Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2021, Article ID 2529984, 8 pages
https://doi.org/10.1155/2021/2529984

Research Article

A Generalized Method for Sentiment Analysis across Different Sources

Abubakar M. Ashir

Department of Computer Engineering, Tishk International University, Erbil, Iraq

Correspondence should be addressed to Abubakar M. Ashir; abubakar.ashir@tiu.edu.iq

Received 2 October 2021; Revised 27 November 2021; Accepted 5 December 2021; Published 18 December 2021

Academic Editor: Francesco Rundo

Sentiment analysis is widely used in a variety of applications such as online opinion gathering for policy directives in government, monitoring of customers, and staff satisfactions in corporate bodies, in politics and security structures for public tension monitoring, and so on. In recent times, the field met with new set of challenges where new algorithms have to contend with highly unstructured sources for sentiment expressions emanating from online social media fora. In this study, a rule and lexical-based procedure is proposed together with unsupervised machine learning to implement sentiment analysis with an improved generalization ability across different sources. To deal with sources devoid of syntactic and grammatical structure, the approach incorporates a ruled-based technique for emoticon detection, word contraction expansion, noise removal, and lexicon-based text preprocessing using lexical features such as part of speech (POS), stop words, and lemmatization for local context analysis. A text is broken into number of tokens with each representing a sentence and then lexicon-dependent features are extracted from each token. The features are merged together using a combining function for a given text before being used to train a machine learning classifier. The proposed combining functions leverage on averaging and information gain concepts. Experimental results with different machine leaning classifiers indicate that improved performance with great deal of generalization capacity across both structured and nonstructured sources can be realized. The finding shows that carefully designed lexical features reinforce learning process in unsupervised learning more than using word embeddings alone as the features. Obtained experimental results from movie review dataset (recall = 74.9%, precision = 70.9%, F1-score = 72.9%, and accuracy = 72.0%) and twitter samples' datasets (recall = 93.4%, precision = 89.5%, F1-score = 91.4%, and accuracy = 91.1%) show the efficacy of the proposed approach in comparison with other state-of-the-art research studies.

1. Introduction

Sentiment analysis is a part of natural language processing (NLP) which receives tremendous attention in recent history. This may not be unconnected to the availability of social media platforms, big data storage, increased Internet connectivity, accessibility, and unending desire by big business and governments to understand people's opinions for policy conceptualizations and monitoring. At the back of this boom is the recent breakthrough in machine and deep learning algorithms leading to an astronomical improvement in performance of NLP tasks. Sentiment analysis crisscrosses subfields of computational linguistic and information retrieval. In general context, the major task in sentiment analysis has to do with tagging a given text according to expressed opinion which usually involves three tasks: (i) determine objectivity of a text (i.e., subjective or objective), (ii) determine the polarity of a subjective text (i.e., positive or negative), and (iii) determine the strength of the subjective text [1]. There are two major approaches that exist in the literature for sentiment analysis: lexicon-based and machine learning-based approach. Each of these approaches has their benefits and drawbacks. Lexicon-based approach is a rule-based method which employs computing sentiments by considering the semantic orientation of the words or phrases in the text [1]. This implies the use of a dictionary of words which are tagged with lexical features such as sentiment polarity orientation, part of speech (POS), and glosses. In fact, the approach represents a piece of word as a token or a bag of words where semantic orientation of each word is computed within the local context which are then used alongside rule-based combining function to compute the overall sentiment [2–5].

On the other sides, the machine learning approach to classification of sentiment in text depends on the use of labelled data to train classifiers such as Naïve Bayes (NB), support vector machine (SVM), and maximum entropy (ME) using supervised learning approach [6]. Deep learning gated recurrent networks, such as Long-Short-Memory (LSTM) network, are also found to be even more effective in some sentiment and NLP-related tasks as reported in [6, 7]. The machine and deep learning methods require the text to be preprocessed and then converted into the feature vector using numerous schemes such as word embedding. Numerous techniques for performing word embeddings and feature extractions have been used in literature for efficient representation of the semantic context and orientation of a text in NLP-related task [6]. While it is very difficult to build a lexicon-based dictionary and much of the approaches depend on the few existing ones, machine learning approaches are equally challenged by the need for enormous and tedious labor in producing labelled data and lack of clarity on how the features are learned, and the generalization ability of the learners is also of a concern [3, 8]. The contributions of this research are outlined below:

