노이즈 소셜미디어 텍스트를 활용한 감성분석 개선

칸 제브란◆, 이 성 창°

Enhancement of Sentiment Analysis by Utilizing Noisy Social Media Texts

Jebran Khan◆, Sungchang Lee°

요 약

본 논문에서는 노이지 텍스트(noisy text)들을 제거하지 않고 정정하여 활용함으로써 소셜미디어 텍스트의 감성분석 (sentiment analysis)를 개선하는 새로운 방법을 제안하였다. 소셜미디어 텍스트는 특정 언어에 대한 작성자의 한정된 언어능력, 인지적 혹은 오타의 철자오류, 약자나 줄여쓰기의 사용, 감정의 표현 등으로 인해서 수많은 변형과 비일관성을 가진다. 이러한 변형들은 사전적으로 근거없는 (misspelled words) 단어가 되어 노이즈 (noise)로 간주된다. 대부분의 기존의 방식들은 이러한 노이즈를 여과하여 제거하거나 혹은 이 변형들을 위한 별도의 사전을 만든다. 전자에서는 감성 정보의 손실을 초래하며, 후자는 특정 응용 별로 의존적인 단점이 있다. 본 논문에서는 형식을 갖추지 않은 소셜미디어 텍스트에서의 비일관적 변형들을 자동적으로 처리할 수 있는 포괄적 방안을 제시하였다. 제안된 방안에서는, 노이지 텍스트들을 제거하는 대신에 이들을 바로잡는 정정을 함으로써 이들에 포함된 감성정보를 보존하여 사용하며, 앙상블 (ensembled) 감성 분석기법을 비형식 단어들을 정정하기 위한 맞춤법 검사 (enhanced spell-checking) 기법과 통합한다. 시뮬레이션을 통해서 제안한 기법이 최신의 감성분석 기법들보다 성능이 우수함을 보였다.

키워드 : 감성분석, 소셜네트워크, 맞춤법 검사, N-gram, Edit Distance, 나이브 베이지안, SVM

Key Words : Sentiment Analysis, Social Networks, Spell-checking, N-gram, Edit Distance, Naïve Bayesian, SVM

ABSTRACT

We proposed a new method to enhance sentiment analysis of social media text by utilizing the noisy text instead of filtering them out. The social media text contains numerous variations and inconsistencies due to the author's limited vocabulary in a specific language, cognitive and typo spell-errors, abbreviating and shortening, and expression of emotions. Such variations result in generating lexically invalid words (misspelled words), which are considered as noise. Most of the existing work either filter out such noise or create a lexicon of these variations. The former method results in sentiment information loss while the latter method is highly application dependent. In this work, we propose a generic approach that can automatically handle the inconsistencies in the informal social media text. The proposed method conserves the sentiment information present in the noisy text

※ Supported in Basic Science Research Programs through NRF Korea funded by Ministry of Education (NRF-2017R1A2B1010817)
◆ First Author : Department of Information and Communication Engineering, Korea Aerospace University, South Korea(KAU), jebran.khan@kau.kr, 학생(박사), 학생회원
° Corresponding Author : Department of Information and Communication Engineering, Korea Aerospace University, South Korea(KAU), sclee@kau.ac.kr, 정교수, 종신회원
논문번호 : 202003-050-B-RN, Received March 9, 2020; Revised April 1, 2020; Accepted April 2, 2020

by correcting the useful noisy terms rather than eliminating them. In the simulation, we integrated an ensembled sentiment analysis technique with an enhanced spell-checking technique for the correction of invalid words. The proposed scheme outperforms the state-of-the-art sentiment analysis schemes in terms of precision, recall, accuracy, and F1-score in simulation.

Ⅰ. Introduction

Web 2.0 has enabled users to generate enormous data in the form of text, videos, audio and images on the online social networks, smartphone applications, blogs, review sites, etc. These online portals have become an essential part of human life and users spend most of their time online. These online portals are open to all and people from different background and regions shares their views, reviews, experiences and emotions on these portals in the form of blogs, comments, posts, reviews on products/items, text messages and images with text.

