Received: 25 July 2016 | Revised: 3 May 2017 | Accepted: 8 July 2017

DOI: 10.1111/exsy.12233

WILEY Expert Systems

ARTICLE

T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme

Muhammad Zubair Asghar[1] | **Fazal Masud Kundi**[1] | **Shakeel Ahmad**[2] | **Aurangzeb Khan**[3] | **Furqan Khan**[1]

[1] Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan, Pakistan

[2] Faculty of Computing and Information Technology in Rabigh (FCITR), King Abdul Aziz University, Jeddah, Saudi Arabia

[3] Department of Computer Science, University of Science and Technology, Bannu, Pakistan

Correspondence
Muhammad Zubair Asghar, Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan 29050, Pakistan.
Email: zubair@gu.edu.pk

Abstract

Of the many social media sites available, users prefer microblogging services such as Twitter to learn about product services, social events, and political trends. Twitter is considered an important source of information in sentiment analysis applications. Supervised and unsupervised machine learning-based techniques for Twitter data analysis have been investigated in the last few years, often resulting in an incorrect classification of sentiments. In this paper, we focus on these issues and present a unified framework for classifying tweets using a hybrid classification scheme. The proposed method aims at improving the performance of Twitter-based sentiment analysis systems by incorporating 4 classifiers: (a) a slang classifier, (b) an emoticon classifier, (c) the SentiWordNet classifier, and (d) an improved domain-specific classifier. After applying the preprocessing steps, the input text is passed through the emoticon and slang classifiers. In the next stage, SentiWordNet-based and domain-specific classifiers are applied to classify the text more accurately. Finally, sentiment classification is performed at sentence and document levels. The findings revealed that the proposed method overcomes the limitations of previous methods by considering slang, emoticons, and domain-specific terms.

KEYWORDS

emoticons, hybrid, microblog, sentiment analysis, slang, Twitter

1 | INTRODUCTION

The Web is a huge repository of facts and opinions available to people around the world for expressing views about a particular product, service, issue, or policy. With the rapid increase in microblogging sites such as Twitter, consumers are now relying on these services for their purchase options as are businesses for quick access to customer feedback. Microblog sites differ from conventional blogging sites in that their posts are short (140–200 characters). Tumblelogs was the first microblogging service, which appeared in 2005. Twitter, Tumblr, FriendFeed, and Plurk (2008) have since emerged as popular microblogging websites. Twitter uses the term “tweet” for its short message.

In the last few years, Twitter-based sentiment analysis applications have received considerable attention from online consumers desiring information about a product and companies needing to respond quickly to user opinions. However, due to several unique characteristics of Twitter data, it is difficult to analyse the text using previous approaches to determine sentiments from tweets. Therefore, it is an important task to develop a unified method to extract and analyse Twitter data through the automatic classification of tweets as positive, negative, or neutral, with emphasis on a domain-specific paradigm.

The main challenges faced in developing such applications include the following: (a) slang and abbreviations, irregular and insufficient words expressed by users in their posts; (b) emoticons, limited lexicons of emoticons, resulting in low accuracy of the classifier in detecting tweet polarity; and (c) domain-specific words, limited coverage of domain-specific words in the existing general-purpose lexicons, such as SentiWordNet (SWN), which often results in incorrect scoring and classification of sentiments.

The aforementioned issues often result in incorrect detection and classification of user sentiments expressed in Twitter posts. Therefore, it is an important task to develop a method to detect and analyse user sentiments from Twitter through the automatic classification of tweets as +ive, −ive, or neutral.

The main inspiration for this work is the Twitter-based opinion mining approach suggested by (Khan, Bashir, & Qamar, 2014), which classifies tweets using several classification algorithms. That study attempted to more effectively classified tweets by passing them through a pipeline of classifiers: (a) emoticon, (b) improved polarity, and (b) SWN-based. A recent study (Masud, Khan, Ahmad, & Asghar, 2014) addressed the challenges of sentiment analysis in Twitter data and proposed an efficient method of tweet classification as +ive, −ive, or neutral that considers emoticons and opinion words using different types of lexicons.

In this work, we propose an integrated framework for sentiment analysis of Twitter data using a hybrid classification scheme. As such, we primarily focus on accurately classifying slang, emoticons, opinion words, and domain-specific language for the sentiment detection and classification of tweets in multiple domains. The proposed technique is inspired by previous studies on Twitter sentiment analysis (Asghar, Khan, Ahmad, Qasim & Khan, 2017; Khan et al., 2014; Masud et al., 2014; Prieto, Matos, Alvarez, Cacheda, & Oliveira, 2014) Those studies have used supervised and unsupervised classification schemes to detect and classify the sentiments expressed by Twitter users into +ive, −ive, or neutral classes. However, we propose a hybrid approach using a slang classifier (SC), emoticon classifier (EC), and general-purpose sentiment classifier (GPSC) in a step-wise fashion to classify the reviews more accurately. Finally, we classify a tweet using an improved domain-specific classifier (IDSC) to assign accurate sentiment scores to domain-specific words, which is one of the major issues omitted from earlier studies. A tweet is classified as +ive, −ive, or neutral on the basis of results produced by the GPSC and IDSC. This assists in improving the performance of Twitter sentiment classification in multiple domains.

This research aims to improve the performance of Twitter-based sentiment classification by introducing a hybrid classification scheme. The main contribution of the proposed work is the development of an integrated sentiment classification system based on a set of newly developed classifiers: SC, EC, SWN classifier, and domain-specific classifier. The following is a synopsis of the contributions presented in this work.

- The development of an SC to detect and classify slang terms expressed in the review text by creating an enhanced slang dictionary.
- The development of an enhanced EC to detect and classify emoticons present in the review text by creating an extended emoticon dictionary.
- The creation of a GPSC for classifying tweets using part of speech (POS) tagging to extract all senses of a word for a particular POS and assigning polarity scores to each of the sentiment-bearing words retrieved from SWN. Word sense disambiguation is performed at a basic level by aggregating multiple senses of a word with a specific POS tag. The GPSC performs sentiment classification at the word and sentence levels without considering any domain-specific word sentiment.
- The introduction of a new domain-specific classifier to detect and classify domain-specific words using a probability-based measure and to assign updated sentiment scores to those domain-specific words that are not available in the SWN lexicon by proposing a revised term weighting scheme.
- The proposal and implementation of a unified framework for Twitter sentiment analysis applications using a hybrid scheme of classification that can be used by both academia and industry. The proposed framework consists of three major components: data acquisition, preprocessing, and sentiment classification. The sentiment classification module is further composed of four submodules: SC, EC, SWN-based classifier, and domain-specific classifier.

The performance of the proposed sentiment classification scheme for tweets is tested on different datasets in multiple domains, and it produces results with improved accuracy, precision, recall, and F-measure. The rest of the paper is structured as follows. Section 2 presents a literature review. In Section 3, we describe the proposed method. The experiment design is presented in Section 4. The final section summarizes the work, with a discussion on possible future expansions.

2 | RELATED WORK

There are several studies regarding the analysis of users' sentiments posted on Twitter, with a focus on classifying the tweets as positive, negative, or neutral.

It was observed by Khan et al. (2014) that a pipeline of classifiers using a hybrid classification scheme can improve the accuracy of microblog classification. The focus of the study was on various sentiment-related issues: (a) preprocessing, (b) emoticons, and (c) SWN-based sentiment scoring. After applying preprocessing techniques, the input text is passed through different classifiers, such as an EC and an SWN-based classifier.

They achieved higher accuracy than the methods offered for comparison. However, further improvement can be made by incorporating domain-specific words, slang, and extended set of emoticons.

Masud et al. (2014) proposed a lexicon-centric approach that combines different lexicons and dictionaries for the sentiment analysis of tweets. Their main emphasis was on the accurate classification of sentiments with respect to slang in the tweets. Their proposed method has different modules, namely, (a) tweet capturing and filtering, (b) subjectivity detection, and (c) sentiment scoring. These modules are supported by different lexicons, such as the opinion lexicon, WordNet, SWN, and emoticon repositories. They achieved 92% accuracy in binary classification and more than 85% in multi-class classification. However, the system needs improvement in terms of precision in the −ive class and recall in the neutral class. *Moreover, there was no provision for handling extended set of emoticons, slang, or domain-specific words in multiple domains.*

Asghar et al. (2017), in their work on sentiment classification, proposed a lexicon-based method to extract, preprocess, and classify user sentiments from online communities. They used different lexicons, including SWN and user-defined dictionaries, to determine the polarity scores of sentiment words. In order to classify user's reviews efficiently, they proposed using different classifiers: (a) an SWN-based classifier, (b) a modifier and negation classifier, and (c) a domain-specific classifier. In addition to the relationships among words, domain-specific words were also extracted to resolve the issue of domain dependency. *The major limitations of their approach included lack of handling compound sentences, and a lack of slang and sarcastic sentences, which, if incorporated, can result in enhanced sentiment classification*.

In another work, Prieto et al. (2014) collected tweets on the basis of different query terms, such as "flu," "depression," "pregnancy," and "eating disorders," and applied specially crafted regular expressions and machine learning algorithms for feature selection and classification to monitor public concern and disease information in Portugal and Spain. They achieved F-measure values of approximately 0.8 and 0.9, which are quite promising compared to the baseline methods. *However, the system depends on the labelled training dataset, and there is no support for classification of emoticons, slang, and domain-dependent terms in multiple domains*.