1. Propose and implement a rule-based dataset for emoticon and word contractions' expansion for translation of commonly used expressions such as emojis, slangs, and abbreviations by social media platforms' users
2. Propose a lexical-based procedure for sentiment scores' computation with local context
3. Propose two combining functions for merging scores from the various tokens in a text into a single feature vector
4. Use the proposed feature extraction method with unsupervised machine learning classification

2. Review of Relevant Literature

Some of the most well-known gold-standard English language lexical dictionaries for sentiment analysis includes the Linguistic Inquiry and Word Count (LIWC), General Inquirer (GI), WordNet, and Affective Norms for English Words (ANEW) [9–12]. LIWC and GI are made of straightforward dictionary of list of words that are categorized into binary classes (positive and negative polarities) purely based on the context-free semantic orientation of the words. These two mostly suffer from the lack of coverage for sentiment express in social media and do not capture variation in intensities of the sentiment expressions between the words of same classes. Unlike LIWC and GI, ANEW encodes the intensities of the expression by providing a set of normative ratings which depends on the strength of the emotion in the words such as pleasure, arousal, and dominance [11]. On the contrary, WordNet lexical database provides a clustering scheme where words are placed into groups of synonyms known as *synsets* [12]. Based on these human-validated lexical databases, quite a significant number of lexicon and rule-based sentiments analysis such as SentiStrength, sentiWordNet, and Valence Aware Dictionary for Sentiment Reasoning (VADER) were developed [3, 5, 13]. For instance, in VADER, the authors used combination of qualitative and quantitative methods. They constructed list of lexical features, and each feature is associated with sentiment intensity measures. These features are specifically designed to handle sentiment in microblog-like contexts. To emphasize on the intensity of the sentiment in those texts, general rules that encapsulate grammatical and syntactical conventions for expressing and emphasizing sentiment intensity were also considered. The authors reported higher performance and better generalization across the contexts compared to other state-of-the-art methods [13].

Due to overwhelming popularity of the social media among the populace with conversations, we devoid syntactical and grammatical structure; the conventional rule-based methods suffer a decline in performance. Recently, machine learning-based sentiment analysis is prevalent [2, 6, 14–17]. In the submission titled "Deep Learning for Automated Sentiment Analysis of Social Media," Li-Chen Cheng and Song-Lin Tsai [8] proposed a method for sentiment analysis framework based on deep learning models in which sentiment data were extracted from social media. They used gated recurrent neural network and bidirectional LSTM to train the prepared semantic. Results from their approach were evaluated using Accuracy, Precision, Recall, Specificity metrics, and overall average score of 75% and were attained. Antoine Boutet et al. [18] presented a sentiment analysis on data extracted the main stream of Twitter related to the 2010 UK to predict election results for three major political parties. They proposed a simple and practical algorithm to identify the political leaning of users using the amount of Twitter messages which seem related to political parties. SVM and NB classifiers were used to classify the sentiments both based on the volumetric and retweets' partition. They claimed that the Bayesian classifier on retweets and volumetric sematic performed best with prediction accuracy and showed that the best-performing classification method—which uses the number of Twitter messages referring to a particular political party—achieved about 86% classification accuracy without any training phase.

Saifuddin Ahmad et al. [19], in an article "Tweets and votes: a four-country comparison of volumetric and sentiment analysis approach," proposed a method for election prediction using both volumetric, supervised, and unsupervised sentiment analysis of data extracted from the social media. The authors considered volumetric, sentiments and regional Internet availability, and social media engagement of various region using 12 metrics. Their finding suggested that social media-based expresses sentiments can provide a reasonable prediction. To implement their supervised analysis methods after cleaning up the extracted data, a sentiment lexicon called SentiStrength [5] was used in the python natural language processing libraries and naïve Bayes classifier.

