This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Feature-Based Twitter Sentiment Analysis With Improved Negation Handling

Itisha Gupta and Nisheeth Joshi

***Abstract*— There is remarkable progress in the research of Twitter sentiment analysis (TSA) which is a technique of extracting opinion by automatically processing digital data. In this article, we propose a feature-based TSA system in conjunction with improved negation accounting by leveraging different types of features such as lexicon-based, morphological, POS-based, *n*-gram features, and many more, which would be used for classifier training and have the strong impact on polarity determination. We use three different state-of-the-art classifiers such as support vector machine (SVM), Naive Bayesian, and decision tree, and the series of experiments are conducted to determine which classifier works well with which feature group. In addition, this work focuses on investigating a significant linguistic phenomenon called negation which can either change polarity or strength of polarity of opinionated words. To enhance the classification performance, an algorithm is also developed to handle those negation tweets in which the presence of negation does not necessarily mean negation. The proposed feature-based Twitter system with negation accounting is evaluated on the benchmark Twitter data set SemEval-2013 Task 2. The experimental results demonstrate that the SVM classifier outperforms the other classifiers and the state-of-the-art TSA system developed by the NRC Canada winning team of SemEval-2013 Task 2. In addition, extensive experiments are also conducted to demonstrate that the proposed negation strategy with incorporated negation exception rules provides a substantial improvement by preventing misclassification of tweets. Finally, impact of each preprocessing module on classification performance is presented.**

***Index Terms*— Corpus-based approach, feature engineering, negation exception, negation handling, preprocessing, Twitter sentiment analysis (TSA).**

I. Introduction

TWITTER is one of the most famous social media platforms that allow users to express their feelings in the form of short messages. It is Twitter which gives us the prompted views of users on various products, services, and events. Among various microblogging sites, Twitter is a rich source of opinionated data because Twitter user varies from common man, politician, to big personalities. Thus, valuable information can be extracted by analyzing the Twitter data.

Twitter sentiment analysis (TSA) is one such efficient way of determining people's opinions from a piece of text. It plays a significant role in determining the opinion of people and their opinions' impact on society [9], [12]. Thus, it is an important NLP task for detecting and analyzing opinionated text, identifying sentiments. The majority of the existing works in TSA use a machine learning statistical technique that requires sufficient training corpora and have shown considerable success in various projects such as detection of fake news [33], classification of movie reviews [29], and many more [5], [10], [19]. Though deep learning is a new trend and has been used by various researchers [24] for TSA, in this work, we adopt a supervised machine learning approach because we crafted features manually. In the existing literature, a wide variety of features and techniques for TSA have been proposed with varying results. However, such methods suffer from challenges such as the polarity of opinionated text can be changed due to the presence of linguistic elements such as negation (one of the contextual shifter [2]), intensifiers (e.g., very, extremely), diminishers (e.g. hardly, barely), and discourse markers (e.g., but, although, if, until). That is sometimes reasonable accuracy is obtained and sometimes not.

Negation either reverses the polarity or changes the strength of polarity of affected opinionated words. Intensifiers basically increase the strength of polarity of affected word, that is, either make the sentiment of affected word more positive or more negative. For instance, the phrase "extremely good movie" conveys more positive sentiment than the phrase "good movie." Here, the intensifier term "extremely" is intensifying the polarity score of positive polarity word "good." Traditional approach for handling intensifier is to either increase the strength of affected polarity word by a constant value or use a percentage modification scheme [3]. In percentage scheme, a list of intensifiers is created where each intensifier is annotated with different strengths of intensification. For instance, intensifier "extremely" has more intensification strength than "very." Few earlier works [26] used the real-valued score of intensifiers (from SentiWordNet) for modifying the polarity of affected words.

On the other hand, discourse markers (such as but, although, despite, and many more) impact the segment of an opinionated text, depending on its position in the text. For instance, in case of conjunction "but," part of the text that comes after "but" is considered to be more important (high weightage). For instance, consider a tweet "Bn traveling since day of #DeMonetisation, meetin ppl frm various fields. Conclusion is "Facing problems but a great step by @narendramodi." This tweet expresses over all positive sentiment toward "demonetization" in spite of the presence of negative polarity word

Manuscript received March 11, 2020; revised March 15, 2021; accepted March 23, 2021. *(Corresponding author: Itisha Gupta.)*

The authors are with the Department of Computer Science, Banasthali Vidyapith, Vanasthali 304022, India (e-mail: itishagupta07@gmail.com; jnisheeth@banasthali.in).

Digital Object Identifier 10.1109/TCSS.2021.3069413

2329-924X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

"problems." This is due to the presence of conjunction "but," which gives more weightage to the segment "great step by @narendramodi." Most of the earlier works [26] handled the discourse markers through linguistic rules which modify the n-grams based on the position of markers.

In this work, our main focus is on handling the negation particularly as a linguistic element. Thus, we aim to handle the TSA by introducing a novel adaptable feature-based TSA system which accounts for negation and negation exception cases (negation cue is present but there is no sense of negation). The goal of this work is not only to explore various syntactic and semantic features with classifiers but also to attempt to model linguistic knowledge such as negation, which is one of the major challenges often encountered in TSA. For handling the negation, we use the corpus-based statistical approach recommended by Kiritchenko *et al.* [40], because negation does not reverse the polarity of words under scope every time. Our work is an improvement over the work done by Mohammad *et al.* [25] in negation handling with the addition of few linguistic rules for handling the cases in which negation presence does not necessarily mean negation. We extend the work of [25] arguing that there exist cases where the presence of negation does not necessarily mean negation. Though few earlier works [14], [22] discussed the negation exception cases, no one has provided the implementation of such linguistic rules. A detailed explanation of negation accounting in conjunction with the proposed algorithm for handling the negation exception cases is given in Section III. Whereas negation modeling in our work is somewhat similar to the work of [40], it is worth noting that the novelty of this work is to explore different features with classifiers and to explore negation exception rules. We will prove the effectiveness of our proposed feature-based Twitter system with improved negation modeling on the benchmark Twitter data set SemEval-2013 Task 2 [41] through experiments (see Section IV). The SemEval-2013 data set is an imbalanced data set with having a high number of neutral tweets. It might lead the biasness toward neutral class during classification. Thus, we give more weightage to minority class (negative) and less to the majority class (neutral) to handle the imbalance problem. This, in turn, would lead to an increase in the classification performance. We demonstrate the comparative analysis of three different classifiers [support vector machine (SVM), naive Bayesian (NB), and decision tree classifier (DTC)] with each feature group to determine which classifier works well with which feature group. We also present the comparative analysis of our proposed negation handling with state-of-the-art methods available for negation handling such as reverse polarity. Moreover, we observe the impact of different preprocessing modules on the best classifier performance by performing a series of experiments when modules are applied separately or jointly. The experimental results show that SVM turned out to be the best performing classifier with an accuracy of 0.726, macroaveraged recall of 0.73, and macroaveraged $F1$ score of 0.71 (69.5, when $F1$ is calculated based on positive and negative classes only). Our proposed SVM model outperforms the official submitted result of the winning team [25] of the SemEval-2013 task 2 competition. With SVM, n-grams and lexicon-based features turned out to be the most influential features. Lexicon-based features alone provide a gain of more than 8% points in the macroaveraged $F1$ score. Specifically, the contribution of automatic lexicon-based features is more than manual lexicon features. This proves the importance of handling negation by the use of automatic lexicons. Lexicon features also turned out to be the most influential for DT classifier. However, for NB classifier n-grams turned out to be the most influential because NB works best with non-negative features. We observe a gain of 2% points in the macroaveraged $F1$ score by the incorporation of negation exception rules in the SVM model (outperforms the other classifiers). We also analyze the behavior of negation exception rules with DTC and NB. We observe that incorporation of negation exception rules maintains the consistency of performance raise with other classifiers too.