The unsupervised learning approach is another important direction in the sentiment analysis of Twitter data (Montejo-Ráez, Martínez-Cámara, Martín-Valdivia, & Urena-Lopez, 2012). To compute sentiment scores at the word level, the authors used SWN, and a random walk technique was proposed to analyse the weighting of tweets. The proposed unsupervised algorithm has no dependency on the labelled training dataset and has shown major improvement over the baseline methods. The limitations included a lack of negation handling, manual annotation of tweets, and inconsistencies in the computation of the final sentiment score.

Internet slang has a strong impact on the accuracy of Twitter-based sentiment analysis applications. To address this issue, Kundi, Ahmad, Khan, and Asghar (2014) presented a framework to detect and score the slang in tweets using different polarity lexicons such as SWN and other sentiment resources. They achieved better results compared with the baseline methods. The major limitations of their work include insufficient concentration on handling emoticons and the need for more sophisticated context-aware and sentiment-sensitive spell correction modules. Moreover, they did not address the issue of domain-specific words, which makes the system less effective.

To investigate the relationship between the Twitter platform and stock returns, the authors of (Ranco, Aleksovski, Caldarelli, Grčar, & Mozetič, 2015) reported that Twitter has a significant impact on stock market prices. To analyse Twitter text, the "event study" economic technique was adopted for automatic identification of events as Twitter-based volume loads. It assists in analysing the +ive and −ive sentiments expressed during the loads. Finally, "event study" was applied to identify the relationship between tweets and stock values. The main limitation of the work was the lack of emoticon and slang modules for more accurate sentiment classification of tweets in stock market prediction.

Ribeiro, Weigang, and Li (2015) proposed a unified approach for performing Tweet-based sentiment analysis. The proposed approach is composed of four modules: (a) data collection, (b) noise reduction, (c) lexicon generation, and (d) sentiment classification. They introduced four algorithms to implement the aforementioned modules. The results obtained from the experiments conducted on the "iPhone 6" dataset demonstrated that the proposed approach is more effective than similar methods. However, to achieve more efficacy, further experiments are required on larger datasets with tweet scraping in streaming mode.

A propagation-centric sentiment analysis approach for Twitter was proposed by (Tang, Nobata, Dong, Chang, & Liu, 2015). It aims at the integration of different emotional clues into a unified model and trains on both tagged and untagged datasets by switching the propagation phenomenon alternately. The experiments conducted on multiple datasets demonstrated the effectiveness of the proposed approach. The proposed method is based on general-purpose learning and can be enhanced to classify domain-specific words in different domains.

In their work on contextual sentiment analysis, Muhammad, Wiratunga, and Lothian (2016) proposed a lexicon-enhanced polarity classification technique to compute contextual polarity at different levels. They exploited the contextual features at local and global levels. However, the SWN-based sentiment scoring technique produces incorrect polarity scores for domain-specific words. To overcome this limitation, a rich collection of domain-specific vocabulary is required to improve the performance of sentiment classification.

Saif, He, Fernandez, and Alani (2016) presented a lexicon supported technique for Twitter-based data analysis that captures the sentiment class of words in multiple contexts and updates the sentiment scores accordingly. Their approach is based on co-occurrence word signals in different domains at both the tweet and entity levels. The proposed approach was evaluated using three Twitter datasets and achieved 4–5% higher accuracy than the comparison methods. However, the system was not enriched with emoticon and slang classifications.

Poria, Cambria, and Gelbukh (2015) proposed a convolutional multiple kernel learning-based method for the sentiment analysis of short multimedia content, such as text and audio and video clips. Features are extracted from the multimedia content by applying activation values in the inner layer of a deep convolutional neural network model. The results show that an improvement of about 14% is achieved over the baseline methods.

Taboada, Brooke, Tofiloski, Voll, and Stede (2011) performed sentiment analysis of user reviews using a lexicon-based scheme. The proposed method is based on the analysis of text subjectivity, considering intensifiers, negations, and opinion words. They achieved better results in terms of opinionated and nonopinionated sentences. Furthermore, they created an annotated sentiment dictionary. However, their method contains no provision for analysing slang and domain-dependent words.

The previous studies (Asghar et al., 2017; Khan et al., 2014; Kundi et al., 2014; Masud et al., 2014; Prieto et al., 2014) on Twitter sentiment analysis used supervised (Gu & Sheng, 2017; Gu, Sheng, & Li, 2015; Gu, Sheng, Tay, Romano, & Li, 2015; Wen, Shao, Xue, & Fang, 2015; Xia, Wang, Sun, Liu, & Xiong, 2016) and unsupervised (Muhammad et al., 2016; Taboada et al., 2011; Zheng, Jeon, Xu, Wu, & Zhang, 2015) learning algorithms for the detection of tweet polarity. The supervised machine learning-based classifiers are trained on tweet datasets using different features, such as

N-grams, POS tagging, and emoticons, whereas the unsupervised approaches are mainly dependent on a set of lexicons. Although recent methods for Twitter data classification have achieved relatively good results, they exhibit various deficiencies: (a) insufficient consideration of slang in sentiment classification of tweets, (b) limited coverage of emoticons, and (c) incorrect scoring of domain-specific words, as the existing general-purpose lexicons, such as SWN, may assign incorrect scores to domain-specific words.

These problems occur for three main reasons: (a) use of abbreviations and slang words due to the length restriction of a tweet (140 characters), (b) lack of sufficient use of emoticons in text classification, and (c) limited coverage of domain-specific words in the existing general-purpose lexicons, such as SWN, which may assign incorrect scores to most of the domain-specific words. The polarity score of an opinion word often changes with a change in domain, whereas SWN generally assigns scores to opinion words without considering their respective domain. Solving these problems leads to the development of a unified framework for Twitter sentiment analysis using a hybrid classification scheme with an emphasis on classifying and scoring slang, emoticons, and domain-specific words in a more effective way.

The proposed method is based on a hybrid scheme of sentiment classification supported by different lexical repositories, such as slang lexicons, emoticon dictionaries, the SWN lexicon, and a supervised learning-based domain-specific classification module. The key addition to the current state-of-the-art methods (Asghar et al., 2017; Khan et al., 2014; Kundi et al., 2014; Masud et al., 2014) is the classification of slang, emoticons, general-purpose words, and domain-specific terms in a pipelined manner. Our system can automatically detect and classify slang, emoticons, general-purpose opinion words, and domain-specific opinion words in tweets. That is, slang terms are automatically detected and classified using our newly proposed SC; emoticons are classified using our proposed extended emoticon repository and EC; and sentiment words are assigned sentiment classes and scores by passing them through SWN-based and probability-based domain-specific classifiers. The enhancements made in the SC, EC, and domain-specific classifiers are our major contributions to the state-of-the-art methods for accurate sentiment classification.

From the overview of recent studies (Table 1), the existing supervised approaches for sentiment classification are dependent upon the availability of large annotated datasets, which results in performance degradation. The problem with unsupervised techniques is due to (a) their dependence upon publically available sentiment lexicons and (b) their use of basic sentiment dictionaries with limited support for emoticons, slang, and domain-specific words. Therefore, a more accurate and precise classification method is needed that can more effectively classify tweets. To address this gap, more work is required to design an efficient Twitter-based sentiment analysis system using a hybrid scheme of classification with an enhanced set of slang words and emoticons, and a pipeline of sentiment classifiers with the ability to accurately classify general-purpose and domain-specific opinion words while simultaneously achieving more-robust results that are comparable to the performance results of supervised and unsupervised approaches.

TABLE 1 Overview of selected studies

Study	Methods	Type	Results	Limitations
Khan et al. (2014)	SentiWordNet Emoticons, sentiment words	Hybrid	85% Accuracy	Lack of domain specific words Lack of slang classification
Masud et al. (2014)	SentiWordNet Opinion lexicon	Unsupervised Lexicon-based	85% Accuracy	Lack of provision for handling: emoticons, slangs, and domain-specific words in multiple domains
Asghar et al. (2017)	SentiWordNet Emoticon Dictionary Modifier and negation classifier	Unsupervised Lexicon-based	88% Accuracy	lack of handling slangs limited set of emoticons use of manual annotation scheme in handling domain specific words
Prieto et al. (2014)	Regular Expression	Supervised learning	0.8 (F-score)	dependency on the labelled training dataset, no support for classification of ○ emoticons, ○ slangs, and ○ domain-dependent terms in multiple domains
Montejo-Ráez et al. (2012)	SentiWordNet Personalized page rank vectors Emoticons	Unsupervised Random walk	0.628 (F-score)	lack of negation handling, manual annotation of tweets, and inconsistencies in the computation of the final sentiment score
Kundi et al. (2014)	SentiWordNet Slang repositories Opinion lexicon	Unsupervised Lexicon-based		limited coverage of slangs no support for emoticons lack of domain specific words support
Ranco et al. (2015)	Stock-trends related tweets correlation analysis and Granger causality	Supervised	5% significance level	Lack of historical data handling
Ribeiro et al. (2015)	SentiWordNet Domain-specific classifier	Hybrid	82% accuracy	Further experiments are required on larger datasets with tweet scraping in streaming mode.

3 | METHODS

The Twitter sentiment classification framework (T-SAF) implements a hybrid technique using an enhanced version of emoticon handling (Khan et al., 2014), slang classification (Kundi et al., 2014, general-purpose sentiment classification (Baccianella, Esuli, & Sebastiani, 2010), and enhanced sentiment classification using a domain-specific strategy (Asghar, Khan, Ahmad, Khan, & Kundi, 2015).