In our method, a rule-based approach is designed to handle both structured texts which follow with general grammatical and syntactical rules and unstructured text from microblogs and social media platforms which do not usually follow the general grammatical and syntactical rules. We deployed rule and lexicon-based approach to preprocess the text and used the approach in [13] together with proposed combining functions to extract features which serve as the feature vector for the supervised machine learning classification. The rest of the study is organized into 5 sections. Sections 1 and 2 present the introduction and review of the related literature, respectively. Section 3 contained details about the proposed method. In Section 4, results from the experiments and discussion were presented. Conclusion about the research was presented in Section 5.

3. Implementation

Three critical stages were designed to realize the implementation of the proposed method. Stage one deals with the text preprocessing which addresses most of the prevalent issues with text from a nonstructured source such as Twitter. The second stage is concerned with the extraction of sentiment-aware features from the preprocessed text which is subsequently used to train the machine learning algorithm. The last stage is about training methods for a number of classifier models.

3.1. Text Preprocessessing. Bulk of the nowadays data used for sentiment analysis comes from microblogs and online social media fora such as Twitter and Facebook. These texts are mostly noncompliant with the grammatical and syntactical rules. They contain components such as emojis, abbreviations, slang expressions, and impurities such as URL, hashtags, and many others. Such texts distort the performance of rule-based sentiment analyzers and will hitherto skew or render meaningless outcome of any sentiment analyzer if not properly handled. Four rule-based stages were designed as shown in Figure 1 to preprocess the text.

3.1.1. Emoticon and Contraction Expansion. In this context, contractions refer to the words, phrases, or sentences that are usually shortened by dropping some of their letters or even completely represented in different forms for the ease of writing, e.g., “I’ll *c* u ltr” for “I will see you later.” A lookup table was designed (Table 1) in spreadsheet containing 400 commonly abbreviated/shortened words with their expanded/full forms, which is referred to contraction expansion. Similarly, emoticons or emojis are embedded in most online forum applications which are used to quickly convey emotions without typing a word. In some works [15], a text containing emoticon is automatically classified in accordance to its emoticon ignoring the textural message it contains. Here, emoticons in a text are converted into its most commonly used meaning interpreted from a lookup table with over 500 emoticons, as shown in Table 1.

3.1.2. Noise Removal. In most of the online fora, tagging of people, topics, named-entity, and URL are a very common practice. Such tagging which can be easily recognized by their initials, e.g., (@, #, http://, https://etc.) have virtually no contribution to the conveyed sentiment in text. On the contrary, they increase computational cost and could degrade the performance or generalization ability of an analyzer. Carefully crafted rule-based approach was included to search, find, and remove such occurrences in the text.

3.1.3. Lexical Transform. This is a context-aware stage that solely depends on the semantic orientation of the text in a local context. Three major lexical features are used which include sentence/word tokenization, word lemmatization, and POS tagging. Tokenization allows the text to be broken into number of sentences, whereas word tokenization breaks down a sentence into sort of word list or bag-of-word representation. POS tagging uses linguistic corpora to associate or tag a word with a part of speech based on the word meaning and its local context use. POS is used to remove stop word (e.g., punctuation, conjunction, preposition, and interjection) and named-entity word (e.g., APPLE, AMAZON, and Silicon valley) which are independent of the sentiment in the text. Word lemmatization is used to convert word that can appear in different forms into its unambiguous root form, e.g., the words “run, ran, running” have lemma root as “run.” These practices greatly improve the learning process by avoiding ambiguity and feature redundancy. These analyses are done using NLTK and Spacy libraries in python which come with numerous linguistic corpora.

3.2. Feature Extraction. Feature extraction concerns with the representation of the preprocessed text into a vector of integers or floating-point numbers that are compatible with the machine learning classifiers. Two approaches were used, one of which is to use sentiment intensity analyzer (SIA) proposed in [13], which is also embedded in NLTK library. SIA is a rule-based sentiment analyzer which predicts not only the polarity of the text but also the strength of each opinion classes in the text. VADER has four numerical scores as output: Positive, Negative, Neural, and Compound scores. The compound score of SIA is the normalized scores for positive and negative sentiment over the interval [−1 +1], where negative expression interpreted as having compound scores less than zero and positive sentiments having compound scores greater than zero. Most positive score with most negative expression taking +1 and −1 compound scores, respectively. These numerical outputs serve as the basis for the feature vector. The other feature vector considered is using text vectorization technique, which also converts the text into the numerical vector. Though many forms of word embedding schemes are available, in this work, a word2vector [4] word embedding techniques are deployed for vectorization. In word2vector, texts are converted to a fixed-length vector.