The text generated by the users can be of different polarities and can express their emotions and attitude towards the target scenario. These polarities are semantic and depend on the application and situations e.g. is the product review positive or negative, is the customer requirements satisfied or not satisfied, how people respond to some ad/item/news and what is the bloggers' attitude towards some situation, etc. Finding the polarities and attitudes from text is done through sentiment analysis (SA) of the text generated by the users.

Sentiment analysis determines the evaluative nature of a piece of information. Sentiment analysis is applied in several applications, such as customer opinion mining towards products, movies, books, etc., tracking political views, customer relation improvement models, happiness and well-being detection, and auto-dialogue/response systems improvisation. There has been an exponential growth in the use of micro-blogging services such as Twitter, Facebook due to mobile access to the World Wide Web. This developed an interest in opinion mining of informal communication, such as tweets, posts, comments, and SMS messages, across different domains.

Online text can be categorized into formal and informal text information. The formal text generally has less variation as compared to the informal text generated by the users. Social media and smartphones are generally used for informal communication. The informal text is generally short, twitter allows 140 characters, which has lots of variations due to using abbreviations, shortening of text, slangs, wide range of topics to discuss, large vocabulary pool, special strings, emotions expression, and typing errors. These results in different variants of words in a language, which may not be correct according to the rules of that language, but their use is so frequent and accepted worldwide, therefore they are considered to be correct and sometimes more expressive.

This paper aims to deal with this problem of sentiment analysis of informal and abbreviated social media text. The system is based on the spell-checking techniques like dictionary lookup, edit distance and n-grams and supervised text classification techniques such as Naïve Bayesian and SVM.

The rest of the paper is organized as follows. In Section 2 we have discussed the related work. Next, in Section 3 we discussed the methodology and description of the techniques used in this work. In Section 4 we have discussed the dataset, results, and performance of the proposed scheme. Section 5 is the conclusion of this work.

Ⅱ. Background Study

Sentiment analysis is one of the most prominent and popular research areas over the last decade. Many studies have been conducted to explore different facets of the sentiment analysis i.e. sentences categorization as subjective and objective, positive, negative and neutral, personal behavior detection from the text, emotions detection, sentiments visualization, and applying sentiment analysis to areas of commerce, healthcare, and disaster management. A survey by Pang et. al.[1] and Liu et. al.[2] gives a summary of these studies in the area.

Variants of sentiment analysis have been applied to a variety of texts in different works including customer review[3], weather reports[4], newspaper headlines[3], novels[5-8], emails[8,9], tweets[10] and blogs[11-13]. Often these systems outfit specific needs, which suits well to the target application, of the text such as style (formal/informal), frequencies of words, length of utterances, etc. In the context of social media, the sentiment analysis systems developed especially for tweets includes those in researches referenced as[14-18]. A survey by Martínez-Cámara et. al.[19] provides a detailed overview of the opinion mining from tweets.

As discussed in Section I, the text generated in informal communication may contain a wide range of variations. Most of these variations result in the spell-errors. The text with spell-errors is not recognized and corrected by the sentiment analysis systems previously developed. In some studies, these variations are dealt with by creating a manual list of variants of words[20] or creating lexicon[21] from the available dataset. But such type of list creation is not enough to be a general solution to this problem due to vast variations in a social media text. Therefore, in this work, we aim to integrate the spell-checking techniques with the sentiment analysis approach to deal with the informal/abbreviated text data problems.

The process of detection and correction of the misspelled words, that are not valid in a language, in a document is known as spell-checking. The spell-checking process is composed of two phases i.e. error detection and error correction.

Spell-errors are found when one or more words, in a target document, are not found in a list of valid words. This list of valid words is referred to as the lexicon of a language[22]. Until now, most spelling detection and correction techniques are proposed based on patterns and trends of errors; so many studies were carried out in order to analyze the trends and types of errors[23]. And the most famous among these studies are those that are performed by Damerau[24] and Peterson[25].

These studies showed that spell-errors can be divided into two types, i.e. typographic (when the writer knows the correct spell, but the error is due to mistyping) and cognitive (when the writer has no idea of what the correct spelling is or has forgotten it)[23].

According to Damerau[24], in typographic errors, 80% of the errors are one of the following types; insertion error, deletion error, transposition error, and Substitution error. This was later confirmed by Peterson[25].