The following are the main contributions of this article.

1) We present a detailed description of feature-based TSA system (incorporated with improved corpus-based negation modeling approach), which classifies tweets based on syntactic and semantic features extracted from them.
2) This work contributes in presenting a feature extraction system that would help in generation of varieties of feature sets, which can be used as an input to classifiers.
3) We provide an algorithm for implementing a set of rules for handling those tweets where negation occurrence does not necessarily mean negation.
4) We present the analysis of three different classifiers (SVM, NB, and DTC) with each feature group to determine which classifier works well with which feature group.
5) We determine the effectiveness of negation handling with incorporated negation exception rules in performance improvement of classifiers.
6) We conduct experiments on the benchmark Twitter data set SemEval-2013 to present comparison with the baseline and state-of-the-art TSA developed by the winning team of SemEval-2013 task 2 (NRC Canada). Our results show that the proposed model is robust and competitive.

The rest of this article is organized as follows. Section II presents the literature review of earlier works on TSA, Section III describes the framework of our proposed feature-based sentiment analysis system, Section IV provides the evaluation result, and Section V provides conclusion of this article with possible future directions.

II. Literature Review

In the existing literature, a lot of work has been done in the field of TSA for polarity detection. Quite often, researchers addressed the problem of TSA using a supervised learning approach [1], [7], [10], [16], [17], [22], [23], [25], [28], [34], [40], [42], [44]. In supervised sentiment analysis approach, different types of features are extracted from labeled training, then machine learning classifiers are trained on features, and finally trained classifiers are used for polarity classification of unseen test data. Different works used different classification algorithms with different features for

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

classification. Early attempt for TSA through supervised learning significant methodology was done by Go *et al*. [17], Pak and Paroubek [28], and Spencer and Uchyigit [42]. Their work is considered as notable work in the field of TSA. Pak and Paroubek [28] and Spencer and Uchyigit [42] tested their proposed technique for TSA on manually annotated 216 tweets, whereas Go *et al.* [17] evaluated on 359 manually annotated tweets. Other works exploited sophisticated features too such as Twitter-specific features [1], [7], [16], [23], [25], [40], lexicon-based features [16], [25], [26], [44], sentiment features [10], [35], and semantic features [43] apart from *n*-grams and POS-based features because feature engineering is an important step toward success of machine learning classifiers. Few earlier works [35] performed a deeper level of analysis in feature engineering and explored new feature groups such as semantic and sentiment features to resolve the data sparsity problem. Nonetheless, they stated that traditional classifiers such as SVM and NB suffer from data sparsity problems due to the unstructured nature of tweets. Recently, Carrillo-de-Albornoz *et al.* [10] provided an extensive evaluation of various features such as syntactic, semantic, lexical, network-based, and word embedding features for representing patient-authored texts for polarity classification. They presented the evaluation of various features' combinations by the use of state-of-the-art machine models such as random forest, NB, sequential minimal optimization (SMO), and vote (ensemble of various classifiers). They obtained the best results with vector-based models in combination with sentiment-based features and lexical features.

All the aforementioned works focused on the generation of different types of feature vector. Thus, it can be concluded that in standard practice, an opinionated text is represented through the bag-of-words model (BOW). That is, each text is represented as a vector of words. Though this model is efficient and quite simple, such a BOW model suffers from various issues such as ignoring the position of words, discarding semantic information, and many more. Therefore, it leads to some fundamental problems such as polarity shift. Polarity shift is an important linguistic phenomenon in which polarity of words or phrases can be reversed or changed due to linguistic elements such as negation (e.g., this is not an interesting movie) and contrast (e.g., fairly good, but not my style).

Over the years, a wide variety of approaches have been explored to model the negation problem, including rule-based, machine learning classifier [11], [34], and more recently, deep learning [31], [37], [38]. Most of the previous contributions [2]–[4], [13], [14], [19], [22], [32] used the traditional and most common reverse polarity approach for handling negation, that is, simply reverse the polarity of word or phrase under negated context. Notwithstanding, few works [26], [40] argued that the stated method is less efficient because polarity not always translates to the opposite, for example, the phrase "not excellent" is still more positive than the phrase "not good." Considering this fact, one of the noteworthy and enduring works for handling negation was done by Kiritchenko *et al.* [40] with the use of corpus-based statistical approach. They developed the two Twitter-specific lexicons (NRC Hashtag and S140 lexicon) for handling the negation, in which each word of a tweet gets the two scores. One score is for the word in negated context and another score is for the word in affirmative context. Using their negation addressing approach, they managed to obtain first position in the SemEval-2013 task 2 competitions. Thus, afterward many of the other works used their corpus-based approach for addressing negation.

In 2016, Muhammad *et al.* [26] used a different approach of shifting polarity for accounting contextual polarity negation in their work. They introduced SMARTSA lexicon-based system using hybridized SWN for capturing the local and global context of the term. They hybridized the SentiWordNet lexicon with domain-specific knowledge for handling global context. The experimental results showed significant improvement in classification due to this hybridized approach (hybridizing SWN lexicon) which takes into account contextual polarity. However, the SWN lexicon used in their work is a formal lexicon and does not contain the scores for unstructured elements of tweets such as slang, misspelled, and many more.

Xia *et al.* [39] proposed a slightly different shift elimination method for handling the polarity shift problem such as negation. Their approach is to simply eliminate the negation cue and replace the affected opinionated word by its antonym. However, again their shift elimination approach is somewhat similar to reverse polarity.

Recently, few researchers [31], [32], [37], [38] addressed negation in their deep learning framework too. For instance, Teng *et al.* [37] have done a noticeable work by presenting a weighted sum model for accounting linguistic knowledge such as negation and intensification. They used the bidirectional LSTM for negation handling and weight learning (weighted sum model). They evaluated their approach effectiveness on three data sets such as Semeval-2013, mixed domain, and movie review data set (Stanford Sentiment Treebank). It is worth noting that their negation addressing approach is much simpler and easier than the rules given by Taboada *et al*. [36].