The proposed system (Figure 1) operates in four phases: (a) We classify the slang terms using the SC, (b) we perform emoticon classification using the EC, (c) we classify the sentiment words using the SWN-based classifier, and (d) we use the domain-specific classifier to classify the tweets as +ive, −ive or neutral, with emphasis on the domain-specific paradigm.

The proposed T-SAF is based on a hybrid classification scheme to classify tweets using four classifiers: (a) SC, (b) EC, (c) GPSC, and (d) IDSC.

The EC is used to classify emoticons on the basis of +ive and −ive emoticon sets. It detects the presence or absence of emoticons in a given tweet that are then classified as +ive, −ive, or neutral. The SC uses a list of +ive and −ive slang terms and slang definitions, stored in two database files. It detects the presence or absence of slang terms in a given tweet that are then classified as +ive, −ive, or neutral. We use GPSC for the sentiment classification of words and tweets, using the SWN lexicon to retrieve the sentiment score of each word. The IDSC module is used to perform sentiment classification of domain-specific terms that are either not present in SWN or have a sentiment score not accurately identified by SWN.

Let T be a list of tweets t defined as

T = {t1,t2,…,tn}.

Let W be a list of words in each tweet t defined as

W = {w1,w2,…,wn}.

Let S be a list of slang terms in each tweet t defined as

S = {s1,s2,…,sn}.

Let E be a list of emoticons in each tweet t defined as

E = {e1,e2,…,en}.

We propose the following four sentiment classifiers for the sentiment classification of a tweet.

3.1 | Slang classifier

Slang consists of the abbreviated words used by online users in their chats, blogs, reviews, and Tweets. Kundi et al. (2014), in their work, observed that detection and scoring of slang is an important task for determining the semantic nature of tweets and that the presence of a large number of slang terms has a high impact on the sentiment score of a tweet. They achieved 76% accuracy for a manually annotated slang-based tweet dataset.

There are different slang resources available online such as "noslang.com", "onlineslangdictionary.com", "netlingo.com", and "http://smsdictionary.co.uk/". We built a comprehensive slang dictionary using the aforementioned resources. We asked five manual annotators to label the slang terms as −0.5 (−ive), −1.0 (−ive), +0.5(+ive), 0 (neutral), and +1.0 (+ive).

After performing the manual annotation of the entire list, we received five votes for each slang, and whichever annotation had the maximum number of votes was deemed the winner. The proposed SC module is an enhancement of the work performed by (Kundi et al., 2014). They used 7,046 slang terms, whereas we have acquired 8,453 slang terms and their meanings; 50% are labelled as positive, 45% are marked as negative, and 5% are declared to be neutral. In 87.4% of the cases, our five annotators assigned similar scores to slang. A sample list of slang terms with their

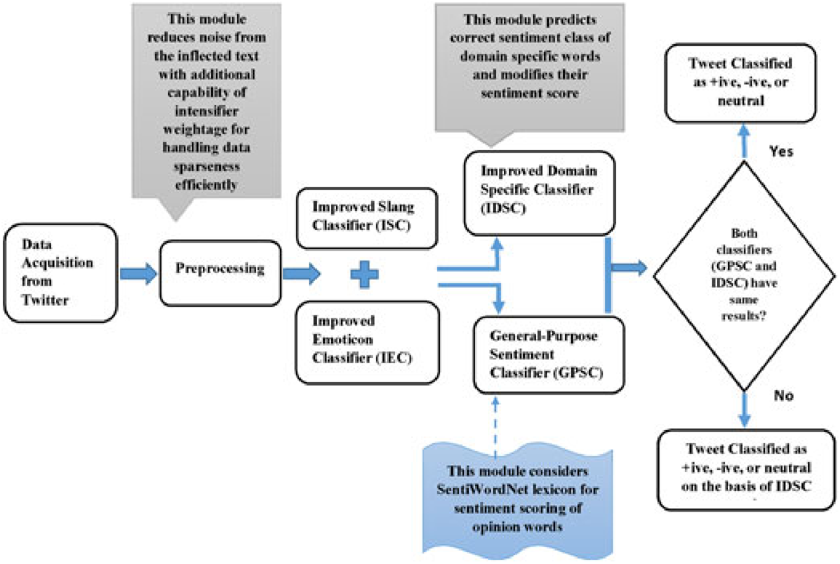
This module reduces noise from the inflected text with additional capability of intensifier weightage for handling data sparseness efficiently
This module predicts correct sentiment class of domain specific words and modifies their sentiment score
Tweet Classified as +ive, -ive, or neutral
Yes
Improved Domain Specific Classifier (IDSC)
Improved Slang Classifier (ISC)
Data Acquisition from Twitter
Preprocessing
Both classifiers (GPSC and IDSC) have same results?
Improved Emoticon Classifier (IEC)
General-Purpose Sentiment Classifier (GPSC)
No
Tweet Classified as +ive, -ive, or neutral on the basis of IDSC
This module considers SentiWordNet lexicon for sentiment scoring of opinion words

FIGURE 1 Proposed system

TABLE 2 A partial list of slangs with their sentiment class and scores

Slang	Meaning	Sentiment class	Sentiment score
Coolio	Cool	Positive	+0.080338
gr8	Great	Positive	+0.30814
Xoxo	Hugs and kisses	Positive	+0.137839
Air	Alright	Positive	+0.25
Happs	Happy	Positive	+0.5625
Smh	Shaking my head	Negative	−0.0671
Damn	Disbelief/condemn	Negative	−0.16477
Hehehe	Laughing	Negative	−0.1875
Notta	Not	Negative	−0.67262
Chale	Disagreement/disproval	Negative	−0.2822
Gonna	Want to go	Neutral	Obj(0.023256)
Haha	Laughing	Neutral	Obj(0.011628)
Rofl	Rofl rolling on floor laughing	Neutral	Obj(0.008854)
Wanna	Want to	Neutral	Obj(0.011628)

sentiment classifications is shown in Table 2. Slang is labelled as +ive if it is found in the positive list. Slang is marked as strongly positive if it is in the strongly positive category. Similarly, slang is labelled as neutral, negative, or strongly negative if it belongs to the respective category.

Let PSL be a list of +ive slangs, SPSL be a list of strongly +ive slangs, NSL be a list of negative slangs, and SNSL be a list of strongly negative slangs associated with each tweet, represented as

PSL = {list of positive slangs}

SPSL = {list of strongly positive slangs}

NSL = {list of negative slangs}

SNSL = {list of strongly negative slangs}.

The sentiment score of a slang "s_i" is computed as

$$Sent_{score}^{slang}(s_i) = \begin{cases} 1, & if\ e_i \in\ SPSL \\ 0.5, & if\ e_i \in\ PSL \\ -0.5, & if\ e_i \in\ NSL\ . \\ -1, & if\ e_i \in\ SNSL \\ 0 & else \end{cases} \tag{1}$$

The sentiment score of a slang s_i is a value ranging between 1 and −1, where 1 means the sentiment is strongly positive, 0.5 means it is positive, 0 means it is neutral, −0.5 means it is negative, and −1 means it is strongly negative.

3.2 | Emoticon classifier

Emoticons are the textual facial expressions employed by users in instant messages and chats on social media. Emoticons are composed of punctuation marks, letters, and numbers. Khan et al. (2014), in their work on Twitter data, reported that the evaluation of emoticons as opinion terms in Twitter data can produce improved results in sentiment analysis applications. They achieved 85% accuracy for different datasets containing emoticons.

The proposed EC is an extension of the technique proposed by Khan et al. (2014) that constructs a comprehensive emoticon dictionary in which emoticons are sorted into positive and negative classes with numeric scores assigned to them. The previous authors used 145 trained emoticons, whereas we have included 721 emoticons, 389 of which are labelled positive, 278 are marked as negative, and 84 are declared neutral. Moreover, we further classify the emoticons as positive, strongly positive, neutral, negative, and strongly negative.

There exists a number of emoticon lists.[1,2,3,4,5] We propose to integrate these five existing lists into a single dictionary, discarding the duplicate entries. We obtained a list of 721 emoticons. Five human annotators manually assigned polarity class and score to the emoticons in our dictionary. The annotators were told to assign scores of −0.5 (−ive), −1.0 (−ive), +0.5 (+ive), 0 (neutral), and +1.0 (+ive).

[1] http://www.msgweb.nl/en/MSN_Images/Emoticon_list/.

[2] http://netforbeginners.about.com/cs/netiquette101/a/bl_emoticons101.htm.

[3] http://www.sharpened.net/emoticons/.

[4] http://www.windweaver.com/emoticon.htm.

[5] https://en.wikipedia.org/wiki/List_of_emoticons.

The score nearest to the average of the annotators' scores is computed for each emoticon. In 85.2% of the cases, our five annotators assign similar scores to the emoticons. A sample list of emoticons is shown in Table 3. The emoticon is labelled as +ive if it is found in the positive list. The emoticon is marked as strongly positive if it is in the strongly positive category. Similarly, the emoticons labelled as neutral, negative, or strongly negative if it belongs to the respective category.

Let PEL be a list of +ive emoticons, SPEL list of strongly +ive emoticons, NEL a list of negative emotions, and SNEL a list of strongly negative emoticons associated with each tweet, represented as

PEL = {list of positive emoticons}

SPEL = {list of strongly positive emoticons}

NEL = {list of negative emoticons}

SNEL = {list of strongly negative emoticons}.