Other factors that can result in the misspelled word may be due to usage of a short form of words like “gd mrng” instead of good morning, “grt” instead of great. These words are generally formed by deleting the vowels from the correct dictionary words or replace the letters by some phonetically similar characters. The misspelled words can be a result of some text written in more intense emotions like “grrrrrreeeaattttt” instead of great, “aaawwwwwww” instead of “aw”.

The techniques that are used for spell correction are edit distance, Similarity key, rule-based, N-gram, probabilistic and neural network. In many languages, the edit distance technique is the most commonly studied and used[26,27]. The edit distance technique operates by computing the minimum number of editing operations i.e. insertion, deletion, substitution and transposition, which are required to convert the misspelled word invalid dictionary word[27]. A similarity key technique[26,27] is designed to transform words into keys and operates on the principle that words with similar spells will have similar keys. Hence computing keys for misspelled words will pick out the words with similar spells in the lexicon. A rule-based technique[26,27] involves algorithms that represent common spell-error trends as rules, which are used to convert the misspelled words into correct ones. An N-gram based technique as described above is either used as a standalone or in conjunction with other techniques to perform error correction. A probabilistic technique[27] based on the N-gram technique. A neural network[28] technique capable of doing associative recall based on incomplete input[26].

In this work, we apply the isolated word spell-error correction to the sentiment analysis system in order to deal with the problems exist in the sentiment analysis of informal/abbreviated social media text.

Ⅲ. Proposed Sentiment Analysis Method

We proposed a new strategy for enhancement of sentiment analysis of informal social network textual conversation/posts by utilizing the sentiment information in the noisy text which are usually filtered out. We integrated an ensembled text classification method with spell-checking algorithm to enhance the performance of sentiment estimation system for noisy textual data. The block diagram in Figure 1, shows the methodology of our proposed sentiment analysis scheme. The proposed model consists of two phases, i.e. spell-checking phase and sentiment analysis phase.

Algorithm 1 shows the stepwise approach of the proposed model. The social network text is divided into training and test dataset. The training and test datasets are then preprocessed to clean, tokenize,

Algorithm 1: A stepwise proposed algorithm for the sentiment analysis of informal/abbreviated:

Step1: Pass the social network text data (test data) to the system

Step2: Split the dataset in training, and test data.

Step3: Preprocess the training and test data, tokenization, lemmatization, filtering, and data cleaning i.e., remove stop words.

Step4: Train the ***Naïve Bayesian*** and ***SVM*** classifiers.

Step5: Analyze the test data to extract the invalid words (misspelled words), using dictionary lookup algorithm.

Step6: Apply the spell-correction technique such as proposed in ***Algorithm 2***. The error-correction algorithm will replace the misspelled words with closest valid dictionary words.

Step7: Input the text data from ***Step6*** to the classifiers to find the polarity of the text returned by each classifier.

Step8: Final polarity of the text is decided based on Equation 1.

and lemmatize the text leaving the noisy text. In the next step the sentiment analysis models are trained

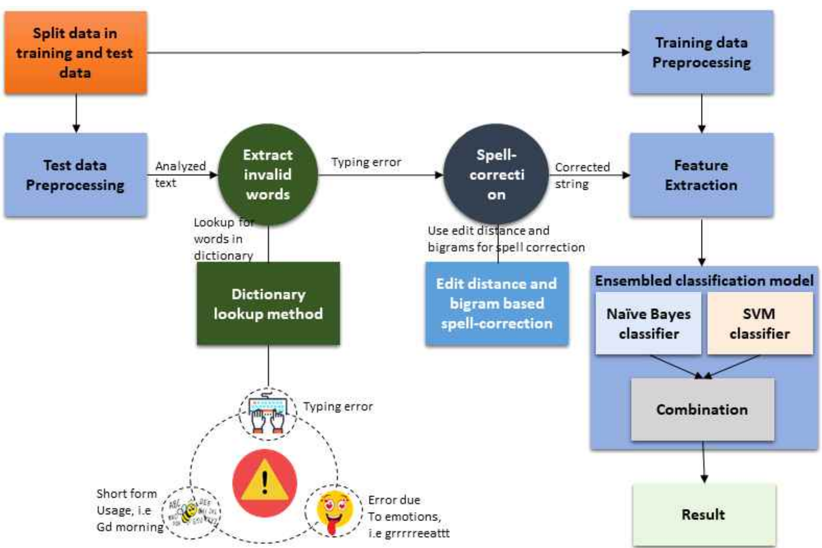

Fig. 1. Block diagram of the proposed sentiment analysis scheme for informal social networks text.

by using the training data. The preprocessed test dataset is passed from the spell-checker, where the text is checked to identify and correct the misspelled words in the text, as shown in Figure 2 and Algorithm 2.

