Classification of Short Text Using Various Preprocessing Techniques: An Empirical Evaluation

H. M. Keerthi Kumar and B. S. Harish

Abstract In recent decades, microblogs generate large volumes of data in the form of short text. Twitter has been one of the most widely used microblogging sites. Twitter data consist of noise due to shortness, which need to be preprocessed to find the accurate sentiment expressed by the user. The major challenges in short texts are the presence of noisy data like URLs, misspelling, slang words, repeated characters, punctuation, etc. To handle these challenges, this paper proposes to combine various preprocessing techniques with different classification methods as a tool for Twitter sentiment analysis. We evaluated the effect of noisy data like URLs, hashtags, negations, repeated characters, punctuations, stopwords and stemming. We use n-gram representation model to find the bindings and further applied support vector machine (SVM) and K-nearest neighbors (KNN) multi-class classifiers for sentiment classification. Experiments are conducted to observe the effect of various preprocessing techniques on Stanford Twitter Sentiment Dataset. The extensive experimental results are presented to show the effect of various preprocessing techniques to classify short texts.

Keywords Short text · Preprocessing · Support vector machine
K-Nearest neighbor · Classification

1 Introduction

In recent years, microblogs play a vital role in information sharing and communication. Microblog is a form of multimedia blogging that allows users to send brief text updates or micro-media such as photos or audio clips and publish them.

H. M. Keerthi Kumar (✉)
JSS Research Foundation, Mysuru, Karnataka, India
e-mail: hmkeerthikumar@gmail.com

B. S. Harish
Sri Jayachamarajendra College of Engineering, Mysuru, Karnataka, India
e-mail: bsharish@sjce.ac.in

P. K. Sa et al. (eds.), *Recent Findings in Intelligent Computing Techniques*,
Advances in Intelligent Systems and Computing 709,
https://doi.org/10.1007/978-981-10-8633-5_3

Many micro-blog sites like Tumblr, Twitter, Posterous, FriendFeed, etc., are used for information sharing and communication. Twitter is one of the most popular and commonly used microblogging services. Twitters are accessible through website interface and numerous mobile devices. Millions of users are sharing information on various topics ranging from political debate, products, stock market, etc. On Twitter, users post and read messages are restricted to 140-characters, which are called "tweets" [5]. In tweets, users share their views and opinions known as "sentiment", in the form of text, photos, and audio clips, where text shares a major part in communication. These tweets hold the key for determining sentiment of a population. By analyzing sentiment on tweets, we can identify the kind of emotions, mainly as positive, negative or neutral.

Sentiment analysis is treated as a classification task as it classifies the orientation of tweets into different classes or polarity [3, 21]. Sentiment classification methods can be classified into machine learning, lexicon based methods, and linguistic methods [18]. Many researchers [7, 12, 13] have claimed that lexicon-based methods and linguistic methods do not perform well on sentiment classification, due to nature of an opinionated text which requires more understanding of text. However, the occurrence of some keywords could be the key for an accurate classification [10]. In sentiment analysis, machine learning methods are used to train an algorithm based on a set of keywords or features, which describes the polarity and then test on another set whether it is able to detect the keywords and give the accurate classification. Machine learning classifiers such as Naive Bayes (NB), maximum entropy (ME), and support vector machine (SVM) are used in [13] for sentiment classification.

Twitter sentiment analysis using machine learning techniques encompasses tasks such as preprocessing, feature extraction and selection, representation, classification or clustering and evaluation. In tweets, users make spelling mistakes, slang words and use emoticons for expressing their views. Moreover, tweets contain a large amount of noise data, such as URLs, punctuation, etc. [3], which need to be preprocessed. In this paper, we are exhibiting the impact of preprocessing in determining sentiment on tweets. However, this work concentrates more on conventional preprocessing techniques used to eliminate noisy data which do not contribute enough to Twitter sentiment classification. Although few works concentrate on twitter preprocessing techniques, still generic solution needs to be developed for efficient classification. In this work, we focus on exploring preprocessing techniques to uplift the performance of sentiment classification, including the effect of URLs, usernames, hashtags, negations, repeater characters normalization, punctuation, stopword removal and stemming. The experimentations are performed rigorously on Stanford Twitter Sentiment Dataset [1] to show that sentiment accuracy increases when URLs are removed, username elimination, Hashtag content retained, negation transformation and repeated character normalization. We also represented feature space in unigram, bigram, and trigram representation and further applied support vector machine (SVM) and K-nearest neighbors (KNN) multi-class classifiers for sentiment classification.