The sentiment score of an emoticon "e_i" is computed as

$$Sent_{score}^{emoticon}(e_i) = \begin{cases} 1, & if\ e_i \in\ SPEL \\ 0.5, & if\ e_i \in\ PEL \\ -0.5, & if\ e_i \in\ NEL\ . \\ -1, & if\ e_i \in\ SNEL \\ 0 & else \end{cases} \tag{2}$$

The sentiment score of an emoticon e_i is a value between 1 and −1, where 1 stands for strongly positive, 0.5 means positive, 0 is neutral, −0.5 means negative, and −1 stands for strongly negative.

3.3 | General-purpose sentiment classifier

The GPSC module is used to assign polarity scores to sentiment words. There are different sentiment lexicons such as SWN (Baccianella et al., 2010), SentiStrength (http://sentistrength.wlv.ac.uk/), and SentiWS (http://asv.informatik.uni-leipzig.de/download/sentiws.html). We choose to use SWN because of its wide range of words with sentiment scores. SWN is a general-purpose lexicon with more than 60,000 synsets obtained dynamically from WordNet. Each of the words is tagged with three sentiment scores: +ive, −ive, and objective, represented by $sent^+$, $sent^-$, and $sent^o$, respectively. The scores range in the interval 0.0 to 1.0, and the overall sum is equal to 1.0 for each word.

TABLE 3 Partial list of emoticons with their sentiment class and scores

Emoticon	Meaning	Sentiment class	Sentiment score
:*	Kiss	Positive	+0.5
;),=) :-)	Happy, smile	Positive	+0.5
;3	Happy face or smiley	Positive	+0.5
<3	Heart, I love you	Positive	+0.5
XD	Laughing	Strongly positive	+1
:))	Big smile	Strongly positive	+1
BD	Big grin with glasses	Strongly positive	+1
:D	Big grin	Strongly positive	+1
:)))	Really happy	Strongly positive	+1
8D	Big eyes & big smile	Strongly positive	+1
\m/	Hi 5	Strongly positive	+1
:\	Undecided	Neutral	0
:-)8:	Woman	Neutral	0
:|	Straight face	Neutral	0
</3	Broken heart	Negative	−0.5
B(	Sad with glasses	Negative	−0.5
:(	Sad	Negative	−0.5
:e,;e	Disappointed	Negative	−0.5
X-(	Angry	Strongly negative	−1
:-<	Very sad	Strongly negative	−1
D:	Disgust, sadness, horror	Strongly negative	−1
:'(	Crying	Strongly negative	−1

To disambiguate between the multiple meanings of a sentiment word, we evaluate its correct sense by calculating three average values, Sent_score+, Sent_score⁻, and Sent_score, for all the POS of a term "*ti*" as follows:

$$Sent\_score^{+}(t_i,p)=\frac{1}{numSyn}\sum_{i=1}^{n}sent^{+}\_score(t_i)\ ,\tag{3}$$

$$Sent\_score^{-}(t_i,p)=\frac{1}{numSyn}\sum_{i=1}^{n}sent^{-}\_score(t_i)\ ,\tag{4}$$

$$Sent_{score^{o}(t_i,p)}=\frac{1}{numSyn}\sum_{i=1}^{n}sent^{o}_{score(t_i)},\tag{5}$$

where *Sent_score*$^{+}$, *Sent_score*$^{-}$, and *Sent_score*^{o} represent the average sentiment score (+ive, −ive, and objective) of synset *i* for term *ti*, *p* denotes POS (adjective, noun, verb, adverb, etc.), and numSyn represents the total number of synsets of a term for a corresponding POS.

There may be multiple meanings of a word in a specific grammatical category in SWN. For example, the word "bad" has 14 meanings in the Adjective category, two meanings in the Adverb category, and one meaning in the Noun category. To disambiguate multiple meanings in a particular category, we compute the average of the positive, negative, and objective scores as follows: *Sent_score*$^{+}$ = ("bad," "adverb") = 0.125 + 0.125 = 0.25/2 = 0.125, *Sent_score*$^{-}$ = ("bad," "adverb") = 0.25 + 0.25 + 0 = 0.5/2 = 0.25, and *Sent_score*^{o} = ("bad," "adverb") = 0.625 + 0.625 = 1.25/2 = 0.625. The average polarity triplet for the word "bad" in the adverb category is <0.125, 0.25, 0.625>.

After computing the mean (average) for different synsets of a word in a particular POS category, we obtain three scores: +ive, −ive, and objective. The final score of the sentiment term is calculated by choosing its dominant polarity as follows:

$$Sent\_score^{swn}(t_i,p\ )=\begin{cases} sent\_score^{+}(t_i,p), \quad if max(sent\_score^{+}(t_i,p),sent\_score^{-}(t_i,p),sent\_score^{o}(t_i,p))=sent\_score^{+}(t_i,p\ ) \\ sent\_score^{-}(t_i,p) if\ \max(sent_{score}{}^{+}(t_i,p),sent_{score}{}^{-}(t_i,p),sent_{score}{}^{o}(t_i,p))=sent_{score}{}^{-}(t_i,p\ ) \\ sent\_score^{o}(t_i,p) else \end{cases}.\tag{6}$$

ti is +ive if the positive score is greater than both the negative and neutral scores. We obtain negative polarity by applying the corresponding rule. The sentiment score is objective if the positive and negative sentiment scores are equal or the objective polarity is greater than both the positive and negative ones. For example, the average sentiment window <*sent_score*$^{+}$(t_i, p) , *sent_score*$^{-}$(t_i, p) , *sent_score*^{o}(t_i, p)> for the term "Awesome" is <0.875, 0.125, 0>; therefore, Sent_score^{swn}("Awesome") = 0.875, which is +ive.

3.4 | Improved domain-specific classifier

Domain-specific words all have the same polarity class in SWN; however, their presence in labelled reviews sometimes indicates a strong association with another sentiment class. Therefore, identification of such words is required for accurate sentiment classification. For example, in a health-related domain, if a term's SWN-based aggregated sentiment score is positive but its occurrence in negative reviews is sufficiently high compared to its positive occurrences, we change its sentiment score.

The IDSC module is used to assign accurate sentiment classes and scores to domain-specific words in different domains. In this technique, we first identify domain-specific words using a frequency-based probability measure. In the next step, the sentiment scores of domain-specific words are recalculated using improved polarity update methods. The proposed technique is inspired by the method presented in Asghar et al. (2015) for identifying and scoring opinion words in multiple domains.

The IDSC classifier operates in two stages, (a) predicting a term's sentiment class and (b) modifying a word's sentiment score.

3.4.1 | Predicting a term's sentiment polarity class

To determine which terms occur more frequently in one class compared to another, we compute the frequency-based probability (Smeureanu & Bucur, 2012) of each term in labelled reviews. A supervised learning approach was used in Smeureanu & Bucur (2012) for the sentiment classification of movie reviews as +ive or −ive. We enhance this approach to predict the sentiment class of domain-specific words in labelled tweets as follows:

$$Polarity(w)=\begin{cases} positive, if P(w,c_{+}){>}P(w,c_{-}) \\ negative, otherwise \end{cases},\tag{7}$$

where w is the word being considered, and *P*(*w*, c_{+}) and *P*(*w*, c_{-}) are the probabilities of word w occurring in +ive and −ive tweets of the training set, respectively, computed as follows:

$$P(w,c_{+})=\frac{count(w{\in}R_{+})}{|R_{+}|},\tag{8}$$

$$P(w,c_{-})=\frac{count(w{\in}R_{-})}{|R_{-}|}.\tag{9}$$

$\mathbf{R_{+}}$ and $\mathbf{R_{-}}$ are training sets of positive and negative reviews, respectively.

For example, the word "clot" has the objective sentiment class in SWN, but $P(w, c_-) > P(w, c_+)$ for "clot," indicating that it has a more −ive association. Similarly, the term "relax" has a neutral sentiment in SWN, whereas its $P(w, c_+)$ value is greater than $P(w, c_-)$, indicating that it is more associated with the +ive class. A partial list of +ive and −ive terms is presented in Table 4.

3.4.2 | Updating a term's sentiment score

When the SWN-based average score (Equations 3 to 6) and the probability-based score of a term (Equation 7) are not equal, we modify the sentiment score. Moreover, if a term is not found in SWN, then sentiment classification and scoring of the term become difficult. Therefore, the sentiment scores of such terms need to be updated accordingly. The aforementioned problems can be solved by proposing a revised scheme of sentiment scoring for domain-specific words.

The recent studies (Aldayel & Azmi, 2016; Asghar, Ahmad, Qasim, Zahra, & Kundi, 2016; Duwairi & El-Orfali, 2014; Ikeda, Takamura, Ratinov, & Okumura, 2008) have demonstrated that their sentiment update methods produce promising results with respect to accurate scoring of domain-specific words in the sentiment analysis of user reviews. They achieved significant improvement in accuracy in both the hotel and movie domains. The proposed polarity update method improves on the approach presented by Aldayel and Azmi (2016) by using switch and switch-neutral methods for changing the polarity of a term used in tweets. The authors of Aldayel and Azmi (2016) used a supervised learning technique for updating the polarity of words in reviews, whereas we compute the new sentiment score of a term using a revised scheme of (a) polarity switching, (b) neutral switching, and (c) term weighting. This enhancement contributes significantly to the accurate scoring of domain-specific words, as reported in the results and evaluation section.