The Algorithm 2, shows the stepwise approach for spell-checking process. The block diagram of spell-checker is shown in Figure 2. In this method the input text is passed from the spell-checker, where *ErrorDetection(D[i],dict)* function checks each word *D[i]*, in the input document *D*, against a dictionary *dict* of valid words to check the validity of the words in the input document *D*. If the function return *false* a misspelled word is detected in the document. When misspelled words are found the words are passed from spell-correction *SpellCorrection(D[i],dict)* function to find a valid replacement of the misspelled words.

The spell-correction *SpellCorrection(D[i],dict)* is a combination of two widely used strings proximity estimation methods, i.e., edit distance and bi-grams. The edit-distance provide a list of correct words from dictionary which are nearest neighbors of the

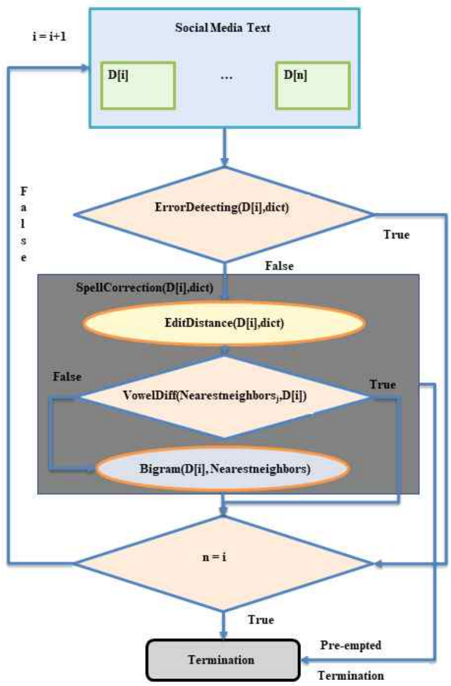

Fig. 2. Block Diagram for spell-checking process.

Algorithm 2: A stepwise algorithm for enhanced spell-checking:

Step1: initialize *i=0*, *D[i]* is the word to be checked

Step2: if *ErrorDetection(D[i],dict)* returns *true*, jump to *Step4*. If *ErrorDetection(D[i],dict)* returns *false* continue to *Step3*.

Step3: The function *SpellCorrection(D[i],dict)* return a closest replacement for the misspelled word. *EditDistance(D[i],dict)* returns a list of word (*Nearestneighbors*) which have least number of editing distance, with the misspelled word, from the lexicon. If the function *VowelDiff(Nearestneighbors$_j$,D[i])* returns *true* the then the word *j* is the correct replacement of *D[i])*, else the *Bigram (D[i], Nearestneighbors)* select the most appropriate words from the list that is returned by the function *SpellCorrection(D[i],dict)* as a replacement word (this process can be pre-empted or terminated manually or may continue to *Step4*). *Update the dictionary.*

Step4: if *i<n* get the next word *D[i]* at *i=i+1*and repeat *Step2*, elseif *i=n* go to *Step5*.

Step5: Terminate algorithm

---

misspelled word. The bi-grams of each word, in list of nearest neighbors, and the misspelled word are matched to return the nearest neighbor with maximum bi-grams matched with the misspelled word. The word returned is the correct replacement for the misspelled word.

Moreover, the invalid words found in the text are fed stored in the dictionary for future to consider this word as a valid word. This invalid word is categorized and scored as that of the suggested correct dictionary word in the error correction stage. So this invalid word will be considered as variant of that suggested correction in its future occurrences. The algorithm is repeated till the last word *D[n]* in the document, where *n* is the number of words in the document.

3.1 Methods Used for Spell Checking

The spell-checking task consists of two sub tasks; spell-error detection and spell-error corrections.