Considering various approaches used in the literature for negation handling, we aim to fill the gap of sentiment analysis in terms of negation by augmenting the corpus-based approach for negation accounting with an algorithm for handling the tweets in which negation cue is present, but negation sense is absent. Such tweets come under negation exception cases. A detailed explanation of negation handling and algorithm for negation exception cases is given in Section III.

III. Methodology

This section presents our proposed feature-based TSA system considering improved negation handling mechanism (negation modeling is incorporated with the negation exception algorithm). We start with the tokenization and POS tagging of the public benchmark Twitter data set SemEval-2013. For this, we use a Twitter-specific tagger, namely, CMU POS tagger [15]. This tagger is designed specifically for Twitter and has the ability to capture Twitter-specific entities such as URLs, punctuations, usernames, hashtags, and many more as distinct entities. Gimpel *et al*. [15] have shown 89% tagging

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

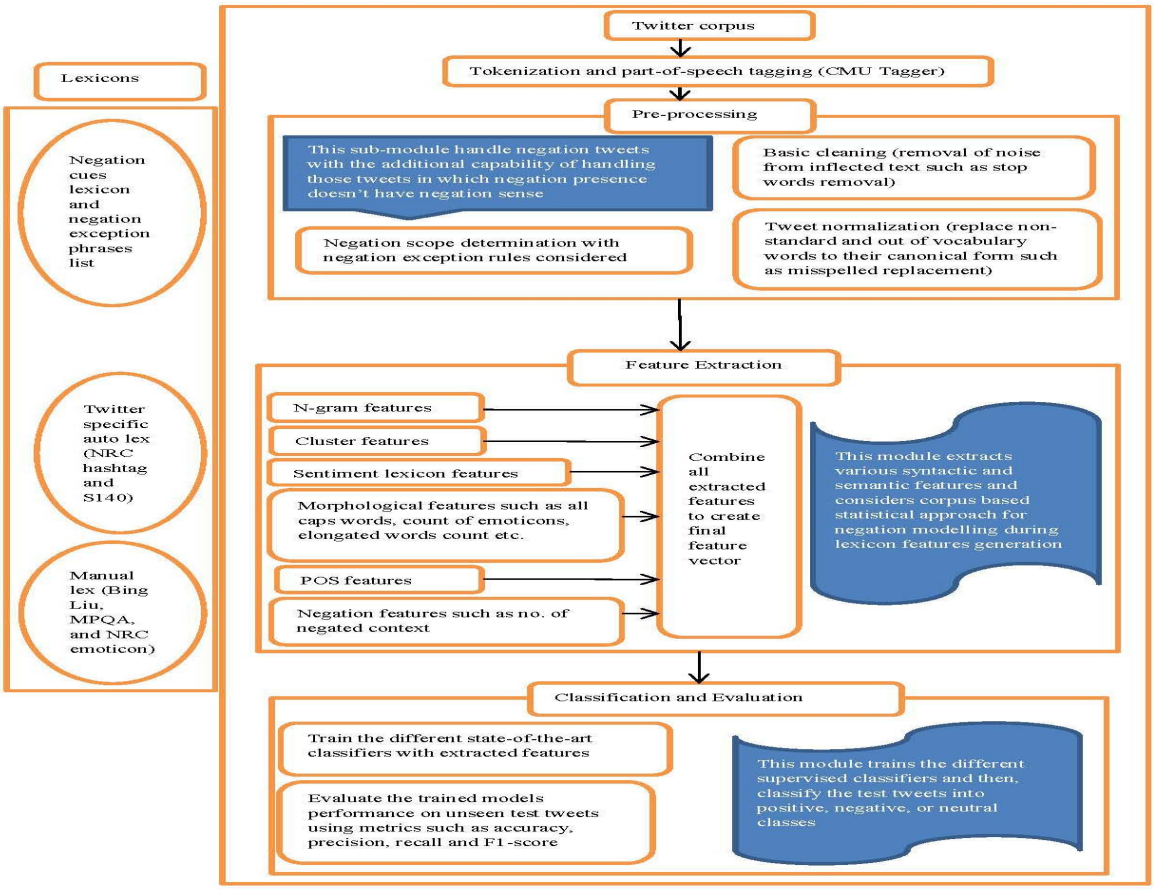

Fig. 1. Workflow of the feature-based TSA system.

accuracy by this tool. Hence, each tweet of the SemEval-2013 data set has tweet tokens and POS of each token. Then, the proposed framework is implemented in many folds such as tweet normalization, feature engineering, training of classifiers, and finally classification. Fig. 1 shows the illustration of our proposed TSA system including the preprocessing framework.

A. Tweet Normalization

Tweet normalization is a process of removal of noise from a tweet (stop words, usernames, and URLs) and normalizing the unstructured elements such as out-of-vocabulary words. There is wide usage of preprocessing in [6], [8], and [22]. For performing preprocessing of the Twitter data set, we used a framework that we have implemented in our previous work [18]. This framework consists of two phases: basic cleaning operations (noise removal) and tweet normalization (replacing nonstandard, ill-formed words, and out-of-vocabulary words to their canonical forms). So we get a clean, noise-free, and normalized Twitter corpus which would be then fed to the second phase that is feature engineering.

It is worth noting that negation scope determination (words affected by negation comes under the negation scope) is performed as a part of preprocessing itself so that Twitter-specific automatic lexicon could be used to determine the score of words under negation scope.

1) Negation: It is an important phenomenon [14], [21] that has a significant impact on classification as negation can change polarity or the strength of polarity. Our proposed TSA system deals with the most common type of negation that is syntactic negation which can be identified by explicit negation cues. Syntactic negation is the most well-known form of negation [45] and has scope associated with it (words affected by negation). The primary step toward negation handling is the identification of negation cues. For this, we analyzed the SemEval-2013 data set, and, then compiled a big list of negation words in addition to misspelled negation words too.

The second phase of negation handling is to decide the words which are impacted by negation called negation scope [11]. The scope can be a word immediate to the negation or maybe even more words. Broadly categorized, approaches for scope determination are divided into three groups, namely, rule-based methods [2], [3], [8], [10], [13], [14], [17], [22], [25], [26], [40], machine learning [11], [30], [34], and shallow semantic parsing. Importantly, most of the state-of-the art TSA systems such as SemEval task researchers [25], [40] used the simple rule-based concept of marking all words between negations till first punctuation mark to be under the scope. Simply, this approach would unnecessarily mark all words (polar and nonpolar both) between cue till the first punctuation mark.