Revised polarity switch

In the revised polarity switch technique, we change the polarity of the term to its opposite based on its occurrences in +ive and −ive tweets (using Equation 7). For example, if a term's SWN-based polarity (Equation 6) is positive but it has more occurrences in negative tweets (Equation 9), then we switch it to negative polarity and vice versa:

Revised neutral switch

If a term's SWN-based polarity is neutral (Equation 6) but it occurs most frequently in positive tweets (Equation 8), then the SWN-based neutral score is switched to its corresponding positive score (and similarly for terms in negative tweets):

$$Sent\_score^{DS} = \begin{cases} Pol_{score}{}^{+}, & (Pol_{score}{}^{+} > Pol_{score}{}^{-}) \wedge (Pol_{score}{}^{+} > Pol_{score}{}^{o}) \wedge ((P(w, c_+) > P(w, c_-)) \\ Pol_{score}{}^{+*}\ (-1), & (Pol_{score}{}^{+} > Pol_{score}{}^{-}) \wedge (Pol_{score}{}^{+} > Pol_{score}{}^{o}) \wedge ((P(w, c_-) > P(w, c_+)) \\ Pol_{score}{}^{-*}\ (-1), & (Pol_{score}{}^{-} > Pol_{score}{}^{+}) \wedge (Pol_{score}{}^{-} > Pol_{score}{}^{o}) \wedge ((P(w, c_+) > P(w, c_-)) \\ Pol_{score}{}^{-}, & (Pol_{score}{}^{-} > Pol_{score}{}^{+}) \wedge (Pol_{score}{}^{-} > Pol_{score}{}^{o}) \wedge ((P(w, c_-) > P(w, c_+)) \end{cases} . \tag{10}$$

For example, the SWN-based score of the term "scream" is obj(0.75), but it appears most often in negative health-related tweets (Equation 9); we update its polarity score (Equation 10) to −0.75, which represents a negative sentiment score.

3.4.3 | Revised term weighting scheme

If a word is not available in SWN, then its sentiment classification and scoring become difficult. To compute a sentiment score of a term not found in SWN, we integrate the term frequency (*tf*), inverse document frequency (*idf*), and frequency-based probability (Equation 11). The proposed metric is an enhancement of the existing delta scoring technique (Paltoglou & Thelwall, 2010). The previous technique is based on weighted scores using a general-purpose technique. The performance of the existing delta scoring technique on different datasets was evaluated, and it obtained

TABLE 4 Partial list of +ive and −ive terms in heath and automobile domains

Term	SentiWordNet polarity	Predicated polarity using Equation 7
Moral distress	Not found	Negative
Relax	Neutral	Positive
Clot	Neutral	Negative
Brakes	Neutral	Positive
Sleepwalking	Not found	Negative
Ride	Neutral	Positive
Cruise control	Not found	Negative
Cold starting	Not found	Negative
Mehran	Not found	Positive
Bolan	Not found	Positive

better accuracy than the general-purpose technique. However, we propose to integrate the frequency-based probability measure (Equation 7) with the existing *tf x idf* scheme, providing a significant increase in classification accuracy for domain-specific words compared with baseline methods. It is computed as

$$Sent\_score^{DS} = \begin{cases} tfxidfxP(w,c_+),\ (w \notin SWN) \wedge (P(w,c_+) > P(w,c_-)) \\ tfxidfxP(w,c_-),\ \ (w \notin SWN) \wedge\ ((P(w,c_-) > P(w,c_+)) \\ \\ \end{cases} . \tag{11}$$

For example, the term “sleepwalking” is not available in SWN. Using revised term weighting (Equation 11), the score of “sleepwalking” becomes −3.61. Table 5 presents a sample list of words and applied formula for *tf*, *idf*, *tf x idf*, $P(w, c_+)$, $P(w, c_-)$, and *tf x idf x* $P(w, c_-)$, which is helpful for understanding how the *tf x idf* metric is distinguished from the frequency-based probability measure. Furthermore, it is obvious that the proposed frequency-based probability measure successfully assigns revised sentiment scores to the terms that are not found in SWN.

A partial list of words affected by revised polarity or weighting is presented in Table 6.

The major limitation of some statistical measures (Goker & Davies, 2009) is that they do not provide reliable results for rare terms. However, the proposed IDSC efficiently considers rare terms. For example, the term, “worst,” appears only twice in the *negative* training corpus. However, it has a frequency-based probability of 0.0013 and a revised sentiment score of −0.42 (using Equation 11), which is very reliable, because the score for the term, “worst,” is not available in SWN.

3.5 | Classifying the tweet

For each tweet in a review, we compute a sentiment score by adding the scores of all of the sentiment words, slang terms, and emoticons. First, we classify a tweet using a GPSC-based technique:

$$tweet_{class} = \begin{cases} postive,\ \sum_{i=1}^{n} \left(Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i + Sent\_score^{swn}{}_i\right) > 0 \\ negative,\ \sum_{i=1}^{n} \left(Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i + Sent\_score^{swn}{}_i\right) < 0 \\ neutral,\ \sum_{i=1}^{n} \left(Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i + Sent\_score^{swn}{}_i\right) = 0 \vee \\ \sum_{i=1}^{n} \left(Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i + Sent\_score^{swn}{}_i\right) isobjective \end{cases} . \tag{12}$$

An example tweet in the health domain is written: <tweet> = “I am having the worst anxiety and depression by using this medicine :(.” To perform the sentiment classification of the given tweet, we first use the GPSC (Section 3.3) classifier, computing sentiment scores of slang terms, emoticons, and opinion words in the input tweet as follows:

TABLE 5 List of words and applied formula for different metrics used in proposed formula

Word (w)	*tf*	*idf*	*tf x idf*	$P(w, c_+)$ Equation 8	$P(w, c_-)$ Equation 9	Polarity (w) using Equation 7	$tfxidfxP(w, c_-)$
Cold starting	112	1.6	179.2	0.007	0.0156	−ive	−2.79
Sleep walking	98	2.79	273.42	0.002	0.0132	−ive	−3.61
Cruise control	74	1.32	97.68	0.001	0.0161	−ive	−1.57

TABLE 6 Words and their modified semantic orientation

Term	SWN score	Modified sentiment score	Example tweet	Dataset
Moral distress	Not found	−2.5 (negative; using Equation 11)	It didn't helped for moral distress and sleepless night.	Health
Relax	0.625 (neutral)	+0.625 (positive; using Equation 10)	Beta blocker relaxes me during hypertension.	Health
Clot	1 (neutral)	−1 (negative; using Equation 10)	The doctor diagnosed a blood clot in the brain.	Health
Sleep-walking	Not found	−3.61 (negative; using Equation 11)	Sleep walking is one of the terrible down side.	Health
Cruise control	Not found	−1.57 (negative; using Equation 11)	Jeep Toyota 2012 has enormous power, however the cruise control is unsatisfactory.	Automobile
Cold starting	Not found	−2.79 (negative; using Equation 13	The cold starting issues arises at the start of winter season, it really troubles me.	Automobile
Brakes	0.782 (neutral)	−0.782 (negative; using Equation 10)	Suzuki Mehran lacks the ABS system and therefore not reliable in brakes.	Automobile

Slang scoring: Using Equation 1, the sentiment score of the slang is evaluated as follows: $Sent\_score^{slang}(s_i) = 0$; the 0 shows that there is no slang, so slang scoring is not applicable.

Emoticon scoring: There is one emoticon in the example tweet; by Equation 2, the polarity score of emoticon = $Sent\_score^{emoticon}(ei)$ = $Sent\_score^{emoticon}(" : (") = -0.5$ because the sad face emoticon is negative, with score = −0.5.

Sentiment word scoring: Using Equation 6, the sentiment scores of the three opinion words, “worst,” “anxiety,” and “depression,” are computed as 0.75 (negative), 0.5 (neutral), and 0.85 (neutral), respectively.

Therefore, $Sent\_score^{swn}("worst") = -0.75$, $Sent\_score^{swn}("anxiety") = \mathrm{obj}(0.5)$, and $Sent\_score^{swn}("depression") = \mathrm{obj}(0.85)$, where obj represents a neutral score.

Using Equation 12, we compute the sentiment score of the given input tweet as follows:

$$= \sum_{i=1}^{n} \left(Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i + Sent\_score^{swn}{}_i \right) = 0 + (-0.5) + (-0.75) + \mathrm{obj}(0.5) + \mathrm{obj}(0.85) = \boldsymbol{obj}(0.1).$$

The overall sentiment of the tweet is neutral, with score = obj(0.1), where obj denotes a neutral score.

The major problem with the GPSC classifier is that it may inaccurately score domain-specific words, which may lead to the incorrect classification of tweets in some domains. For example, “depression” and “anxiety” are domain-specific words, and their scores computed by the GPSC classifier are not correct, resulting in a neutral (0.1) overall tweet score for the previous input tweet, which is incorrect. Therefore, we further classify a tweet using the IDSC technique.