The rest of the paper is organized as follows: Sect. 2 explains related work on sentiment analysis on Twitter data. Section 3 portrays the methodology used in this paper. Section 4 contains experimentation and related results along with the discussion. Finally, we conclude our work with outlining future work in Sect. 5.

2 Related Work

Over the time, microblogs are used for expressing sentiments on an event or topic. Twitter is one of the most commonly used micro-blog to express sentiment over the current issues. Many researchers concentrated their studies to understand sentiments expressed in twitter. Twitter contain a large amount of noisy data, such as URLs, user names, punctuations symbols, etc. These characters make sentiment classification a bit difficult and challenging and thus preprocessing plays a vital role in Twitter sentiment analysis.

Based on the research work by many researchers, it has been proved that the preprocessing is the main aspect in sentiment analysis [4, 5, 10, 16, 20]. To deal with these, many researchers proposed various preprocessing techniques along with the algorithm based on supervised, semi-supervised, and unsupervised machine learning approaches with lexicon-based approaches. Bao et al. [4] described the effectiveness of preprocessing techniques on Twitter data. The method uses unigram, bigram representation with Liblinear classifier to classify the data into positive and negative classes. The experiment shows that noisy data like URLs, negation transformation and repeated characters normalization have a positive impact on classification accuracy while stemming has a negative impact. Singh et al. [16] brief the role of text preprocessing in Twitter sentiment analysis. The method explains text normalization as the process of purification of tweets, where each step eliminates the noise data. This method defines the significance and sentiment strength of slang and unidentified words in tweets. Support vector machine (SVM) classifier is used to evaluate and measure the impact of preprocessing on sentiment classification.

In [10], Haddi et al. describe the sentiment analysis on online movie reviews. The different preprocessing methods are used to reduce noise in the text. The results of preprocessing techniques show that data transformation and filtering can significantly enhance the performance of SVM classifier on sentiment identification. Uysal and Gunal [20] explore the impact of preprocessing on text classification. Sentiment analysis was conducted on Turkish and English languages by choosing appropriate preprocessing task such as tokenization, stopword removal, lowercase conversion, and stemming. By employing preprocessing, a significant improvement was found on classification accuracy whereas inappropriate combinations resulted in degrading the accuracy. Tripathy et al. [19] used machine learning techniques such as naive Bayes (NB), maximum entropy (ME), SVM, and stochastic gradient descent (SGD) classification using n-gram approach. Unigram, bigram, trigram models and their combinations are used for classification on IMDb movie review

dataset. The accuracy of different methods is examined in order to access their performance on the basis of parameters such as precision, recall, f-measure, and accuracy. Agarwal et al. [3] applied novel approaches to preprocess tweets. The methods replace URL, target name, negations, and repeated characters with appropriate terms. The results of their experiment illustrated that appropriate text preprocessing methods can significantly increase the accuracy of the classifier. Saif et al. [15] described the role of preprocessing to reduce sparsity issue in twitter sentiment analysis. Experiment results illustrated that appropriate text preprocessing techniques can significantly reduce sparsity and increases the classification accuracy.

In literature, many researchers [3, 10, 15, 16, 19, 20] described the role of preprocessing techniques by selecting the appropriate combination of techniques to improve the classification performance. The Twitter data consists of URLs, slang words, misspellings, punctuation, and abbreviations which make preprocessing a challenging task. By eliminating the above noisy data, we can reduce misclassification in sentiment analysis. Although various preprocessing techniques exist in the literature, the problem of sentiment classification on short text is still challenging, with no generic solution and remains an open research area. In this work, we perform extensive experimentation to show the impact of preprocessing techniques on Stanford Twitter Sentiment Dataset.

3 Methodology

The text preprocessing is the initial step in sentiment analysis, where noisy data are eliminated from the dataset. Here, we apply various preprocessing techniques to reduce the noisy data. We adopt the following process for Twitter sentiment analysis.