Thus, we have used the concept of the static window of five words to be in the scope in addition to linguistic features which

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Pre-condition: Twitter corpus must passed through CMU tagger for tokenization and pos tagging
Input: List of tweet tokens, pos tokens, negation cues list, and list of negation phrases having no sense of negation.
Output: Tweet tokens with negated context words marked with "_NEG" **exempting** those negation tweets in which negation presence doesn't have negation sense.

```
1. For each (tweet Ti)
2.     For each (tweet token Wj in Ti)
3.         If (Wj in negation cue list) then
4.             If (j!=0) then
5.                 a:=Wj ; b:=Wj-1
6.             Else
7.                 a=Wj ; c=null string
8.             End If
9.             If (j!=length(Ti)-1) then
10.                b:=Wj+1
11.            Else
12.                b=null string
13.            End If
14.            d=c+a+b ; e=a+b
15.            If ((postoken(Wj)='D') or (postoken(Wj)='!')) then
16.                No negation handling would be performed. Iterate over next tweet token (continue).
17.            Else If ((postoken(Wj)='V') and (postoken(Wj+1)='O')) then
18.                No negation handling would be performed. Iterate over next tweet token (continue).
19.            Else If ((postoken(Wj)='V') and (postoken(Wj+1)='D')) and (postoken(Wj+2)='A')) then
20.                No negation handling would be performed. Iterate over next tweet token (continue).
21.            Else If ((d in negation phrase list) or (e in negation phrase list)) then
22.                No negation handling would be performed. Iterate over next tweet token (continue).
23.            Else
24.                Set negation scope to true and perform the negation scope determination using static window approach.
25.            End If
26.        Else
27.            If (negation scope is true) then
28.                Marked the word under negation scope with tag "_NEG" as long as it is adjective, verb, Noun, adverb, and hashtag.
29.                If (next adjacent word Wj+1 is in negation list) then
30.                    Set negation scope of current negation cue to false because of double negation.
31.                End If
32.            End If
33.        End If
34.    End For
35. End For
```

Fig. 2. Algorithm for modeling negation with negation exception rules.

may bound negation scope before static window termination, for example, conjunction "but" limit the negation scope to one clause only and does not let negation impact to be stretched to the next clause. Hence, the negation scope is not fixed. It may be restricted to only the next word or wider range (maximum window of five words) that depends on linguistic features. Moreover, tag "_NEG" is attached to words that are under the scope. However, the tag is only attached to nouns, adjectives, adverbs, verbs, and hashtags to avoid any unopinionated word subject to negation as negation existence does not affect all words in scope [21], and there is no sense in negating emoticons and exclamation symbols. The third and final phases of negation modeling are to handle the words which are marked by tag "_NEG" during scope determination.

We proposed the use of sophisticated corpus-based approach in conjunction with the negation exception algorithm (used to identify those negation tweets in which negation presence has no sense of negation) for addressing the negation impact. Based on the statistic that negation can change the strength of polarity too, we have used automatic lexicons provided by Kiritchenko *et al*. [40] to estimate the score of each word in negated or affirmative context rather than simply reversing the polarity of negated words. Such automatic lexicon-based features are extracted during the feature engineering phase. Put simply, we incorporate some linguistic rules (for negation exception) into the corpus-based strategy for handling the negation to prevent the misclassification. Basically, we observe two such negation exception cases from negation tweets which are described as follows.

Case 1(Negation Phrase): When negation cue is a part of the phrase (such as not just, not only, at no times, no one, by no means, and many more) in a tweet, then no negation handling would be performed. Also, most of the times negation cue POS tag would be "D" in such phrases. For example, "I am not sure about #DeMonetisation but in such rhetoric there is no one as good as him. I know this since over a https://t.co/cYG5OcdHPQ." Though it is an example of negation tweet, in the literal sense negation cue "no" does not affect the polarity of opinionated word "good." For handling this case, a big list of such phrases is created, and, then by lookup mechanism if such a phrase is found in a negation tweet, then no negation handling would be done.

Case 2(Rhetoric Question): When negation cue is present in negative rhetoric questions, then, negation handling would not be performed on that tweet. For example, "I'm in Petrolia. The sun's out, is nt that bright enough for you?" Rhetoric questions are recognized by heuristics that it ends with a question mark and has a negation cue in the first three words [21]. However, such heuristics are difficult to implement on tweet because of unstructured nature of the tweet. So to handle this case specifically in tweets, we have manually analyzed the tweets and looked for POS tags of neighboring words of negation cue in negative rhetoric questions. We observed some patterns (implemented in algorithm form) from negative rhetoric questions in which negation cue presence has no negation sense.

From the above discussion, it is apparent that handling negation exception cases is a challenging task in TSA. We devise an algorithm (as shown in Fig. 2) for handling negation exception

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

cases (mentioned above), based on the above discussion. This algorithm runs for each tweet in the Twitter corpus. However, the precondition for this algorithm is to get the tweet tokens and POS tokens for a tweet by the use of CMU tagger. Thus, the input is a collection of tweet tokens and their corresponding POS tokens. Also, we need a negation cues list and negation phrase list (list of those negation phrases in which there is no sense of negation) as the input. The output of this algorithm is the tweet tokens list in which negated context words are marked with tag "_NEG," exempting the scope for those negation tweets which satisfy the rules presented in the below algorithm. Level 1 in this algorithm is the list of tweet tokens and the corresponding POS tokens of a tweet from the collection of the entire corpus list. Level 2 checks for each tweet token from the list of level 1. If that tweet token matches with the negation cue from the negation cues list, then first we check for the negation exception cases. Lines 4–14 are to implement the negation exception case 1 (described above) and help in getting the previous and next adjacent tweet tokens for the current tweet token (only if current token matches with negation cues list). Thus, we are able to obtain two negation phrases ("$W_{j-1}W_jW_{j+1}$" and "W_j W_{j+1}" where W_j is the current negation cue word, W_{j-1} is the previous token of negation cue, and W_{j+1} is the next adjacent token to negation cue) from the current tweet under processing. Now, such phrases are looked into the negation phrase list. If a match is found, then we do not perform negation handling on that tweet. Lines 15–20 implement the negation exception case 2. Here, we check for the pattern of POS tags of negation cue and the next two adjacent tweet tokens. For instance, if the POS tag of the negation cue is "V" (verb), then we check for the POS tags of the next adjacent tokens. If the adjacent token POS tag is "D" (determiner) and the next to next token tag is "A" (adjective) (line no. 19), then we do not perform negation handling on that tweet. Similarly, we check for three more patterns using POS tags for the negation exceptions (line nos. 15–18). If none of the negation exception cases is satisfied, then the negation scope is set to true (line nos. 23 and 24) for performing scope determination. Also, if there is a negation word in the scope of the current negation cue, then we terminate the scope of current negation cue to handle double negation (line nos. 29 and 30).

B. Feature Engineering

In this phase, valuable information in the form of features is extracted from tweets. We have used a modular approach for feature engineering so that we could experiment with different options with each module. The following set of features are generated from tweets.

