informal text for efficient sentiment analysis. International Journal of Emerging Trends & Technology in Computer Science. 2012; 1(2):58-61.
[37] Jianqiang Z, Xiaolin G. Comparison research on text pre-processing methods on twitter sentiment analysis. IEEE Access. 2017; 5:2870-9.
[38] M. AS, Mustapha M. The effect of noise elimination and stemming in sentiment analysis for Malay documents. In proceedings of the international conference on computing, mathematics and statistics (iCMS 2015) Bridging Research Endeavors 2017 (pp. 93-102). Springer Singapore.
[39] Boban I, Doko A, Gotovac S. Sentence retrieval using stemming and lemmatization with different length of the queries. Advances in Science, Technology and Engineering Systems. 2020; 5(3):349-54.
[40] Kariyawasam KT, Senanayake SY, Haddela PS. A rule based stemmer for Sinhala language. In 14th conference on industrial and information systems 2019 (pp. 326-31). IEEE.
[41] Akhmetov I, Pak A, Ualiyeva I, Gelbukh A. Highly language-independent word lemmatization using a machine-learning classifier. Computing and Systems. 2020; 24(3):1353-64.
[42] Balakrishnan V, Lloyd-yemoh E. Stemming and lemmatization: a comparison of retrieval performances. In proceedings of SCEI Seoul conferences. 2014 (pp.10-4).
[43] Ozturkmenoglu O, Alpkocak A. Comparison of different lemmatization approaches for information retrieval on Turkish text collection. In international symposium on innovations in intelligent systems and applications 2012 (pp. 1-5). IEEE.
[44] Dalianis H, Jongejan B. Hand-crafted versus machine-learned inflectional rules: the euroling-siteseeker stemmer and CST's lemmatiser. In LREC 2006 (pp. 663-6).
[45] Korenius T, Laurikkala J, Järvelin K, Juhola M. Stemming and lemmatization in the clustering of finish text documents. In proceedings of the thirteenth ACM international conference on information and knowledge management 2004 (pp. 625-33).
[46] Gupta D, Kumar YR, Sajan N. Improving unsupervised stemming by using partial lemmatization coupled with data-based heuristics for Hindi. International Journal of Computer Applications. 2012; 38(8):1-8.
[47] Kurniasih A, Manik LP. On the role of text preprocessing in BERT embedding-based DNNs for classifying informal texts. Neuron. 2022; 1024(512):927-34.
[48] Haque TU, Saber NN, Shah FM. Sentiment analysis on large scale Amazon product reviews. In international conference on innovative research and development 2018 (pp. 1-6). IEEE.
[49] Krishna A, Akhilesh V, Aich A, Hegde C. Sentiment analysis of restaurant reviews using machine learning techniques. In emerging research in electronics, computer science and technology: proceedings of international conference 2019 (pp. 687-96). Springer Singapore.
[50] Makkar K, Kumar P, Poriye M, Aggarwal S. A comparative study of supervised and unsupervised machine learning algorithms on consumer reviews. In world conference on applied intelligence and computing 2022 (pp. 598-603). IEEE.

Kartika Makkar was born in Chamba district of Himachal Pradesh. She received her bachelor degree in Computer Science& Applications from government PG College Chamba under Himachal Pradesh University, Shimla in 2013. She received her master degree in Computer Science & Applications from department of Computer Science & Applications in 2017 and now she is pursuing PhD in Computer Science & Applications from department of Computer Science & Applications, Kurukshetra University, Kurukshetra (Haryana) and her current research work is in Machine Learning and NLP.
Email: sonikartika19@gmail.com

Dr. Pardeep Kumar received his PhD in Computer Science & Applications, M.Sc. (Computer Science) and M.Sc. (Statistics) degree from Kurukshetra University, Kurukshetra. Presently he is working as Associate Professor in Kurukshetra University, Kurukshetra. His research interest lies in Optimization, Cloud Computing, Network Routing and Soft Computing. He has published more than 75 research papers in referred journals and international conferences.
Email: pmittal@kuk.ac.in

Dr. Monika Poriye, is currently working as an Assistant Professor in Department of Computer Science and Applications, Kurukshetra University. She has completed her doctorate in Computer Science & Applications, M.Tech. and M.Sc. from Kurukshetra University, Kurukshetra. Her area of interest is Information Security, Web Development, Cloud Computing, Machine Learning etc. Dr. Monika has published more than 40 research papers in various journals/conferences.
Email: monikaporiye@gmail.com

Dr. Shalini Aggarwal is currently working as an Assistant Professor in the Department of Computer Science, S.U.S. Govt. College, Matak Majri, Indri (Karnal). She has published a number of research papers in the various national and international journals. She has presented her research in various conferences. She has done her Ph.D in Computer Science and Applications from Department of Computer Science and Applications Kurukshetra University, Kurukshetra in the field of Computer Networks. Her research areas include Computer Networking, Machine Learning, Soft Computing, etc. She is having more than 16 years of teaching experience and more than 10 years of research experience.
Email: aggshamit@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	AI	Artificial Intelligence
2	API	Application Programming Interface
3	BILSTM	Bidirectional LSTM
4	BERT	Bidirectional Encoder Representation from Transformers
5	C	Cost
6	CNN	Convolutional Neural Network
7	CT	Computation Time
8	DNN	Deep Neural Network
9	DT	Decision Tree
10	FN	False Negative
11	FP	False Positive
12	GRU	Gated Recurrent Unit
13	HOLI	Hierarchy of Linguistic Identities
14	HTML	Hypertext Markup Language
15	JJ	Adjective
16	LR	Logistic Regression
17	LSTM	Long and Short Term Memory
18	ME	Maximum Entropy
19	ML	Machine Learning
20	NB	Naïve Bayes
21	NLP	Natural Language Processing
22	NLTK	Natural Language Toolkit
23	NNP	Proper Noun Singular
24	POS	Parts of Speech
25	RF	Random Forest
26	SVM	Support Vector Machine
27	TN	True Negative
28	TP	True Positive
29	TE	Training Error
30	TFIDF	Term Frequency and Inverse Document Frequency
31	TNT	Trigrams‘n’tags
32	URL	Uniform Resource Locator
33	VBN	Verb Past Participle
34	VBZ	Verb Third Person Singular Present