3.1.1 Spell-error Detection

Spell-error detection is the process of to examine the linguistic validity of the target words in the dictionary of a language. A spell-error is detected if linguistically invalid words are found in the input document. Generally, a dictionary lookup method is used to detect spell-errors.

The dictionary lookup is a direct method, in which each word in the input document is examined/compared directly against the list of valid words, known as lexicon/dictionary. If there exist words in the input document which are not found in the lexicon, a spell-error is detected. The errored words are stored as a list with proper index, and the spell-error correction algorithm is revoked, by passing the list of invalid words and dictionary, to suggest the corrections for the misspelled words from the dictionary.

3.1.2 Spell-error Correction

The process to replace the lexically incorrect word with a valid dictionary word which is most likely to be intended. In this work we used a combination of nearest-neighbors and bigram for spell-error correction as shown in Algorithm 2.

We used edit-distance method to find out the nearest neighbors. The edit-distance works on the principle of minimum editing operations. In case of spell-error correction the edit distance of two words is the minimum number of editing operations required to replace one word by another. The editing operations can be insertion, deletion, substitution or transposition of characters. In this method a list of words which have minimum edit-distance with misspelled word is extracted from the lexicon. This list is then passed from a bi-gram matching method.

Bi-grams are substrings of every 2 adjacent characters in a string. In this method the target words, the misspelled word and words obtained from minimum edit-distance method, are breakdown into sequences of 2 characters at step of 1 character till the end of the word. The bi-grams of the misspelled word are checked against the corresponding bigrams of each word in the list. Each word in the list obtained from the edit-distance method are checked with the misspelled word to see their differece. If the difference are only vowels that word is the correct replacement. Otherwise, each element in the list is scored as the number of its bi-gram matching with the misspelled word. The word with the maximum number of bi-gram score is considered to be as the correct word and the misspelled word is replaced by this correct word.

3.2 Methods Used for Sentiment Analysis

Sentiment analysis is the process to computationally identify, polarize and categorize the opinions, attitudes and emotions articulated in a piece of text. It is data mining technique to measure the author opinions disposition leveraging the tools and concepts of natural language processing (NLP), computational linguistic and text analysis.

In this work we used the functions of Natural Language Toolkit (NLTK)[29] in Python for data preprocessing. NLTK is the leading platform to work with human language data. The toolkit is used for tokenization, part of speech (POS) tagging and building tree models.

We input the preprocessed and spell-checked textual data to the sentiment analysis module. We used an ensembled method for sentiment analysis of the text, combining Naïve Bayesian and Support Vector Machine (SVM) methods.

Naïve Bayesian[30] classifier is a simple machine learning technique based on Bayes' theorem and it assumes conditional independence of the predictors. Naïve Bayesian method basically is used for binary classification, but it can also be utilized for multi-class classification.

The SVM algorithm is a 'simple' linear classification/regression algorithm[31]. The SVM is state of the art learning algorithm proved to be effective on text classification and categorization tasks and robust on large feature spaces. It tries to form a best possible hyper-plane, a surface which best separate the target classes as optimal as possible. The SVM depends on the kernel function which may be linear or radial.

In our model, we first train a linear-kernel SVM, and multi-variable Naïve Bayes classifiers on the available training set. In the next step we passed the preprocessed, spell checked test set from both classifiers. The results of both classifiers are then combined to get the final result. If both the classifiers unanimously classify the target document as positive, neutral or negative the final decision is the same. In case of different results from both classifiers the decision is made based on the result's patterns, i.e., i) opinion is positive if one of the classifier results positive while other neutral, ii) opinion is negative if one of the classifier results negative while other neutral, and iii) opinion is neutral if one classifier results as positive and other results negative. This decision criteria is shown in Equation 1.

$$Op = \begin{cases} Op_{NB} \text{ or } Op_{SVM}, & Op_{NB} = Op_{SVM} \\ Op_{NB}, & Op_{SVM} = \neq utral \\ Op_{SVM}, & Op_{NB} = \neq utral \\ \neq utral, & Op_{NB} \neq Op_{SVM} \neq \neq utral \end{cases}$$

Where Op_{NB} and Op_{SVM} are the resultant opinions from Naïve Bayesian and SVM models respectively, and Op is the final opinion of the document.