1) n-Gram Features: We consider the binary representation of n-grams (unigrams and bigrams), that is, we measure only the presence or absence of n-grams rather than their frequency count. This is because of tweet nature where the repetition of a word is unlikely in the tweet. Moreover, we have also used chi-squared univariate feature selection method to reduce the number of unigrams and bigrams which in turn reduce the dimensionality. Chi-square is a statistical method used to measure dependence between word and class of tweet it occurred. So chi-square value is low for the word which occurs in many classes and chi-square value is high for the word which occurs in few classes. For instances, some of the top 20 n-grams selected by chi-square are good, positive, happy, love, sorry, shit, and so on.

2) Morphological Features: They are Twitter-specific features and extracted based on linguistic peculiarities of a tweet. The following are the morphological features: count of hashtags words, count of words with all letters capitalized, presence and number of elongated words, the existence of subjective emoticons at the tweet end, count of subjective emoticons (positive, negative, neutral, extremely positive, and extremely negative), the number of exclamation marks, question marks, number of tokens containing only question marks or exclamation marks, the existence of exclamation mark at the message's end, the existence of question marks at the message's end and the existence of exclamation or question marks at the message's end, and the presence of slang.

3) POS-Based Features: The number of occurrence of each unique POS tags. Overall, 21 unique POS tags are identified.

4) Cluster-Based Features: The CMU Twitter NLP tool provides 1000 clusters which are an alternate demonstration of tweets. The presence or absence of each token from each of the 1000 clusters and the number of occurrences of each cluster are extracted as features.

5) Lexicon-Based Features: We have used two automatic Twitter-specific lexicons (Sentiment140 and NRC-Hashtag-Sentiment) and three manual lexicons (Bing Liu, MPQA, and NRC Emotion) for extracting lexicon-based features. Automatic lexicons contain unigrams and bigrams with their real-valued score in negated and affirmative context. Hence, those lexicons are specifically used for handling negation. The following set of features are generated from automatic lexicons: the number of tokens with nonzero sentiment score, the sum of sentiment scores, the maximum of the score, and the sentiment score of the last token.

The above features are generated for all tokens (four features), all positive tokens (whose score >0, 4 features), all negative tokens (whose score <0, 4 features), and for both unigrams and bigrams. Hence, a 12-D feature vector is generated for each tweet using each automatic lexicon.

Manual annotated NRC Emotion and Bing Liu lexicons [20] have no real-valued score. They just specify whether a word has a positive or negative sentiment. So we used score $+1$ for the word found in lexica and having a positive sentiment and -1 for negative sentiment word. Also, MPQA lexicon [27] has the strength of polarity in addition to positive and negative labels so we used $+1$ or -1 for weak intensity and $+2$ or -2 for strong intensity. The following set of features are generated from each manual annotated lexicon: the sum of positive scores of words in the negated context, the sum of positive scores of words in the affirmative context, the sum of negative scores of words in the negated context, and the sum of negative scores of words in the negated context.

Repeat the above four features for hashtags. Now, we have a total of eight features including hashtag features. The above eight features are generated for all cap words (8) and for

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

lowercase words (8), and for each unique POS tags (21 unique POS tags are identified by CMU tagger on our data set excluding tag "U" and "@" because tags "U" and "@" depicting URLs and usernames, respectively, are removed before feature extraction process), $21 \times 4 = 84$ features are generated. So $21 \times 4 + 8 + 8 = 100$ dimensional feature vectors are generated from each tweet using manual lexicons.

6) Negation Features: The occurrence and number of negated contexts. It is worth noting that negation affects n-grams and lexicon features.

After extracting different sets of features, they are concatenated column-wise to get the final feature vector that would be used for classifier training. It is important to discuss here that feature optimization is needed to cut down computational cost. It can be done either by feature selection or by feature extraction approach. Feature selection techniques work by selecting subset of features that epitomize most accurate picture of the data. Some of the feature selection techniques are filter, wrapper, embedded, and hybrid. The filter approach selects the subset of features based on correlation. It is a cheaper technique to deal with high-dimensional data. In this article, we have used chi-square filter technique for selecting the most informative n-grams from the pool of n-grams. On the other hand, wrapper method uses machine learning algorithm for assessing the quality of all possible subsets of features. The embedded method encompasses the merits of both filter and wrapper techniques.

Feature extraction technique works by generating new features from the original features rather than dropping irrelevant features such as principal component analysis (PCA). PCA is an unsupervised approach for dimensionality reduction (high dimensionality states large number of features in the data set), that is, conversion of points from high dimension to low dimension, which improves computational efficiency. High-dimensional data are transformed into low-dimensional components called principal components and a subset of principal components are selected which capture most percentage of variance in data.

C. Training

In this phase, three different state-of-the-art classifiers including SVM (tuned parameter is $C = 0.0001$), NB (tuned parameter is alpha $= 0.01$), and DTC (tuned parameters are: criterion $=$ "gini," max_depth $= 7$, and min_samples_split $= 10$) are trained on final feature vector generated from the feature engineering phase to prove the effectiveness of our proposed TSA system. The grid search is performed for parameter tuning of each classifier to get the optimal set of parameters. Cross-validation is done with stratified sampling so that each fold will have the same number of classes.

It is worth noting that SVM classifier takes more time to train in comparison to NB and DTC. SVM is expensive in cost and time for big data set because computation of maximum margin hyperplane depends on the square of the number of training examples. Hence, kernel-based SVM needs $O(n^2)$ computations for training and $O(nd)$ for prediction (d is the number of dimension and n is the number of instances). On the contrary, NB model turns out to be the fastest one among SVM and DTC because it takes very less time to train the NB. The run time complexity of NB is $O(nk)$, where k is the number of labels or classes, and n is the number of features. DTC works faster than SVM because we set the criterion parameter of DTC to "gini," which is computationally efficient (no need to do logarithmic calculation). Time complexity for decision tree is $O(Nkd)$, where N is the number of training examples, k is the number of features, and d is the depth of tree.

TABLE I
SEMEVAL-2013 DATA SET STATISTICS

Dataset	Positive	Negative	Neutral	Total
Training	2978 (37.14%)	1162 (14.49%)	3878 (48.37%)	8018
Test	1306 (40.84%)	484 (15.13%)	1408 (44.03%)	3198

IV. EVALUATION

In this section, we present details of our experiments conducted to prove our system's effectiveness. Also, we describe our corpus details here. We have used python 3.4 for the implementation of our framework and Scikit-learn for the various state-of-the-art classifiers.

A. Corpus

In this work, we used the publically available SemEval-2013 Twitter corpus, which is used widely for the TSA. This corpus was released during the SemEval-2013 competition Task 2: Sentiment analysis in Twitter. This includes two subtasks: one is for term level and the other one is for message-level polarity classification. Our goal is message-level polarity classification, that is, classifying a tweet into positive, negative, or neutral classes. Training and testing data were provided by the SemEval-2013 competition organizers. Table I provides a statistic on training and testing data sets, which clearly shows the imbalance nature of the data set (majority tweets are neutral).