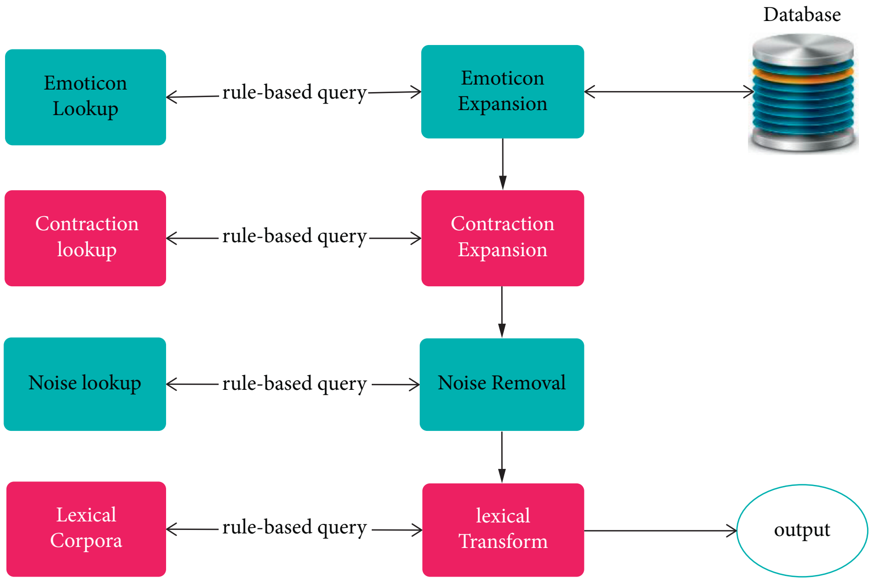


Figure 1: Flowchart for a text preprocessing using rule and lexical-based approach.

Table 1: Sample of lookup table database used for translating emoticons, abbreviations, and slangs in preprocessing.

Emoticons	Meaning	Abbreviation	Meaning
:-),:-],:-3,:->,8-),:-},:o),:c),:̀), =], =),),:],:3,:>,8),:}	Happy	LOL, lol, lol	Laughing
:-D, 8-D, x-D, X-D,:D, 8D, xD, XD, = D, = 3, B̀D, c:, C:	Laughing	Math	Mathematics
:-))	Very happy	coz	Because
:-(,:-c,:-<,:-[,:(,:c,:<,:[,:-||,>:[,:{,:@,:(,; (	Angry	PE	Physical education
:'-(,:'(	Crying	Cc	Carbon copy
:'-),:')	Happiness	No.	Number
D-':,D:< *D*:,D8,D; ,*D* = ,DX	Sadness	Eg	For example,
:-O, : O,:-o,:o,:-0 8–0, >:O	Surprise	CV	Curriculum vitae
:-*, :*, :×	Kiss	ATM	At the moment

3.2.1. Combining Function. The SIA was designed to handle microblog sentiment analysis which is usually expressed in a short form. To extend its use to non-microblog sources which are lengthy with multiple sentences (e.g., review of movies and products), the analysis is done based on a tokenized sentence, as shown in Figure 2. For a multisentence text, scores from all sentences need to be combined into a single feature vector. Usually, scores from individual sentence can be combined by taking their averages across attributes (i.e., positive, negative, neutral, or compound), but this may not give the best performance especially in a situation where different sentences contains varying sentiment polarities and strength. In such cases, sometimes, it is difficult to interpret the expressed sentiment based on the scores.