Ⅳ. Results and Analysis

4.1 Dataset Properties

We used Reddit comments dataset (https://www.kaggle.com/cosmos98/twitter-and-reddit-sentimental-analysis-dataset) and tweets (https://www.kaggle.com/crowdflower/first-gop-debate-twitter-sentiment) containing 37,248 and 13,871 sentences along with its sentiment labels. The data set contained comparable ratio of each type of sentences, positive, negative and neutral, as shown in Figure 3, and was not skewed to a certain polarity. We divided the dataset into training and testing sets by ratio of 70% and 30% respectively. Figure 4, shows wordcloud of the cleaned training dataset.

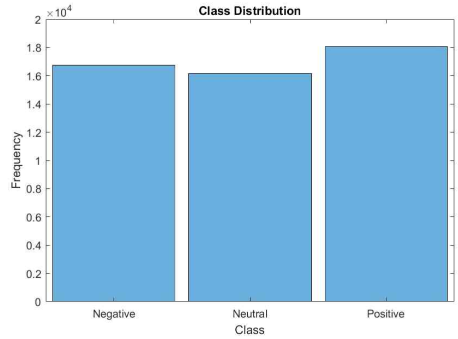

Fig. 3. Ratio of positive, neutral and negative sentences

Fig. 4. Word cloud of the training data used in simulation; the size of the words shows the word frequency.

4.2 Experimental Setup

The simulation was conducted on LG system (LG Electronics Nanjing Displays Company Ltd., China), with Corei5 , 2.3 GHz processor, 8 GHz memory, with operating system Windows 10Pro64－bit (Microsoft, United States), using MATLAB 2019b (Mathworks, Inc., United States) on LG system (LG Electronics Nanjing Displays Company Ltd., China) as simulation tool. We used MATLAB text analytic toolbox (trial version), and Statistics and Machine Learning Toolbox. We trained SVM and multi-variable Naïve Bayes classifiers for multi-class classification, such as positive, neutral, negative.

4.3 Simulation Results

From simulations, we observed that the proposed sentiment analysis method perform better then the previous methods in case of noisy data. We also observed improvement in performance of the spell-checker when we combined edit-distance with bigram.

In Table 1-3, the exemplary results, for previous sentiment analysis systems (without spell-checking) and our proposed system (with spell-checking), are shown.

In Table 1, The results are the same for both the systems as there is no spellings mistake in the sentence passed from the system.

In Table 2, a positive sentence is predicted to be negative by the previous model because the spellings of all the positive words are incorrect and the system was not able to recognize these words as the valid dictionary words and, so they are not scored, while the proposed method was able to predict the polarity of the sentence because the proposed system was able to deal with the spell-errors problem.

In Table 3, the spell of only one positive word is incorrect, therefore can not be recognized by the previous method correctly while others were recognized correctly. The overall result was not correct, and a positive sentence was predicted to be neutral. Here our proposed scheme was able to find out all the polarities correctly. Here in this case the spell-error is not like the normal typo errors. This type of errors is done deliberately to show some emotions and affinity. In our spell-checking system we incorporate the language common rules as well to deal with such types of errors. In this case there occur many consecutive "r" which is against the rule of English language. In English there are very few words where a same letter can occur more than twice consecutively. So, our system first removes the extra letters and then process the text data.

Table 1. Sentence without spell-error.

Text (string)	VADER is smart, handsome, and funny.		Overall rating of the string	
	Previous Technique	Proposed Technique	Previous	Proposed
Positive words	2	2	Positive	Positive
Negative words	1	1		

Table 2. Sentence with spell-errors in all positive words

Text (string)	VADER is smart, handsome, and funny.		Overall rating of the string	
	Previous Technique	Proposed Technique	Previous	Proposed
Positive words	0	2	Negative	Positive
Negative words	1	1		

Table 3. Sentence with spell-errors in one positive word due to some emotion or affinity.

Text (string)	VADER is smart, handsome, and funny.		Overall rating of the string	
	Previous Technique	Proposed Technique	Previous	Proposed
Positive words	1	2	Neutral	Positive
Negative words	1	1		

For performance evaluation of spell-checking we used precision (P), recall (R), and F1-scores.

$$P=\frac{TC}{TC+WC}$$

$$R=\frac{TC}{TC+NC}$$

where TC, WC, and NC are true corrections, wrong corrections, and no corrections. We consider that for any misspelled word there exists a valid dictionary replacement word therefor, we considered the no corrections NC as false negatives.