3.1 Tweet Preprocessing

In this step, we are removing or replacing noisy data in each tweet, which do not contribute much for sentiment classification. We are using eight traits to process tweets, namely URL removal, username replace with white space, hashtag removal, handle negation, characters normalization, punctuation removal, stopword elimination and stemming.

URL removal: In tweets, the user posts URL along with text to provide supporting information about the text like "http://bit.ly/IMXUM". These URL links become noisy data during sentiment analysis. We are eliminating URL links in each tweet and replacing it with a space.

Username: There are usernames like "@LATimesautos", "@XPhile1908", that start with symbol "@", the symbols indicate the username or target person. Here, we are concentrating our work toward finding the sentiment on each tweet and not on any targeted persons. The contribution of username is less on sentiment analysis, so we replace all username with a white space.

Hashtags: Hashtags marked with the symbol "#", which means that tweets are associated with the particular topic and also consists opinion expressed in the tweets. We removed only symbol "#", retaining the contents.

Handling negation: Negations play a vital role in sentiment classification, the negative word, e.g., "not", "n't", etc., in which co-occurrence with other word changes the orientation of text into different polarity. Considering the effect of negation, we applied abbreviations for short terms like "don't", "can't", "n't", etc., terms to "do not", "cannot", "not", etc., words, respectively, which changes the sentiment of the tweet.

Character normalization: Words with consecutive characters, e.g., "looovvv-veee", are more common in tweets and users tend to use this way to express their opinion or sentiments. Thus, it is necessary to deal with these words to make them more formal. Here, consecutive characters mean repeated characters more than 3 times in a word. This needs to be normalized to give a formal representation. Here, we replace repeated characters more than three times to single characters.

Punctuation: We removed all the punctuations symbols like ",", " ' ", "$", "?", "!", etc., from the dataset, which does not contribute to the sentiment of tweets.

Stopwords: Stopword refer to most common words used in the tweets like a, all, am, an, and, any, are, etc. [9]. We eliminated these English stopwords, which contribute less to the sentiment of tweet.

Stemming: Stemming is used to achieve feature reduction. Stemming is the process of bend words to their root or base form, by eliminating words ending with "er", "ing", "ed", etc. [16]. Here, we apply stemming to reduce the feature space.

Emotion symbol is used to express sentiment in tweets, e.g., ":(, =]" means sad or negative emotion. Here in our work, we are considering only the sentiment related to text so we replace these emotions symbol with white space. We eliminated all symbols, digits, single character, and other nonalphabetic symbols.

In this work, we are demonstrating the role of preprocessing in Twitter sentiment analysis by observing the impact on sentiment classification accuracy. The combinations of techniques are applied to identify the impact of methods in sentiment classification.

3.2 Representation

Preprocessed tweets are represented using n-gram representation model [8]. N-gram is a contiguous sequence of n number of words. In this case, each n-gram is one features space whose dimension is equal to the number of n-grams [11]. When n = 1, it represents unigram, where each word represents a feature. Similarly, for n = 2, 3 represent bigram, trigram, respectively. Weight values are associated with each pair using the term frequency (TF) scheme. The unigram, bigram, and trigram feature representations are used to represent the preprocessed tweets. We have used TF to find the number of terms appeared in each tweets.

3.3 Classification

In this work, we employ multi-class support vector machine (SVM) and K-nearest neighbor (KNN) to train the preprocessed tweets. The SVM algorithm has several advantages, which are important for learning a sentiment classifier from a large Twitter dataset [6, 17]. SVM is a widely used classifier in sentiment analysis tasks. It can effectively conduct classification task in high-dimensional feature space [14]. In this paper, we used SVM for multi-class problem to classify the tweet into positive, negative or neutral. On the other hand, KNN is also used as nonparametric method used for pattern classification. KNN classification is based on the class of their closest neighbors, most often, more than one neighbor is taken into consideration, and here K denotes the number of neighbors taken into account in determining the class [2].

4 Experimental Results and Discussion

In this section, we explore the results obtained when various types of preprocessing techniques are applied to Stanford Twitter datasets.