We proposed a combining function that uses both averaging and information gain functions. For n, number of sentences in a text, the averaging function computes the average of attribute (i.e., positive, negative, neutral, or compound) X scores in n sentences. Therefore, average score y_k for an attribute X_k with index k is given in accordance to

$$y_k = \frac{1}{n}\sum_{i=1}^{n} X_{ik}. \tag{1}$$

The averaging results are adopted when the average compound scores are greater than 0.5 or less than −0.5. This means that there is over 50% confidence in the polarity of the text. For *compound* scores outside of that range, Information Gain (IG) is used as the combining function to estimate the measure of clarity or information what will be gained from separating the scores into the three classes (positive "Pos," negative "Neg," and neural "Neu"). To compute the IG, we consider only three attributes and we assign the class for each sentence based on the compound scores. If the absolute value of the compound score is less than neutral score, "Neu" class is assigned; otherwise, we assign "Pos" for compound scores greater than zero and "Neg" class for *compound* scores less than zero (e.g., see Table 2). Therefore, IG, for an attribute k, IG_{a_k}, can be computed using the relation in equation (2). The final feature, when IG is used, will be the three IG values for *positive*, negative, and neutral attributes which are then concatenated with the compound score from SIA. Hence, the feature vector size is always constant whether averaging function or info gain is used:

$$IG_{a_k} = H(X) - H(X \backslash a_k), \tag{2}$$

where $H(X)$ is the combined entropy of all the classes and $H(X \backslash a_k)$ is the conditional entropy of a class X giving an attribute a_k.

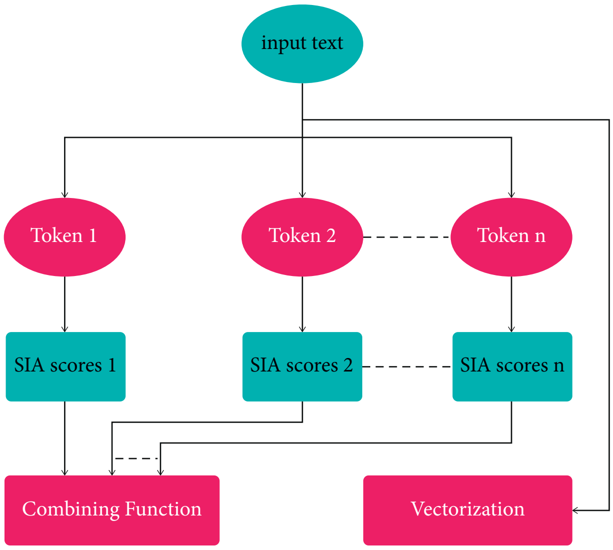


Figure 2: Proposed feature extraction: (a) an input text is broken into sentence tokens, with scores computed from each token and then combined; (b) features vectors are directly computed from the entire text using word embedding.

Table 2: SIA scores for tokens and computation of the average score and class assignment for IG computation.

Tokens	Positive	Negative	Neutral	Compound	Class
1	0.19	0.287	0.522	−0.3182	*Neu*
2	0.0	0.45	0.55	−0.5994	*Neg*
Average	**0.095**	**0.3685**	**0.536**	**−0.4588**	

The bold values stress the final feature that will be used.

The text below is made of two tokens; for each token, SIA scores are computed separately, and since the average compound score (−0.4588) for the entire text is in the range [+5−−5], IG will be used to extract final feature instead of averaging function. The class assignment will be in accordance with the preceding explanation.

“The food tastes great but the place is dirty. I will have no problem going back again.”

3.2.2. Vectorization. Word vectorization in NLP is mostly achieved using world embedding techniques. These techniques represent text word into a real-valued vectors where words with similar semantics have close representation (coordinates). The real-valued vectors have predefined fixed sized length of each and are usually learned from vocabulary from a corpus of text using the neural network model and in some cases document statistics. Some of the popular word embeddings include Embedding Layer, Word2Vec, and Glove embeddings. Though these embeddings capture the semantic of the text, they often miss the sentiment polarity of the words. New embedding techniques to capture sentiment orientation in text are also proposed [4, 20].

3.3. Classifications. The classification process is based on the supervised machine learning which requires labelled data to learn the patterns. Binary classifiers are used to classify the text into two classes: Positive and Negative. We applied normalized exponential function (SoftMax) to normalize the output of a classifier to a probability distribution over predicted output class. The output is a real-valued numbers which sums up to one and indicates the confidence of the prediction for each class. After the training, the model can be used for classifying new data, as shown in Figure 3.