Table 4, shows the performance of our spell-checker in correction of the noisy textual data. From the simulation we found that our proposed combined method performed better than the spell-checking based on edit-distance or bigram individually.

For performance evaluation of the proposed sentiment analysis strategy we used; accuracy,

Table 4. Performance comparison of our combined spell-checker with individual methods based spell-checker in correction of the noisy textual data

Methods	Precision	Recall	F1-score
Edit distance based	75%	68%	73%
Bi-gram based	79%	77%	78%
Combined Spell checker (Edit distance + Bigram)	88%	83%	87%

average precision (AP), average recall (AR), and F1-score.

$$AP = \frac{\sum_{ci=1}^{N} P_{ci}}{N}$$

$$AR = \frac{\sum_{ci=1}^{N} R_{ci}}{N}$$

Where P_{ci} and R_{ci} are the precision and recall of class ci, respectively. In this work we consider 3 classes, which are positive, neutral, an negative, so $N=3$.

$$P_{ci} = \frac{TP_{ci}}{PR_{ci}}$$

$$R_{ci} = \frac{TP_{ci}}{T_{ci}}$$

TP_{ci}, PR_{ci}, and T_{ci} are the true predictions, total predictions, and total number of element of class ci, respectively.

Table 5, shows the performance comparison of the proposed method with previous state-of-the-art methods. From simulation we observed that the performance of the proposed model is better than the previous methods in term precision, recall, accuracy, and F1-score in case of noisy text.

From the experiments done repeatedly it was found that the average accuracy of our system in recognizing the spell-errors and correcting them was about 87%, and the average accuracy of our proposed sentiment analysis system was in the range of 83%. The overall result depends on the performance of the spell checker as well as the type of text.

Table 5. Performance comparison of the proposed method with previous state-of-the-art methods

Methods	Accuracy	AP	AR	F1- Score
Naïve Bayesian	69%	71%	66%	68%
SVM	73%	76%	71%	73%
Ensembled Method (NB+SVM)	77%	78%	75%	76%
Proposed Scheme	83%	85%	81%	82%

The performance of the proposed system is improved in terms of accuracy, however the combination of spell-checking with sentiment analysis introduced a processing time overhead. But this integration is inevitable for more accurate prediction, without losing useful information and process automation (noise detection and correction).

V. Conclusion

In this work, we proposed a new approach for enhancing the sentiment analysis of the social media textual data by leveraging the sentiment information present in the noisy text. We utilized the noisy text by correcting it instead of excluding it, to improve the performance of the sentiment analysis system. We combined an ensemble sentiment analysis model with an enhanced spell-checking technique to correct the noisy words in the text before its sentiment estimation.

Form experiments, we found that the performance of our proposed model was superior in case of the informal/abbreviated social network text data than the existing methods. Our system was able to tackle different variants of text data without losing any sentiment information. Our system correctly identified most of the alternatives of the text, and it updated the dictionary used in this work for future occurrences of these words, which minimizes the spell-checking overhead.

The performance of our model depends both on the performance of the spell-checker and sentiment analysis system. Our spell-checker and sentiment analysis model outperforms the existing methods in terms of precision, recall, accuracy, and F1-scores.

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Info. Retrieval*, vol. 2, no. 1-2, pp. 1-135,

2008.

[2] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*, pp. 415-463, 2012.

[3] J. R. Bellegarda, "Emotion analysis using latent affective folding and embedding," in *Proc. NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles, California, 2010.

[4] A. C. Boucouvalas, "Real time text-to-emotion engine for expressive internet communication," *Emerging Commun.: Stud. New Technol. and Practices in Commun.*, vol. 5, pp. 305-318, 2002.

[5] D. John, A. C. Boucouvalas, and Z. Xu, "Representing emotional momentum within expressive internet communication," in *Proc. 24th Int. Conf. Internet and Multimedia Syst. and Appl.*, pp. 183-188, 2006.

[6] V. Francisco and P. Gervvas, "Automated mark up of affective information in English texts," *Text, Speech and Dialogue*, LNCS, vol. 4188, pp. 375-382, 2006.