B. Experiments

In this section, we present the results of different evaluation experiments that determine the performance of our proposed system on the SemEval-2013 corpus. The experiments also prove the advantage of our improved negation handling with incorporated exception rules over reverse polarity strategy. The performance of various classifiers is evaluated in terms of macroaveraged recall and $F1$-score as the data set is pretty imbalanced and cannot determine the classifier performance properly by accuracy alone.

Table II shows the comparative performance of the different classifiers on the SemEval-2013 test tweets only. Here, classifiers' performances are evaluated in terms of recall and $F1$, averaged on only positive and negative classes. From the below result, it is worth noting that SVM classifier turned out to be the best classifier with an accuracy of 0.726,

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
COMPARATIVE PERFORMANCE OF STATE-OF-THE-ART CLASSIFIERS ON THE SEMEVAL-2013 TEST DATA SET

Classifiers	Macro-avg. recall	Macro-avg. F1
Support Vector Machine	67.5	**69.5**
Decision Tree Classifier	66	63.5
Naive Bayesian	59.5	62.5

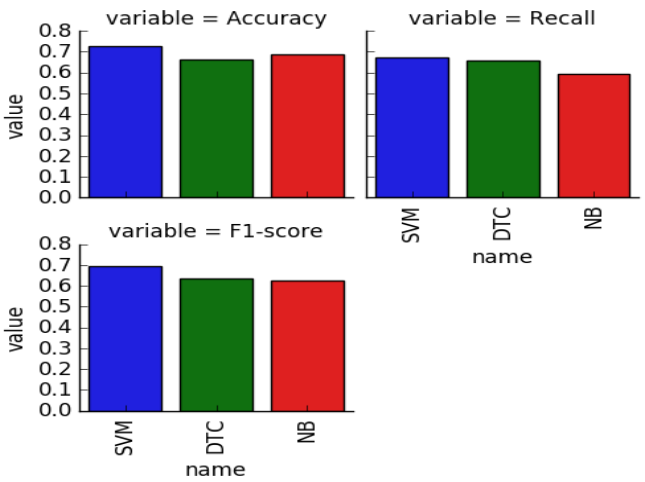

Fig. 3. Comparative performance of classifiers.

TABLE III
MACROAVERAGED $F1$ SCORE ON THE SEMEVAL-2013 TEST DATA SET

System	Macro avg. F1 score
a. Baseline (majority classifier)	28.9
b. Official SemEval-2013 submission [25]	69.02
c. Our system (SVM with all features)	69.50

average recall of 67.5, and average $F1$ of 69.5. We have used LibLinear SVM classifier as it verified effective on text classification and strong on large number of features. NB classifier obtained the lowest performance because the NB classifier works well with positive features, but in our feature set lexicon-based features have negative value too. Hence, the contributions of lexicon-based features are very low in NB classifier performance. Also, DT classifier performance is lower than the SVM. One possible explanation is that big size tree is needed for incorporating all the features and to predict the label of the test tweet; an algorithm must navigate many nodes till it reaches a leaf. This would increase the probability of mistakes and thereby decreasing the performance of DTC. Fig. 3 portrays the comparative performance of classifiers in terms of accuracy, macroaveraged recall, and $F1$-score in a single plot.

The next set of experiments (see Table III) is to present the comparison of our trained SVM model (with all features) with baseline (baseline classifier is a benchmark for other classifiers) and with the state-of-the-art system submitted officially

TABLE IV
ABLATION EXPERIMENTAL RESULTS WITH SVM WHEN ONE FEATURE GROUP IS [36] REMOVED AT A TIME

Experiments	Macro-avg. F1
a) all features	69.5
b) all-lexicons	60
b.1) all-manual	65.5
b.2) all-auto	65
c) all-clusters	68
d) all-POS	69
e) all-ngrams	68
f) all-morphological	68.5
g) all-negation	69

in the SemEval-2013 task 2 [25]. In this experiment, we have used macroaveraged $F1$ score as the evaluation metric because it was chosen to be the primary metric in SemEval-2013 task 2 [41]. Moreover, the macroaveraged $F1$ score is calculated on the basis of positive and negative classes only because the winning NRC-Canada team [25] of SemEval-2013 task 2 presented their official submitted results in terms of macroaveraged $F1$ score (averaged on positive and negative classes only). We achieved a macroaveraged F1 score of 69.5 on the tweet test set, which clearly outperforms the officially submitted result of 69.02 [25] on the SemEval-2013 task 2 tweet test set. Table III also shows the result of the baseline classifier (also known as majority classifier). The majority classifier always chooses the most frequent class, which is positive in this case. (We are not considering the neutral class in this experiment because here the macroaveraged $F1$ score is calculated based on the positive and negative classes only.) The percentage of positive tweets in the test set is 40.84% so baseline classifier accuracy is 40.84%.

We also examine to determine the contributions provided by different feature groups. Hence, experiments were carried out with each classifier by removing one feature group at a time to observe which feature works well with which classifier. Table IV presents such results of ablation experiments with SVM. From the ablation experiments' result, it is observed that with SVM classifier n-grams (row e) and lexicon-based features (row b) turned out to be the most influential features. Lexicon-based features provided a gain of 9.5% in the macroaveraged $F1$ score (averaged on positive and negative classes only). Moreover, it is interesting to note that the contribution of automatic lexicons is more than manual lexicons. Another important feature observed is cluster-based features (row c). Twitter-specific (row f) and POS-based features (row d) have little impact on SVM performance. Negation features too provided improvement in SVM performance.

Table V presents the results with DT classifier when one feature group is removed at a time. From Table V, we observed lexicon-based features to be the most significant. The second most important turned out to be the morphological features

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
ABLATION EXPERIMENTAL RESULTS WITH DTC WHEN ONE FEATURE GROUP IS REMOVED AT A TIME

Experiments	Macro-avg. F1
a) all features	63.5
b) all-lexicons	47.5
b.1) all-manual	57
b.2) all-auto	61
c) all-clusters	62.5
d) all-POS	62.5
e) all-ngrams	62.5
f) all-morphological	62
g) all-negation	63.5

TABLE VI
ABLATION EXPERIMENT RESULTS WITH NB CLASSIFIER WHEN ONE FEATURE GROUP IS REMOVED AT A TIME

Experiments	Macro-avg. F1
a) all features	62.5
b) all-lexicons	61.5
b.1) all-manual	61.5
b.2) all-auto	62.5
c) all-clusters	61.5
d) all-POS	62.5
e) all-ngrams	61
f) all-morphological	63
g) all-negation	62.5

that drop the average $F1$ score by 1.5% points. Rest of the features provide negligible contribution.

Table VI describes the ablation experiments' results with the NB classifier. From Table VI, it is clear that n-gram features (row e) are the most significant features for NB classifier as they improved the average $F1$-score by 1.5% points. Cluster features (row c) turned out to be the second most important. Lexicon features (row b) provided only a small improvement in performance because NB works best with non-negative features. The contribution of morphological and POS-based features is negligible.