4.1 Dataset Description

The experiment was carried out on Stanford Twitter Sentiment Dataset [16]. The dataset is in English language which consists of 498 tweets, where 182 positive, 177 negative, and 139 neutral tweets. Each tweet comes with the labels: positive, negative, and neutral.

4.2 Experimentation

In the experiment, we carried out step-by-step process to evaluate the impact of preprocessing methods on sentiment classification. The Stanford Twitter Sentiment Dataset of labeled 498 tweets are taken randomly for training and testing purpose. The equal propositions of positive, negative, and neutral tweets are taken for training and testing data. Experimentation was conducted using the combination of 50 training and 50 testing, 60 training and 40 testing, and 70 training and 30 testing bases, respectively. For classification purpose, we used multi-class SVM and KNN classifier for multi-class problem. The classification accuracy is considered as a metric to evaluate the individual and combination of various preprocessing techniques. The experimentation was conducted using R Studio Version 0.99.903 and R-3.1.3 Language to perform sentiment analysis on Stanford Twitter Sentiment Dataset.

In the first step, we removed all the URLs in each tweet and represented using n-gram feature representation. The representation includes unigram, bigram and trigram on each process. Table 1 shows the effect of URLs on the classification accuracy. The result shows KNN classifier on unigram provide better accuracy when compared to bigram and trigram feature representation. The SVM classifier gives better accuracy in bigram when training and testing ratio is increased.

In next method, we removed username which starts with "@" symbol. From Table 2, we can see the effect of username (@) removal from the dataset. The results show the accuracy of SVM classifier increases when the username is removed. The results are obtained for unigram, bigram, and trigram feature representation.

Table 1 Classification accuracy for URL removal

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**68.80**	63.60	36.40
	60:40	**73.00**	68.50	47.00
	70:30	75.40	**85.40**	35.70
KNN	50:50	**77.60**	66.40	65.20
	60:40	**87.00**	78.50	77.50
	70:30	**88.00**	76.10	75.40

Table 2 Classification accuracy for username removal

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**76.00**	74.40	50.80
	60:40	68.50	**78.00**	37.50
	70:30	64.90	**73.50**	72.80
KNN	50:50	**76.50**	65.60	62.80
	60:40	**83.50**	71.00	69.00
	70:30	**86.09**	76.80	76.15

Continuing the process of preprocessing, we removed Hashtag "#" symbol and retained the content. Table 3 shows the impact of hashtags (#) removal on classification accuracy. The result increases in the accuracy using KNN classifier for unigram and bigram feature representation.

Further, we carried out by combining first two techniques, i.e., URL and username removal. Table 4 presents the results of URLs and username removal from the dataset. We get better results after bigram features are affiliated with feature space.

To continue experimentations by reducing features from the original feature space, we applied three methods jointly, i.e., URL, username, and hashtags. Table 5 shows the result of a combination of techniques, which gives average accuracy when compared to Tables 1, 2 and 3.

Table 3 Classification accuracy for Hashtag (#) removal

Classifiers	Training: Testing	Unigram	Bigram	Trigram
SVM	50:50	66.00	**70.80**	54.00
	60:40	67.00	**69.00**	50.50
	70:30	**68.80**	68.20	65.50
KNN	50:50	**77.60**	72.40	71.60
	60:40	**79.00**	67.00	67.00
	70:30	**88.70**	84.10	80.39

Table 4 Classification accuracy for URL and username removal

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**74.00**	69.20	37.60
	60:40	69.00	**77.00**	36.50
	70:30	70.10	**81.40**	72.84
KNN	50:50	**77.20**	64.40	60.80
	60:40	**81.00**	70.00	68.00
	70:30	**80.70**	74.83	76.15

Table 5 Classification accuracy for URL and username removal and hashtags

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**72.00**	69.20	57.20
	60:40	74.00	**75.00**	63.00
	70:30	74.10	**79.40**	70.86
KNN	50:50	**75.60**	65.20	64.00
	60:40	**79.50**	76.00	73.00
	70:30	**86.09**	77.40	77.48

In consideration of the effect of negation, we applied abbreviations for short terms like “don’t”, “can’t”, “n’t”, etc., terms to “do not”, “cannot”, “not”, etc., words respectively, which changes the sentiment of the tweet. Table 6 shows the results of handling negations with other three methods jointly. The experimental results increases when negations are applied on unigram, bigram representation of dataset.