IDSC-based classification: To classify domain-specific words more accurately, we further classify a tweet using the IDSC

$$tweet_{class-DS} = \begin{cases} positive, & \sum_{i=1}^{n} \left(Sent\_score^{DS}{}_i + Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i \right) > 0 \\ negative, & \sum_{i=1}^{n} \left(Sent\_score^{DS}{}_i + Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i \right) < 0 \\ neutral, & \sum_{i=1}^{n} \left(Sent\_score^{DS}{}_i + Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i \right) = 0 \vee \\ & \sum_{i=1}^{n} \left(Sent\_score^{DS}{}_i + Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i \right) is objective \end{cases} . \tag{13}$$

The tweet-level score calculated using the GPSC is obj(0.1). We update the sentiment score of the domain-specific terms “depression” and “anxiety” using Equation 13 and classify the tweet as

$$tweet_{class-DS} = \sum_{i=1}^{n} \left(Sent\_score^{DS}{}_i + Sent\_score^{slang}{}_i + Sent\_score^{emoticon}{}_i \right) = ((-0.75) + (-0.5) + (-0.85) + 0 + (-0.5)) = -2.6 < 0,$$

$$tweet_{class-DS} = negative.$$

When we compare the GPSC-based (obj(0.01953)) and IDSC-based (−2.418) tweet-level scores, the IDSC identification and correct scoring of domain-specific terms have produced a more accurate classification and scoring of the entire tweet and helped reduce the classification anomalies. As explained earlier, the IDSC module is applied only when the GPSC-based technique yields inaccurate classification results. The underlying logic is presented in Algorithm 1.

3.6 | The proposed algorithm

The proposed algorithm (Algorithm 1), based on a hybrid classification technique, illustrates the steps required for tweet classification. First, we classify each tweet using four classifiers in a step-wise fashion. Finally, the tweets are classified as +ive, −ive, or neutral.

Algorithm 1. *Enhanced Twitter sentiment classification.*

```
Input: Tweets
Output: Sentiment Score
Begin

// Scan the entire corpus
1. While (there is tweet in corpus) Do
2.    Call Pre-processing(tweet)
// Subjective and objective tweet Identification
  Search each term of tweet in opinion lexicon, emoticon dictionary and slang dictionary
 3.    if (a tweet contains opinion word/emoticon/slang )
 4.         Subjective Tweet
 5.         Call sentiment_scoring(subjective tweet)
 6.         Go to step#1 to scan next tweet
 7.   else
 8.       Objective Tweet
 9.       Go to step#1 to scan next tweet
 10.    end if
// Sentiment Scoring
11.    For each word in tweet
12.      If word found in Slang dictionary
13.          Perform classification using Slang Classifier (Eq. 1)
14.      If word found in Emoticon dictionary
15.          Perform classification using Emoticon Classifier (Eq. 2)
16.      Perform classification using General-Purpose Sentiment  Classifier (Eq. 6)
17.         Perform classification using Domain-Specific Classifier (Eq. 7, Eq. 10, Eq. 11)
18. Next word
19.      Classify tweet using General-purpose classifier ( Eq. 12)
20.      Classify tweet using IDSC classifier (Eq. 13)
21.   If General-Purpose Sentiment  Classifier produces Correct sentiment classification
             Declare the tweet as +ive, -ive or neutral accordingly
22.    else
23.             Declare the tweet as +ive, -ive or neutral on the basis of IDSC classification
24.        Return $tweet_{class}$
25. end-while
End Function
```

4 | EXPERIMENTS AND RESULTS

To implement the algorithms presented in the previous sections, we use Python and Natural Language Toolkit (NLTK; Bird, Klein, & Loper, 2009). We use datasets from three different domains to conduct the experiments.

4.1 | Data acquisition

Data acquisition involves dataset compilation from Twitter. Tweets may be about products, news, politics, sports, or any other issue. We acquire tweets about different products in three domains, (1) automobile, (b) laptop, and (3) health. The Python Tweepy (Roesslein, 2015) package is used for this purpose. Acquired tweets are stored in an SQL Server 2014 database and used as input to the preprocessing phase for further manipulation. Non-English tweets and retweets are ignored. We use Alchemy API (http://www.alchemyapi.com/api) to classify tweets into +ive, −ive, and neutral sentiment categories. The classified tweets are stored in the database to compile the complete dataset. We divide and store the dataset into two separate database files for training and testing. Table 7 presents details of the acquired datasets.

4.2 | Preprocessing

In the preprocessing phase (Algorithm 2), inflected text is filtered by applying different preprocessing techniques. The preprocessing module works as follows:

TABLE 7 Sample datasets

Datasets	Total no. of tweets	Retweet	English tweets	Query string	Tweet %	Dataset
Dataset#1	3,500	953	1,726	Toyota Honda Suzuki	46% 24% 30%	Automobile
Dataset#2	2,737	838	1,943	Sony Dell HP	42% 32% 26%	Laptop
Dataset#3	4,127	598	3,467	Diabetes Hypertension Psychiatry	44% 32% 24%	Health

4.2.1 | Tokenization

We used the Python-based NLTK tokenizer (Asghar et al., 2015) to tokenize the tweets into individual words, emoticons, and slang terms. Examples of tokenized words and emoticons include “best,” “happy,” “like,” “:)”, “:(”, and “Xoxo.”

4.2.2 | Slang identification

Each word of a tweet is searched in online slang dictionaries (e.g., noslang.com, “onlineslangdictionary.com”, “netlingo.com”, and “http://smsdictionary.co.uk/”). If the word is identified as an abbreviation or slang, then it is marked and removed from the tweet. The removed abbreviation or slang term is stored with its tweet number in a text file for further processing by the SC. For example, in the tweet “wish me luck guys. n have agr8 day. .love”, the term“gr8” is identified as slang, removed from the tweet, and stored in a text file.

4.2.3 | Emoticon identification

Each of the terms in a tweet is searched in emoticon dictionaries such as netlingo.com, techdictionary.com, and lingo2word.com. If a feature/term is identified as an emoticon in any of the dictionaries, then it is removed from the tweet. The removed emoticon is stored with its tweet# in a text file for further processing. For example, in the tweet “love Mobile :) Enjoying new piece tech.”, “:)” is identified as an emoticon. It is removed and stored in a text file. Other examples of emoticons are “:(” and “: D”.

4.2.4 | Filtering

Stop words, hashtags, RT (retweet symbols), punctuation marks, URLs, query terms, and special characters other than emoticons and slang are identified and removed. Multiple spaces are replaced with a single space.

4.2.5 | Lemmatization

We used the NLTK-based WordNet Lemmatizer (http://www.nltk.org/_modules/nltk/stem/wordnet.html) to substitute words with their root forms. For example, lemmatization converts the words “car” and “caring” to the lemmas “car” and “care” (Asghar et al., 2016).

4.2.6 | Spell correction

To perform spell checking and correct misspelled words, we used the Python-based Aspell library (https://pypi.python.org/pypi/aspell-python-py2/1.13). This improves the accuracy of sentiment classification. Because the Aspell library suggests several words, we choose the first word. In preprocessing, the slang identification module (Section 4.2.2) is applied before applying the spell correction step (Section 4.2.6), which preserves the slang terms. Therefore, the Aspell checker does not remove the slang terms; instead, these are retained via the slang identification module.

4.2.7 | POS tagging

To assign SWN scores to each synset of a word in a given tweet, we perform POS tagging using NLTK (http://www.nltk.org/book/ch05.html) functionality implemented in Python. The NLTK assists in applying POS tags such as “noun,” “verb,” and “adjective” to the tweet terms.

4.2.8 | Negations

We use a list of negation terms to check for the existence of negations in a tweet. If a word is found on the negation list, then the polarity of the neighbouring opinion word is flipped by simply multiplying the score of the opinion word by −1. For example, the polarity score of “effective” = 0.65, and therefore, the polarity score of “not effective” = 0.65 * −1 = −0.65.

Algorithm 2. *Preprocessing.*

```
Input: Inflected Tweets
Output: Pre-processed Tweets

Begin
// Scan the entire corpus
1. While (there is tweet in corpus) Do

2. tokens = tokenize(text)
3. For each tok in tokens
4. Remove Stop Words
5. Search token in opinion lexicon/dictionary
6. If found goto step 10
7. Search token in Slang/abbreviations dictionary
8. If found goto step 10
```

```
9. Apply Filtering
10. Correct Spelling
11. Apply Lemmatization
12. Perform POS tagging
13. Manage Negations
14. Repeat steps 3, 5
15.If found goto step 10 Else Remove tok
16. Next tok
```

4.3 | Validation

The first step of a supervised learning algorithm splits the dataset into two distinct parts, the training set and the test set. These two independent subsets of the dataset are used to verify method's efficacy. Therefore, during each iteration, more than one fold of the experiments are required with independent datasets. We use leave-one-out-cross validation (Asghar et al., 2015), a special variation of N-fold cross validation, in our experiments. The process involves dividing the dataset into K-folds (5–10 folds), where one fold is used for testing and the remaining K-1 are used for training the system. For each experiment, we use 10-fold cross validation, where nine of the 10 folds (90%) are used for training and one (10%) for testing. The final result is obtained by finding the average of all 10 experiments.

4.4 | Results and evaluation

In this section, we evaluate the effectiveness of the proposed method using different evaluation metrics, such as precision, recall, F-score, and accuracy, as follows:

$$Precision\ (p) = \frac{tp}{tp + fp},$$

$$Recall\ (r) = \frac{TP}{tp + fn},$$

$$F{-}measure = \frac{2(p)(r)}{p + r},$$

$$\text{Accuracy} = \frac{tp + tn}{\text{tp} + \text{fp} + \text{tn} + \text{fn}},$$

where *tp* is the number of true +ive classifications, *fp* is the number false +ive classifications, *tn* is the number of true −ive classifications, and *fn* is the number of false −ive classifications, as shown in Table 8.