Now to prove the significance of our proposed negation exception rules, we carried out experiments with all three trained classifiers. Table VII depicts the comparison of our trained SVM model with (row a) or without (row b) negation exception rules in terms of macroaveraged recall and $F1$. It is clear from the experimental results that removal of negation exception rules results in drop in recall and $F1$-score in all the three classifiers. There is a drop in recall and $F1$ by 2% points in the SVM model when negation exception rules are removed.

TABLE VII
SIGNIFICANCE OF NEGATION EXCEPTION RULES

System	Macro-avg. recall	Macro avg. F1
a) SVM (without negation exception rules)	65.5	67.5
b) SVM (our system)	67.5	**69.5**
c) DTC (without negation exception rules)	64.5	61.5
d) DTC (our model)	66	**63.5**
e) NB(without negation exception rules)	59.5	61.5
f) NB (our model)	59.5	**62.5**

TABLE VIII
COMPARISON OF DIFFERENT NEGATION PROCESSING STRATEGIES

System	Macro-avg. recall	Macro avg. F1
a) Auto base lexicon		
a.1) disregard negation	64	66
a.2) reverse polarity (without negation exception rules)	64.5	66.5
a.3) reverse pol. (with neg. exception rules)	66	**68.5**
b) Our System (SVM)	67.5	**69.5**

In addition, experiments were also carried out to demonstrate the effectiveness of our proposed negation handling strategy with integrated negation exception rules by presenting a comparative analysis of our proposed negation strategy with reverse polarity strategy. In carrying out those experiments, automatic base lexicons are taken into consideration and disregarding negation is considered as a baseline for proving the effectiveness of the proposed strategy. We carried out this fold of experiment with our best SVM model. Table VIII shows the result of reverse polarity negation strategy (supervised setting row a.2) in comparison to our proposed one (row b) and proves the effectiveness of our proposed negation handling (with integrated exception rules) over the traditional reversing polarity strategy. From the results, it is observed that reverse polarity and our proposed strategy outperform the baseline of disregarding negation (a.1). Moreover, we obtained significant improvement by the use of automatic lexicon S140 and NRC-Hashtag in our proposed strategy (row b) over the reverse polarity on automatic base lexicons (row a.2). It is worth noting that when negation exception rules are removed from reverse polarity strategy (row a.3), then $F1$ score further drops by approximately 2% points which further proves the significance of our proposed rules.

Finally, we present the impact of different preprocessing modules on trained SVM classifier performance. In this experiment, we are not considering the chi-square feature selection method to get the impact of each preprocessing module

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IX
ABLATION EXPERIMENT WITH SVM WHEN ONE PREPROCESSING MODULE IS REMOVED AT A TIME

Experiments	Macro-avg. Recall	Macro avg. F1
a) all pre-processing	70	69
b) all-basic cleaning	68	68
c) all-stop words removal	68	68
d) all-elongated word norm.	69	69
e) all-misspelled replacement	70	69
f) all-emoticon replacement	70	69
g) all-negation replacement	69	68

properly. Thus, we are using all the n-gram features, not the top 300 selected by chi-square. We conducted a series of experiments with each preprocessing module independently to determine their significance in improving the performance of the SVM classifier. Table IX shows the impact of each preprocessing module when one module is removed at a time. From the results, we observed that basic cleaning operations (row b) and stopwords removal (row c) affect the SVM performance the most. Negation cues replacement (row g) also turned out to be the influential one in the performance gain. Elongated word normalization and emoticons contribute a little bit in improving the SVM performance. Replacement of misspelled words has no positive impact on classification performance.

V. CONCLUSION

This article contributes in presenting a comprehensive research in the field of TSA by looking into the critical aspects of NLP that are tweet normalization and negation. A blend of heuristics and machine learning approach is presented for feature-based TSA and a series of ablation experiments have been conducted for compressively studying the strength of several state-of-the-art classifiers (SVM, DTC, and NB) on syntactic and semantic features generated from tweets. This design is incorporated with improved negation modeling approach, which in turn improves the classification performance of semantic and syntactic-based classifiers and produces significant results. SVM was found to be the best performing classifier from the experimental results. We managed to outperform the officially submitted result of the winning team of SemEval-2013 task 2 competition (message-level classification) with our proposed SVM model. n-grams and lexicon-based features turned out to be the most influential in performance improvement of SVM. Lexicon-based features are generated specifically for showing negation effectiveness. Interestingly, comprehensive evaluation of different classifiers helps other researchers in using those best classifiers on a set of features.

Specifically, this article contributes in presenting a negation exception algorithm for handling those tweets in which negation word presence does not necessarily mean negation. We observed substantial improvement of our feature-based model over baseline classifiers due to our proposed negation exception rules. Moreover, we demonstrate the effectiveness of our proposed negation modeling strategy over the most common and traditional reverse polarity strategy. Thus, it can be concluded that our proposed negation handling strategy that uses corpus-based approach in addition to negation exception rules turned out to be most important in performance raise.

In the future, we would explore in the direction of feature engineering (optimizing feature vector size to cut down computational cost) and morphological negation handling. We would improve the negation handling approach by treating each explicit negation cue differently during negation replacement. Finally, we would focus on another type of polarity shifters too such as intensifiers, conditionals, and diminishers.