4. Results and Discussion

To validate the efficacy of the proposed approach, four performance matrices were used with two datasets which have different orientations in terms of structure and mode of expression. The matrices include Precision, Recall, F1-measure, and Accuracy. The precision matric (also called the Positive Predictive Value) is the fraction True Positive results out of the total positive results predicted by the classifier, and it provides a probabilistic measure of how a positive opinion is predicted. The recall metric (also known as sensitivity) is the fraction of the True Positive results out of the total positive results, therein the gold-standard ground-truth benchmark (i.e., human ratings). The last parameter, F-measure, is the harmonic mean between the *recall* and *precision* as expressed in equations 3–6:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$F1 - \text{score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \tag{5}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{6}$$

4.1. Dataset. Two different labelled datasets are used to evaluate the proposed method. The first dataset tagged “movie reviews” and included in the NLTK corpora consists of 1000 positive and 1000 negative processed reviews [21, 22]. The 75% of movie reviews’ dataset was used for training and the remaining 25% was used for testing and the experimental results presented in Table 2. Meanwhile, the second dataset named “twitter samples” retrieved from the Twitter Streaming API consists of 10 thousand labelled tweets categorized into negative and positive sentiments [22, 23]. This dataset was only for used to test the proposed trained model.

4.2. Results. Six different machine learning classification algorithms were used alongside the two proposed feature extraction approaches presented in the paper. These classifiers include maximum entropy SVM, NB, MLP, Adaboost, and Logistic Regression. In Table 3, we present comparison results between the SIA method and three of the best-performing classifiers (SVM, MLP, and LR) with two different feature extraction methods. Meanwhile, in Table 4,

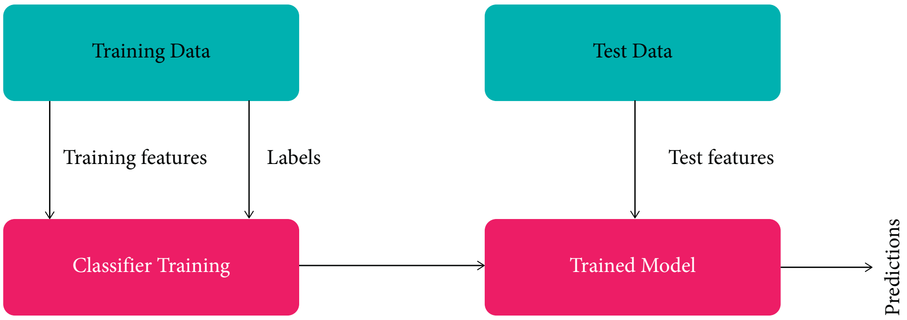


Figure 3: Supervised learning model: extracted features with labels are used for training and unlabeled data are used for testing.

Table 3: Results' comparison from movie review dataset with different feature extractions.

Functions	TP	FP	TN	FN	Recall	Precision	F1	Accuracy
SIA	212	140	109	39	**84.5%**	60.2%	70.3%	64.2%
Support vector machine (SVM)								
Average_IG	182	73	176	69	72.5%	71.4%	71.9%	71.6%
Embeddings	171	85	155	89	66.8%	65.8%	66.3%	65.2%
Multilayered perceptron (MLP)								
Average_IG	188	77	172	63	74.9%	**70.9%**	**72.9%**	**72.0%**
Embeddings	174	94	174	58	75.0%	64.9%	69.6%	69.6%
Logistic regression (LR)								
Average_IG	183	75	174	68	72.9%	70.9%	71.9%	71.4%
Embeddings	177	78	162	83	69.4%	68.1%	68.7%	67.8%

Table 4: Summary of the classifiers' performance between the SIA and proposed method using four metrics and different classifiers.

	SIA	SVM	Naive Bayes	MaxEnt	MLP	Adaboost	Logistics
Precision	84.5	72.5	92.0	96.41	74.9	69.4	75.4
Recall	60.2	71.4	59.7	50.1	70.9	66.0	63.6
F1-score	70.3	71.9	72.4	65.9	72.9	67.7	69.0
Accuracy	64.2	71.6	64.8	50.0	72.0	69.2	68.6

Table 5: Results of Twitter sample dataset with average-IG feature and different classifiers.