4.4.1 | Experiment #1

The first experiment investigates the impact of slang on the sentiment analysis of tweets. For accurate sentiment classification, it is necessary to compute the sentiment scores of slang terms before performing the sentiment classification of a tweet. Table 9 shows the effect of slang on sentiment classification of tweets.

TABLE 8 Confusion matrix

Data class	Actual	Predicted
Positive	TP	FN
Negative	FP	TN

TABLE 9 Comparative results of slang classifier module

Method	Positive				Negative				Neutral				Overall
	P	R	F	A	P	R	F	A	P	R	F	A	A
Masud et al. (2014)	0.91	0.79	0.85	0.79	0.70	0.91	0.70	0.91	0.66	0.91	0.65	0.63	0.81
T-SAF (Proposed)	0.97	0.88	0.92	0.88	0.75	0.96	0.84	0.96	0.83	0.97	0.74	0.67	0.90

Note. T-SAF = Twitter sentiment analysis framework.

4.4.2 | Experiment #2

In another experiment, we analyse the effect of emoticons in tweet classification by classifying the tweets using the EC. Our results (Figure 2) demonstrate that inclusion of emoticon features in the proposed framework improves classification accuracy from 79% to 85%.

4.4.3 | Experiment #3

A third experiment measures the effectiveness of the IDSC module for the sentiment classification of domain-specific words. Figure 3 shows that sentiment analysis accuracy is enhanced significantly by the use of the IDSC classifier.

4.4.4 | Experiment #4

A fourth experiment uses the proposed hybrid classification algorithm on three datasets to classify each tweet into +ive, −ive, and neutral classes. The efficacy of the proposed system is compared with current state-of-the-art methods. The comparative results show that our approach outperforms the related approaches. Tables 10 and 11 present comparative performances for binary and multi-class sentiment classification.

4.5 | Qualitative evaluation

In this section, we present qualitative evaluation of the results returned by T-SAF. Six result categories are evaluated.

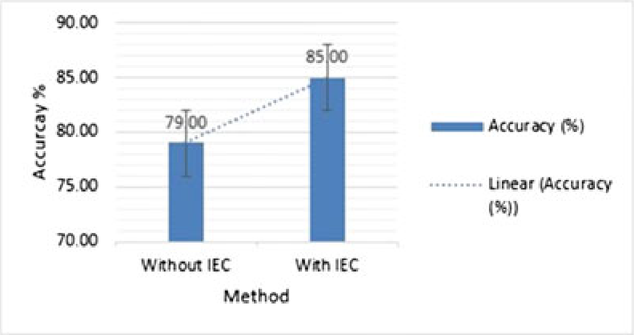

FIGURE 2 Accuracy results of EC classifier. IEC = improved emoticon classifier

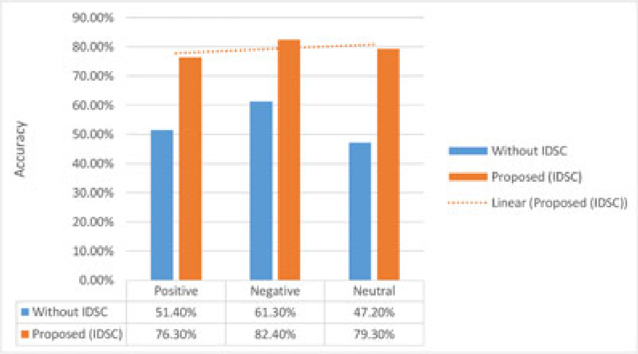

FIGURE 3 Accuracy results of IDSC module. IDSC = improved domain-specific classifier

TABLE 10 Experimental results for binary classification (P: Precision, R: Recall, F: F-measure)

	Positive			Negative		
Method	P	R	F	P	R	F
Asghar et al. (2017)	0.79	0.80	0.79	0.81	0.76	0.79
Khan et al. (2014)	0.81	0.79	0.80	0.79	0.82	0.80
Masud et al. (2014)	0.86	0.78	0.81	0.80	0.82	0.80
T-SAF (Proposed)	0.93	0.84	0.88	0.86	0.89	0.87

Note. T-SAF = Twitter sentiment analysis framework.

TABLE 11 Experimental results for multi-class classification (P: Precision, R: Recall, F: F-measure)

		Positive			W-Prep			Neutral		
Method	Tweets	P	R	F	P	R	F	P	R	F
Asghar et al. (2017)	3,500	0.56	0.67	0.61	0.62	0.45	0.52	0.75	0.73	0.73
Khan et al. (2014)	2,737	0.74	0.51	0.60	0.73	0.63	0.67	0.78	0.71	0.74
Masud et al. (2014)	3,167	0.82	0.78	0.79	0.71	0.85	0.77	0.79	0.63	0.70
T-SAF (Proposed)	4,127	0.89	0.79	0.83	0.84	0.74	0.78	0.84	0.74	0.78

Note. T-SAF = Twitter sentiment analysis framework.

4.5.1 | Emoticons

The accurate classification of emoticons is of paramount importance for automated sentiment analysis systems. The results presented in Figure 2 clearly indicate that including emoticons when classifying tweets provides a key enhancement to the state-of-the-art lexicon-based sentiment analysis method (Masud et al., 2014). The results obtained further suggest that whenever emoticons are present, these pictorial symbols play a pivotal role in expressing user's sentiment. However, there are certain limitations to our method that need to be overcome. Our system is unable to correctly classify tweets in which a user expresses conflicting sentiments. For example, the tweet "The weather is rainy :(, I want clear sunshine:)" is classified as neutral when passed through our system, as the −ive and +ive emoticons cancel the effect of each other in this particular tweet.

4.5.2 | Slang

The proper detection and classification of slang in text plays a significant role in Twitter sentiment analysis. Table 9 shows that slang detection and classification has a strong impact on sentiment classification. The comparative results presented in Table 9 show that the SC module serves as a valuable addition to the state-of-the-art method (Masud et al., 2014). However, the SC module has several limitations. The Urban Dictionary includes millions of terms used in multiple ways; therefore, it becomes very cumbersome to isolate redundant terms and their definitions. For the majority of such cases, we select the first meaning (definition) of a slang term. There is a need to devise a mechanism to disambiguate multiple meanings in an intelligent way.

4.5.3 | Domain-specific words

The presence of domain-specific words in text creates problems in the accurate classification of sentiments. The results presented in Figure 3 demonstrate that considering domain-specific meaning in Twitter sentiment analysis makes a valuable contribution to the state-of-the-art microblog sentiment analysis framework (Khan et al., 2014). Recalling the results in Table 6, the words "Mehran" and "Bolan" are two products of an automaker whose models have considerable popularity in our dataset. Most of the tweets in the health dataset containing the term "clot" are negative (24 negative and 3 positive). The term "relax" is classified as objective in the SWN, whereas it appears mostly in +ive tweets in the health dataset (17 positive and 4 negative), so the word "relax" is considered to be in the +ive class. Therefore, our framework detects more accurately the sentiment associations of words in the experimental datasets. There are certain outliers that need to be addressed. The word "Suzuki" gives neutral sentiment without context. However, "Suzuki" is a famous automaker with significant approval from its consumers. There are 43 tweets about Suzuki models in our dataset, and 94% are +ive. Therefore, the term "Suzuki" has a close association with the +ive class. Our system cannot accurately classify tweets containing domain-specific words that are not addressed by our three datasets. Domain-specific words outside the existing datasets can be exploited in sentiment analysis systems.

4.5.4 | Lack of named entity recognition

Named entity recognition enables sentiment analysis systems to identify and classify text into predefined types, for example, organization names, cities, countries, films, persons, and locations. Our proposed system cannot detect and classify such named entities. For example, the tweet "I can't imagine that, but ok, Pakistan Cricket Team won the quarter final" is labelled as −ive. The words "Pakistan," "Cricket," and "quarter final" are the named entities. They are not able to be classified, and the system becomes trapped.

4.5.5 | Sarcastic tweets

Sarcastic tweets intentionally misuse words and phrases to convey humour, and our system often produces inaccurate classification results in such cases. For example, the tweet "the latter is the best time to do anything" is tagged as +ive by T-SAF. A sophisticated algorithm is needed for classifying this type of humour.

4.5.6 | Complex and compound tweets

Complex and compound tweets express multiple sentiments, and the system often produces inaccurate classification results in these instances. For example, the tweet "Life iz nothing but a thorn of roses" is labelled as neutral by our system. A mechanism is needed to address complex and compound tweets.