REFERENCES

[1] A. Agarwal *et al.*, "Sentiment analysis of Twitter data," in *Proc. LSM*, Portland, OR, USA, 2011, pp. 30–38.
[2] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Comput. Intell.*, vol. 22, no. 2, pp. 110–125, May 2006.
[3] M. Z. Asghar, A. Khan, S. Ahmad, M. Qasim, and I. A. Khan, "Lexicon-enhanced sentiment analysis framework using rule-based classification scheme," *PLoS ONE*, vol. 12, no. 2, Feb. 2017, Art. no. e0171649.
[4] M. Z. Asghar *et al.*, "T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme," *Expert Syst.*, vol. 35, no. 1, Feb. 2018, Art. no. e12233.
[5] R. Sequeira, A. Gayen, N. Ganguly, S. K. Dandapat, and J. Chandra, "A large-scale study of the Twitter follower network to characterize the spread of prescription drug abuse tweets," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1232–1244, Dec. 2019.
[6] Y. Bao *et al.*, "The role of pre-processing in Twitter sentiment analysis," in *Proc. ICIC*, Taiyuan, China, 2014, pp. 615–624.
[7] L. Barbosa and J. Feng, "Robust sentiment detection on Twitter from biased and noisy data," in *Proc. COLING*, Beijing, China, 2010, pp. 36–44.
[8] C. Bhadane, H. Dalal, and H. Doshi, "Sentiment analysis: Measuring opinions," *Procedia Comput. Sci.*, vol. 45, pp. 808–814, 2015.
[9] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.
[10] J. Carrillo-de-Albornoz, J. Rodríguez Vidal, and L. Plaza, "Feature engineering for sentiment analysis in e-health forums," *PLoS ONE*, vol. 13, no. 11, Nov. 2018, Art. no. e0207996.
[11] I. G. Councill *et al.*, "What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis," in *Proc. NeSp-NLP*, Uppsala, Sweden, 2010, pp. 51–59.
[12] M. Ebrahimi, A. H. Yazdavar, and A. Sheth, "Challenges of sentiment analysis for dynamic events," *IEEE Intell. Syst.*, vol. 32, no. 5, pp. 70–75, Sep. 2017.
[13] I. El Alaoui, Y. Gahi, R. Messoussi, Y. Chaabi, A. Todoskoff, and A. Kobi, "A novel adaptable approach for sentiment analysis on big social data," *J. Big Data*, vol. 5, no. 1, p. 12, Mar. 2018.
[14] U. Farooq, "Negation handling in sentiment analysis at sentence level," *J. Comput.*, vol. 12, no. 5, pp. 470–478, Sep. 2017.
[15] K. Gimpel *et al.*, "Part-of-speech tagging for Twitter: Annotation, features, and experiments," in *Proc. HLT*, Portland, OR, USA, 2011, pp. 42–47.
[16] S. Giorgis *et al.*, "aueb. Twitter. sentiment at SemEval-2016 Task 4: A weighted ensemble of SVMs for Twitter sentiment analysis," in *Proc. SemEval*, San Diego, CA, USA, 2016, pp. 96–99.
[17] A. Go *et al.*, "Twitter sentiment classification using distant supervision," CS224N Project Rep., Stanford, CA, USA, Dec. 2009, p. 12, vol. 1, no. 12.
[18] I. Gupta and N. Joshi, "Tweet normalization: A knowledge based approach," in *Proc. Int. Conf. Infocom Technol. Unmanned Syst. (Trends Future Directions) (ICTUS)*, Dubai, UAE, Dec. 2017, pp. 157–162.
[19] H. Han, Y. Zhang, J. Zhang, J. Yang, and X. Zou, "Improving the performance of lexicon-based review sentiment analysis method by reducing additional introduced sentiment bias," *PLoS ONE*, vol. 13, no. 8, Aug. 2018, Art. no. e0202523.

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[20] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Seattle, WA, USA, 2004, pp. 168–177.
[21] L. Jia, C. Yu, and W. Meng, "The effect of negation on sentiment analysis and retrieval effectiveness," in *Proc. 18th ACM Conf. Inf. Knowl. Manage. (CIKM)*, Hong Kong, 2009, pp. 1827–1830.
[22] O. Kolchyna, T. T. P. Souza, P. Treleaven, and T. Aste, "Twitter sentiment analysis: Lexicon method, machine learning method and their combination," 2015, *arXiv:1507.00955*. [Online]. Available: http://arxiv.org/abs/1507.00955
[23] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the OMG!," in *Proc. ICWSM*, Barcelona, Spain, 2011, pp. 1–4.
[24] D. Mahata, J. Friedrichs, R. R. Shah, and J. Jiang, "Detecting personal intake of medicine from Twitter," *IEEE Intell. Syst.*, vol. 33, no. 4, pp. 87–95, Jul. 2018.
[25] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets," in *Proc. SemEval*, Atlanta, GA, USA, 2013, pp. 321–327.
[26] A. Muhammad, N. Wiratunga, and R. Lothian, "Contextual sentiment analysis for social media genres," *Knowl.-Based Syst.*, vol. 108, pp. 92–101, Sep. 2016.
[27] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process. (HLT)*, Vancouver, BC, Canada, 2005, pp. 347–354.
[28] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. LREc*, vol. 10, 2010, pp. 1320–1326.
[29] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. EMNLP*, Philadelphia, PA, USA, 2002, pp. 79–86.
[30] N. Pröllochs, S. Feuerriegel, and D. Neumann, "Negation scope detection in sentiment analysis: Decision support for news-driven trading," *Decis. Support Syst.*, vol. 88, pp. 67–75, Aug. 2016.
[31] Q. Qian *et al.*, "Linguistically regularized LSTMs for sentiment classification," in *Proc. ACL*, Vancouver, BC, Canada, 2017, pp. 1679–1689.
[32] P. Ray and A. Chakrabarti, "A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis," *Appl. Comput. Informat.*, to be published, doi: 10.1016/j.aci.2019.02.002.
[33] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, and E. Cambria, "Supervised learning for fake news detection," *IEEE Intell. Syst.*, vol. 34, no. 2, pp. 76–81, Mar. 2019.
[34] J. Reitan *et al.*, "Negation scope detection for Twitter sentiment analysis," in *Proc. WASSA*, Lisboa, Portugal, 2015, pp. 99–108.
[35] H. Saif, Y. He, and H. Alani, "Semantic smoothing for Twitter sentiment analysis," in *Proc. ISWC*, Bonn, Germany, 2011, pp. 1–4.
[36] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011.
[37] Z. Teng, D. T. Vo, and Y. Zhang, "Context-sensitive lexicon features for neural sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 1629–1638.
[38] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, Paris, France, 2018, pp. 1165–1174.
[39] R. Xia, F. Xu, J. Yu, Y. Qi, and E. Cambria, "Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis," *Inf. Process. Manage.*, vol. 52, no. 1, pp. 36–45, Jan. 2016.
[40] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *J. Artif. Intell. Res.*, vol. 50, pp. 723–762, Aug. 2014.
[41] P. Nakov *et al.*, "SemEval-2013 task 2: Sentiment analysis in Twitter," in *Proc. SemEval*, Atlanta, GA, USA, 2013, pp. 312–320.
[42] J. Spencer and G. Uchyigit, "Sentimentor: Sentiment analysis of Twitter data," in *Proc. SDAD@ ECML/PKDD*, 2012, pp. 56–66.
[43] H. Saif, Y. He, M. Fernandez, and H. Alani, "Contextual semantics for sentiment analysis of Twitter," *Inf. Process. Manage.*, vol. 52, no. 1, pp. 5–19, Jan. 2016.
[44] I. Gupta and N. Joshi, "Enhanced Twitter sentiment analysis using hybrid approach and by accounting local contextual semantic," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1611–1625, Sep. 2019.
[45] Y. Choi and C. Cardie, "Learning with compositional semantics as structural inference for subsentential sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Honolulu, HI, USA, 2008, pp. 793–801.

Itisha Gupta received the Ph.D. degree in computer science from Banasthali Vidyapith, Vanasthali, India.

She is currently a Research Scholar with the Department of Computer Science, Banasthali Vidyapith. Her areas of interests are data analysis, natural language processing, and machine learning.

Nisheeth Joshi is currently an Associate Professor with the Department of Computer Science, Banasthali Vidyapith, Vanasthali, India. He has over 12 years of teaching experience. His primary works involve machine translation, information retrieval, and cognitive computing.

Authorized licensed use limited to: San Francisco State Univ. Downloaded on June 16,2021 at 05:04:56 UTC from IEEE Xplore. Restrictions apply.