Words with consecutive characters, e.g., “looooooooovvvvvveee”, are more common in tweets, and users tend to use this way to express their opinion or sentiments. Thus, it is necessary to deal with these words to make them more formal. Here consecutive character means repeated characters more than three times in a word. This needs to be normalized to give formal representation. Table 7 shows the result after performing normalization of characters with other methods jointly.

In the next set of experiments, we eliminated the punctuation’s present in the corpus and combined it with previous methods. Table 8 presents the results after removing punctuations in the dataset. The punctuations are used in tweets to express the strong feeling toward the polarity but here we are considering tweets related to positive, negative, or neutral. In our work, punctuation becomes noisy data, after eliminating punctuation there is an increment in the accuracy of unigram and bigram representation.

To continue with reducing features from the original feature space, we introduced stopword removal to the dataset. Table 9 shows the effect of stop words removal along with previous techniques. There is a sharp decline in classification accuracy of KNN classifier. The results illustrate that stop words contribute less toward sentiment classification, when applied jointly with other preprocessing methods.

Table 6 Classification accuracy for URL and username removal, hashtags, and negation

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**74.80**	69.20	38.00
	60:40	**78.50**	74.50	45.00
	70:30	80.79	**86.75**	54.30
KNN	50:50	**74.00**	64.00	63.60
	60:40	**86.50**	70.50	67.50
	70:30	**90.00**	79.47	78.80

Table 7 Classification accuracy for URL and username removal, hashtags, negation, and character normalization

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**66.80**	54.40	64.00
	60:40	74.50	**78.50**	62.00
	70:30	80.13	**81.45**	79.40
KNN	50:50	**73.60**	62.00	61.20
	60:40	**81.00**	72.50	70.50
	70:30	**82.78**	76.15	76.80

Table 8 Classification accuracy obtained for URL and username removal, hashtags, negation, and character normalization, punctuation

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**72.40**	71.20	52.00
	60:40	**80.00**	79.50	68.50
	70:30	74.83	**80.13**	76.82
KNN	50:50	**76.80**	65.60	63.60
	60:40	**82.00**	74.00	73.50
	70:30	**90.00**	80.79	80.13

Table 9 Classification accuracy for URL and username removal, hashtags, negation and character normalization, punctuation, and stopwords removal

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**74.40**	60.40	36.00
	60:40	**79.00**	76.00	56.50
	70:30	**82.11**	80.13	65.56
KNN	50:50	**75.60**	68.00	68.00
	60:40	**80.50**	75.50	75.00
	70:30	**88.07**	85.43	76.80

Table 10 Classification Accuracy for URL and username removal, hashtags, negation and character normalization, punctuation, stopwords removal and steaming

Classifiers	Training: testing	Unigram	Bigram	Trigram
SVM	50:50	**80.00**	68.00	20.40
	60:40	**83.50**	61.00	36.00
	70:30	**88.07**	82.78	36.40
KNN	50:50	**76.50**	70.40	68.00
	60:40	**87.50**	77.00	76.00
	70:30	**84.70**	82.11	79.47

Stemming is also used to achieve feature reduction. Table 10 reveals the result of stemming with other processes jointly. The result shows increase in accuracy in both unigram and bigram representation. The experimental results indicate that the preprocessing is a basic step for sentiment classification. The step-by-step processes of applying preprocessing methods increases the accuracy of classification.

4.3 Discussion

The preprocessing techniques are applied to eliminate noisy data and normalize the dataset. In this work, we applied eight text preprocessing methods to normalize the corpus. Each method exhibits a substantial amount of effects on the classification accuracy. Tables 1, 2, 3 shows the results related to URL, username, and hashtag

removal, respectively. The result shows that there is an increase in accuracy when we apply unigram, bigram and classify using SVM and KNN classifiers. From Tables 4, 5, 6, 7, 8 and 9, we applied the combination of various preprocessing techniques to normalize the dataset. The results show a slight increase as well as a decrease in the classification accuracy on the classifiers with respective n-gram representation. Table 10 gives the overall result of various preprocessing techniques applied on the dataset. The overall result illustrates the performance of sentiment classification increases when URL removal, username replace with white space, hashtag removal, negation, character normalization, punctuation removal, stopword elimination and stemming are applied. We observe that various preprocessing techniques clearly indicate an increase in performance of the classifiers with unigram and bigram representations.