TABLE 12 Summary statistics of the Twitter data in three domains

Statistic	Health	Automobile	Laptop
Tweets	3,467	1,726	1,943
Average length (words/tweet)	25.43	23.32	22.86
Std. dev words/tweet	9.02	11.21	12.71
Min. words/tweet	1.00	1.00	1.00
Max. words/tweet	32.00	21.00	39.00
Total no. of tokenized items	41,029	22,621	29,361
Average tokens (tokens/tweet)	28.32	28.46	28.61
Std. dev tokens/tweet	8.11	7.02	9.61
Min. tokens/tweet	1.00	1.00	1.00
Max. tokens/tweet	42.00	51.00	48.00
Avg. emoticons/tweet	3.00	2.00	2.00
Avg. slang/tweet	2.00	1.00	2.00
Avg. domain-specific words/tweet	4.00	3.00	2.00

4.6 | Summary statistics of Twitter data

Table 12 presents a summary of the statistics of the Twitter data obtained from the three datasets. The three datasets contain about 7,136 tweets and 76,000 tokens. The average length of a tweet is almost the same in all the datasets, although the health tweets (25.43 words per tweet) are somewhat longer than the tweets in the other two datasets (23.32 and 22.86 words per tweet). The standard deviation of words in a tweet in a health dataset is low with respect to the other domains. The average number of tokens per tweet is 28.32, 28.46, and 28.61, respectively, in the three datasets. The smallest tweet in all three datasets consists of a single word, and the smallest tweets only consist of a single token (word/emoticon/slang). The standard deviation of tokenized instances in a tweet of an automobile dataset is lower than it is in the other two domains. The average number of emoticons per tweet is between 2 and 3. The average number of slangs per tweet is between and 1 and 2, and the average number of domain-specific words per tweet is between 2 and 4.

5 | CONCLUSION AND FUTURE WORK

This work presents the results of applying a multistage hybrid scheme of classification with a unified framework to detect and classify sentiments expressed by users in tweets. The proposed scheme consists of the following tasks: (a) data acquisition about products in different domains from Twitter; (b) preprocessing of the input text; (c) use of an SC to detect and score the slang expressed in tweets; (d) application of an EC to detect and score the emoticons expressed in tweets; (e) sentiment classification of opinion words using a general-purpose classifier based on SWN; (f) detection and classification of domain-specific words using a probability-based measure with a revised term weighting scheme; and (g) sentiment classification of tweets using general-purpose and domain-specific classifiers.

The proposed technique assists in classifying slang expressed in tweets, detects emoticons and classifies them using an enhanced emoticon dictionary, incorporates a GPSC module for classifying tweets using POS tags and scores retrieved from the SWN lexicon, and boosts the performance of the sentiment classifier by focusing on the domain-specific words in different domains. The improved results in terms of accuracy, precision, recall, and F-measure show that the proposed method's classification results are better than those of the baseline methods. The framework is generalized and can classify tweets in any domain.

A possible limitation of the proposed approach is the lack of automatic scoring of domain-specific words without performing a lookup operation in SWN, which may increase the classification accuracy. To minimize scoring inaccuracy in different domains due to the general-purpose nature of SWN, autoscoring techniques should be investigated. Another possible direction for enhancement is the addition of context-aware features for effective classification of tweets. Analysing the impact of sarcastic tweets on sentiment classification would be another interesting research direction.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

ORCID

Muhammad Zubair Asghar http://orcid.org/0000-0003-3320-2074

REFERENCES

Aldayel, H. K., & Azmi, A. M. (2016). Arabic tweets sentiment analysis—A hybrid scheme. *Journal of Information Science*, *42*(6), 782–797.

Muhammad, A., Wiratunga, N., & Lothian, R. (2016). Contextual sentiment analysis for social media genres. *Knowledge-Based Systems*, *108*, 92–101.

Asghar, M. Z., Khan, A., Ahmad, S., Khan, I. A., & Kundi, F. M. (2015). A unified framework for creating domain dependent polarity lexicons from user generated reviews. *PloS One*, *10*(10). e0140204

Asghar, M. Z., Khan, A., Ahmad, S., Qasim, M., & Khan, I. A. (2017). Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PloS One*, *12*(2). e0171649

Asghar, M. Z., Ahmad, S., Qasim, M., Zahra, S. R., & Kundi, F. M. (2016). SentiHealth: Creating health-related sentiment lexicon using hybrid approach. *SpringerPlus*, *5*(1), 1139.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *LREC*, *10*, 2200–2204.

Gu, B., Sheng, V. S., Tay, K. Y., Romano, W., & Li, S. (2015). Incremental support vector learning for ordinal regression. *IEEE Transactions on Neural networks and learning systems*, *26*(7), 1403–1416.

Gu, B., & Sheng, V. S. (2017). A robust regularization path algorithm for *v*-support vector classification. *IEEE Transactions on neural networks and learning systems*, *28*(5), 1241–1248.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. Beijing: O'Reilly Media, Inc.

Duwairi, R., & El-Orfali, M. (2014). A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *Journal of Information Science*, *40*(4), 501–513.

Goker, A., & Davies, J. (Eds) (2009). *Information retrieval: Searching in the 21st century*. USA: John Wiley & Sons.

Gu, B., Sheng, V. S., & Li, S. (2015). Bi-parameter space partition for cost-sensitive SVM. *IJCAI*, 3532–3539.

Ikeda, D., Takamura, H., Ratinov, L. A., & Okumura, M. (2008). Learning to shift the polarity of words for sentiment classification. *IJCNLP*, 296–303.

Khan, F. H., Bashir, S., & Qamar, U. (2014). TOM: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems*, *57*, 245–257.

Kundi, F. M., Ahmad, S., Khan, A., & Asghar, M. Z. (2014). Detection and scoring of internet slangs for sentiment analysis using SentiWordNet. *Life Science Journal*, *11*(9), 66–72.

Masud, F., Khan, A., Ahmad, S., & Asghar, M. Z. (2014). Lexicon-based sentiment analysis in the social web. *Journal of Basic and Applied Scientific Research*, *4*(6), 238–248.

Montejo-Ráez, A., Martínez-Cámara, E., Martín-Valdivia, M. T., & Urena-Lopez, L. A. (2012). Random walk weighting over sentiwordnet for sentiment polarity detection on twitter. In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis (pp. 3–10). Association for Computational Linguistics.

Paltoglou, G., & Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1386–1395). Association for Computational Linguistics.

Prieto, V. M., Matos, S., Alvarez, M., Cacheda, F., & Oliveira, J. L. (2014). Twitter: A good place to detect health conditions. *PloS One*, *9*(1). e86191

Poria, S., Cambria, E., & Gelbukh, A. F. (2015). Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. *EMNLP*, 2539–2544.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., & Mozetič, I. (2015). The effects of twitter sentiment on stock price returns. *PloS One*, *10*(9). e0138441

Ribeiro, P. L., Weigang, L., & Li, T. (2015). A unified approach for domain-specific tweet sentiment analysis. In *Information Fusion (Fusion), 2015 18th International Conference on* (pp. 846–853). IEEE.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of twitter. *Information Processing and Management*, *52*(1), 5–19.

Smeureanu, I., & Bucur, C. (2012). Applying supervised opinion mining techniques on online user reviews. *Informatica economica*, *16*(2), 81.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, *37*(2), 267–307.

Tang, J., Nobata, C., Dong, A., Chang, Y., & Liu, H. (2015). Propagation-based sentiment analysis for microblogging data. In *Proceedings of the 2015 SIAM International Conference on Data Mining* (pp. 577–585). Society for Industrial and Applied Mathematics.

Roesslein, J. (2015). Tweepy. *Python programming language module*.

Wen, X., Shao, L., Xue, Y., & Fang, W. (2015). A rapid learning algorithm for vehicle classification. *Information Sciences*, *295*, 395–406.

Zheng, Y., Jeon, B., Xu, D., Wu, Q. M., & Zhang, H. (2015). Image segmentation by generalized hierarchical fuzzy C-means algorithm. *Journal of Intelligent Fuzzy Systems*, *28*(2), 961–973.

Xia, Z., Wang, X., Sun, X., Liu, Q., & Xiong, N. (2016). Steganalysis of LSB matching using differences between nonadjacent pixels. *Multimedia Tools and Applications*, *75*(4), 1947–1962.

Muhammad Zubair Asghar is an HEC approved PhD supervisor recognized by Higher Education Commission (HEC), Pakistan. His PhD research includes recent issues in opinion mining and sentiment analysis, computational linguistics, and natural language processing. He is working as Assistant Professor in the Institute of Computing and Information Technology, Gomal University, D.I.Khan, Pakistan. He has authored more than 40 publications in journals of international repute (JCR and ISI indexed) and having more than 20 years of University teaching and laboratory experience in Artificial Intelligence and Research Methods in Computer Science.

Fazal Masud Kundi received his PhD from Gomal University, D.I.Khan. His research interests include recent trends in data mining, text mining, and computational linguistics. He has more than 40 publications in journals of international repute.

Shakeel Ahmad received his PhD from Gomal University and Post Doctorate from Malaysia. He remained Director IT and Director ICIT, Gomal University, D.I.Khan. Recently, he joined Faculty of Computing and Information Technology in Rabigh (FCITR), King Abdul Aziz University, Saudi

Arabia. His research interests includes software quality assurance and social media analytics. He has more than 60 publications in journals of international repute.

Aurangzeb Khan received his PhD from Malaysia. He is currently working as Chairman, Department of Computer Science, University of Science and Technology, Bannu (KP), Pakistan. His research interests includes text mining and opinion mining and sentiment analysis. He has more than 60 publications in journals of international repute. In addition to his research activities, he is acting as Director Academics in UST Bannu. He is considered as founder/pioneer scientist in opinion mining and sentiment analysis in Pakistan.

Furqan Khan received his MS degree in Computer Science from Gomal University. He is actively involved in the research area of machine learning, especially roughest theory and other white box classification algorithms.

How to cite this article: Asghar MZ, Kundi FM, Ahmad S, Khan A, Khan F. T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme. *Expert Systems*. 2018;35:e12233. https://doi.org/10.1111/exsy.12233