Functions	TP	FP	TN	FN	Recall	Precision	F1	Accuracy
SIA	4658	1030	3970	342	93.2%	81.9%	87.2%	86.3%
ME	617	3805	1195	4383	12.3%	14.0%	13.1%	18.1%
NB	441	427	4573	4559	8.8%	50.8%	15.0%	50.1%
LR	4278	879	4121	722	85.6%	83.0%	84.2%	84.0%
MLP	4762	557	4343	338	**93.4%**	**89.5%**	**91.4%**	**91.1%**
Adaboost	3844	855	4145	1156	76.9%	81.8%	79.3%	79.9%
SVM	4674	571	4329	426	91.7%	89.1%	90.4%	90.0%

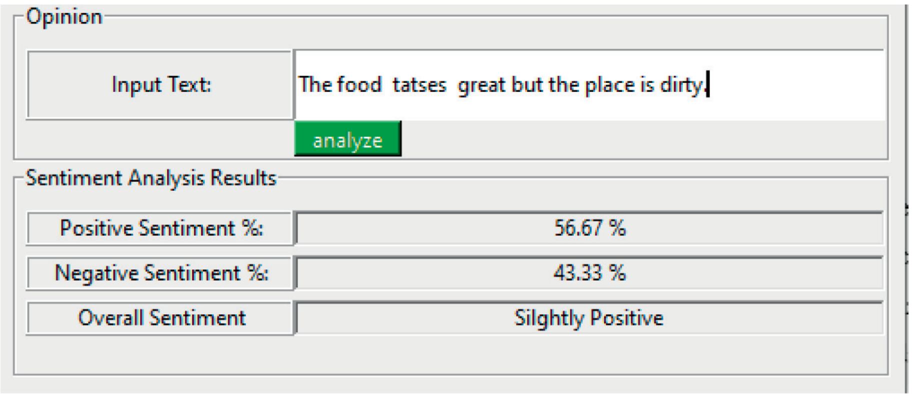
The bold values indicate the best performing entry along the column which is helpful for the reader.

performances between the classifiers were compared based on the four matrices.

5. Discussion

Based on the evidences in the training, the combination of future extraction using averaging and IG with the MLP classifier edges out most of the other possibilities with the exception of Naives Bayes and MaxEnt which outperformed MLP in terms of *precision*. This is further supported in terms of the generalization ability of this method as indicated in Tables 3–5. For all the four matrices, MLP outperforms the remaining classifiers including the SIA. Worthy of note is the poor performance of maximum entropy and NB classifiers with dismal recall and accuracy scores. This could be attributed to overfitting during training, and hence, the classifiers could not learn the necessary pattern to make generalized prediction across different data sources. However, maximum entropy and NB classifiers during the training have the best precision scores (Table 4) of all the classifiers which make them good candidates for detecting positive sentiment in a text.

Similarly, the performance, the feature extraction based on word embeddings across classifiers, is fairly competitive and outperforming SIA in some metrics such as overall accuracy and precision, as evidence in Table 3. This indicates that the proposed preprocessing and feature extraction methods are quite effective.



Input Text: The food tatses great but the place is dirty.

Positive Sentiment %: 56.67 %

Negative Sentiment %: 43.33 %

Overall Sentiment: Silghtly Positive

Figure 4: Python-based GUI implementation of the proposed method.

It is also interesting to note that, apart from positive and negative classification, strength of these expressions can also be deduced from classification scores, as it is presented as a probability distribution function. In most cases observed, the scores from the classifiers conform with human-rated evaluation. Hence, in the final model, as shown in Figure 4, we considered the absolute difference AD between positive and negative scores to interpret sentiment strength. For example, if the AD is less than say 5%, the sentiment could be interpreted as neutral or marginally positive or negative depending on which class has higher probability scores.

6. Conclusion

Opinion mining and sentiment analysis are increasingly getting a lot of tractions in the modern world and are being faced with enormous challenges due to the emergence of online fora where interactions are conducted in a highly nonstructured form. In this research, a new approach for contending with these emerging challenges and generalization problems was proposed and implemented. Experimental results indicate that the rule-based text preprocessing approach has a huge impact in handling the text from social media and also the feature extraction technique with the appropriate classifier produces better performances compared to some of the state-of-the-art methods. In the end, the two fundamental objectives of the research which include improved performance and generalization ability have been realized.