5 Conclusion and Future Work

In Twitter, sentiment analysis has become a recent and meaningful topic for researchers. The length limitation, various topic discussion, informal language, slang words and rich in symbols, all these characters of tweet make sentiment analysis a challenging. In this paper, we conducted a series of experimentation to verify the effectiveness of various preprocessing techniques on Stanford Twitter Sentiment Dataset. We used preprocessing techniques like URL removal, username replace with white space, hashtag removal, negation handling, character normalization, punctuation removal, stopword elimination and stemming. We demonstrated the role of various preprocessing techniques in Twitter sentiment classification.

In future, we would like to incorporate natural language processing (NLP) based text preprocessing techniques like lemmatization techniques, part-of-speech (POS) tags, etc., for topic-based sentiment analysis and also include different feature selection methods which enhance the classification performance.

References

1. http://help.sentiment140.com/for-students/
2. Adeniyi, D., Wei, Z., Yongquan, Y.: Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method. Appl. Comput. Inform. **12**(1), 90–108 (2016)
3. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media, pp. 30–38. Association for Computational Linguistics (2011)
4. Bao, Y., Quan, C., Wang, L., Ren, F.: The role of pre-processing in twitter sentiment analysis. In: International Conference on Intelligent Computing, pp. 615–624. Springer (2014)

5. Bhuta, S., Doshi, A., Doshi, U., Narvekar, M.: A review of techniques for sentiment analysis of twitter data. In: 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), pp. 583–591. IEEE (2014)
6. Chang, C.C., Lin, C.J.: LibSVM: a library for support vector machines. ACM Trans. Intell. Syst. (TIST) **2**(3), 27 (2011)
7. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 231–240. ACM (2008)
8. Fusilier, D.H., Montes-y Gomez, M., Rosso, P., Cabrera, R.G.: Detecting positive and negative deceptive opinions using pu-learning. Inf. Process. Manage. **51**(4), 433–443 (2015)
9. Ghag, K.V., Shah, K.: Comparative analysis of effect of stopwords removal on sentiment classification. In: 2015 International Conference on Computer, Communication and Control (IC4), pp. 1–6. IEEE (2015)
10. Haddi, E., Liu, X., Shi, Y.: The role of text pre-processing in sentiment analysis. Procedia Comput. Sci. **17**, 26–32 (2013)
11. Lima, A.C.E., de Castro, L.N., Corchado, J.M.: A polarity analysis framework for twitter messages. Appl. Math. Comput. **270**, 756–767 (2015)
12. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1275–1284. ACM (2009)
13. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical methods in Natural Language Processing-Volume 10, pp. 79–86. Association for Computational Linguistics (2002)
14. Ren, Y., Wang, R., Ji, D.: A topic-enhanced word embedding for twitter sentiment classification. Inf. Sci. **369**, 188–198 (2016)
15. Saif, H., He, Y., Alani, H.: Alleviating data sparsity for twitter sentiment analysis. In: CEUR Workshop Proceedings (CEUR-WS. org) (2012)
16. Singh, T., Kumari, M.: Role of text pre-processing in twitter sentiment analysis. Proced. Comput. Sci. **89**, 549–554 (2016)
17. Smailovi_c, J., Gr_car, M., Lavra_c, N., _Znidar_si_c, M.: Stream-based active learning for sentiment analysis in the _nancial domain. Information Sciences 285, 181–203 (2014)
18. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. J. Am. Soc. Inform. Sci. Technol. **62**(2), 406–418 (2011)
19. Tripathy, A., Agrawal, A., Rath, S.K.: Classification of sentiment reviews using n-gram machine learning approach. Expert Syst. Appl. **57**, 117–126 (2016)
20. Uysal, A.K., Gunal, S.: The impact of preprocessing on text classification. Inf. Process. Manage. **50**(1), 104–112 (2014)
21. Zainuddin, N., Selamat, A.: Sentiment analysis using support vector machine. In: 2014 International Conference on Computer, Communications, and Control Technology (I4CT), pp. 333–337. IEEE (2014)