Data Availability

The "Movies Reviews" and "Twitter samples" data used to support the findings of this study have been deposited in the NLTK repository (http://www.nltk.org/nltk_data/).

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] A. Esuli, F. Sebastiani, and V. G. Moruzzi, "A publicly available lexical Resource.pdf," in *Proceedings of the 5th Conference on Language Resources and Evaluation*, pp. 417–422, Genoa, Italy, May 2006.

[2] J. Ding, H. Sun, X. Wang, and X. Liu, "Entity-level sentiment analysis of issue comments," in *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, pp. 7–13, Gothenburg, Sweden, June 2018.

[3] S. Baccianella, A. Esuli, and F. Sebastiani, "SENTIWORDNET 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*, Valletta, Malta, May 2010.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, vol. 2013, 2013.

[5] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 1, pp. 163–173, 2012, https://doi.org/10.1002/asi.21662.

[6] M. Wongkar and A. Angdresey, "Sentiment analysis using naive bayes algorithm of the data Crawler: Twitter," in *Proceedings of the 2019 Fourth International Conference on Informatics and Computing (ICIC)*, Semarang, Indonesia, October 2019.

[7] A. M. Ashir, S. Ibrahim, M. Abdulghani, A. A. Ibrahim, and M. S. Anwar, "Diabetic retinopathy detection using local extrema quantized haralick features with Long short-term memory network," *International Journal of Biomedical Imaging*, vol. 2021, pp. 1–12, 2021.

[8] L.-C. Cheng and S.-L. Tsai, "Deep learning for automated sentiment analysis of social media," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1001–1004, Vancouver, British, Columbia, Canada, August 2019.

[9] C. K. Chung and J. W. Pennebaker, "Linguistic Inquiry and word Count (LIWC)," *Applied Natural Language Processing*, vol. 4, pp. 206–229, 2012.

[10] P. J. Stone, R. F. Bales, J. Z. Namenwirth, and D. M. Ogilvie, "The general inquirer: a computer system for content analysis and retrieval based on the sentence as a unit of information," *Computers in Behavioral Science*, vol. 7, no. 4, pp. 484–498, 1962.

[11] M. Bradley and P. Lang, Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings, 1999.

[12] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[13] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," in *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, Ann Arbor, Michigan, USA, June 2014, https://sentic.net/.

[14] A. Boutet, H. Kim, and E. Yoneki, “What’s in your tweets? I know who you supported in the UK 2010 general election,” in *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, Dublin, Ireland, Jun 2012, http://www.aaai.org.

[15] J. Li and L. Qiu, “A sentiment analysis method of short texts in microblog,” in *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, vol. 1, pp. 776–779, Guangzhou, China, August 2017.

[16] Y. Woldemariam, “Sentiment analysis in a cross-media analysis framework,” in *Proceedings of the 2016 IEEE International Conference on Big Data Analysis (ICBDA)*, Hangzhou, China, March 2016, https://hadoop.apache.org.

[17] A. Jurek, M. D. Mulvenna, and Y. Bi, “Improved lexicon-based sentiment analysis for social media analytics,” *Security Informatics*, vol. 4, no. 1, 2015.

[18] A. Boutet, H. Hyoungshick Kim, and E. Yoneki, “What’s in twitter: i know what parties are popular and who you are supporting now!” in *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 132–139, Istanbul, Turkey, August 2012.

[19] S. Ahmed, K. Jaidka, and M. M. Skoric, “Tweets and votes: a four-country comparison of volumetric and sentiment analysis approaches cl-scisumm: the computational linguistics scientific summarization task view project a literature review framework for multi-document summarization of research papers view project,” in *Proceedings of the Tenth International AAAI Conference on Web and Social Media (ICWSM 2016)*, Cologne, Germany, March 2016, https://www.researchgate.net/publication/303062588.

[20] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, June, 2011.

[21] B. Pang and L. Lee, “A sentimental education: sentiment analysis using subjectivity,” in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pp. 271–278, Barcelona, Spain, July 2004.

[22] E. Loper and S. Bird, “NLTK: the natural language toolkit,” 2002, https://arxiv.org/abs/cs/0205028.

[23] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, “ O’Reilly Media, Inc., Newton, MA, USA, 2009.