[7] S. M. Mohammad and T. W. Yang, "Tracking sentiment in mail: How genders differ on emotional axes," in *Proc. ACL WASSA '11*, Portland, OR, USA, 2011.

[8] H. Liu, H. Lieberman, and T. Selker, "A model of textual affect sensing using realworld knowledge," in *Proc. 8th IUI '03*, pp. 125-132, New York, NY, 2003.

[9] S. M. Mohammad, "#Emotional tweets," in *Proc. First Joint Conf. Lexical and Computational Semantics*, pp. 246-255, Montreal, Canada, 2012.

[10] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Affect analysis model: novel rule-based approach to affect sensing from text," *Natural Lang. Eng.*, vol. 17, pp. 95-135, 2011.

[11] M. Genereux and R. P. Evans, "Distinguishing affective states in weblogs," in *Proc. AAAI Spring Symp. Computational Approaches to Analysing Weblogs*, pp. 27-29, Stanford, California, 2006.

[12] R. Mihalcea and H. Liu, "A corpus-based approach to finding happiness," in *Proc. AAAI Spring Symp. Computational Approaches to Analysing Weblogs*, pp. 139-144, 2006.

[13] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. 7th LREC '10*, Valletta, Malta, 2010.

[14] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proc. LSM '11*, pp. 30-38, Portland, Oregon, 2011.

[15] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," *J. Am. Soc. Info. Sci. and Technol.*, vol. 62, no. 2, pp. 406-418, 2011.

[16] S. Brody and N. Diakopoulos, "Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs," in *Proc. EMNLP '11*, pp. 562-570, Stroudsburg, PA, USA, 2011.

[17] F. Aisopos, G. Papadakis, K. Tserpes, and T. Varvarigou, "Textual and contextual patterns for sentiment analysis over microblogs," in *Proc. WWW '12 Companion*, pp. 453-454, Lyon, France, Apr. 2012.

[18] A. Bakliwal, P. Arora, S. Madhappan, N. Kapre, M. Singh, and V. Varma, "Mining sentiments from tweets," in *Proc. 3rd WASSA '12*, pp. 11-18, Jeju, Korea, 2012.

[19] E. Martínez-Cámara, et al., "Sentiment analysis in twitter," *Natural Lang. Eng.*, vol. 20, no. 1, pp. 1-28, 2014.

[20] S. Dinakar, P. Andhale, and M. Rege, "Sentiment analysis of social network content," *2015 IEEE Int. Conf. Info. Reuse and Integration*, San Francisco, CA, USA, Aug. 2015.

[21] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *J. Artificial Intell. Res.*, vol. 50, pp. 723-762, 2014.

[22] D. Naber, "*A Rule-Based Style and Grammar Checker*," Aug. 2003.

[23] T. Naseem, "A hybrid approach for urdu

spell-checking," M.S. Thesis, National Univ. of Computer & Emerging Sciences, Nov. 2004.
[24] F. J. Damerau, "A technique for computer detection and correction of spelling errors," in *Commun. ACM*, vol. 7, no. 3, pp. 171-177, Mar. 1964.
[25] L. J. Peterson, "A note on undetected typing errors," in *Commun. ACM*, vol. 29, no. 7, pp. 633-637, Jul. 1986.
[26 B. W. W. H. L. Liang and D. G. Kourie, "Classification for selected spell checkers and correctors," School of Computing, University of South Africa, 2008.
[27] JIANG Xue-mei and XIE Fang, "Error analysis and the EFL classroom teaching," *US-China Edu. Rev.*, vol. 4, no. 9, pp. 10-14, 2007.
[28] D. J. Hovermale, "SCALE: Spelling correction adapted for learners of English," in *Proc. CALICO-08 ICALL Special Interest Group Pre-conference Workshop*, San Francisco, CA, USA, 2008.
[29] "*Natural Language Toolkit – NLTK 3.2.5 Documentation*," 2017, Nltk.Org. http://www.nltk.org/.
[30] A. McCallum, and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1, 1998.
[31] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learning Res.*, vol. 2, pp. 45-66, Nov. 2001.

칸 제브란 (Jebran Khan)

JEBRAN KHAN received his B. Sc. and M. Sc. degree in Computer Systems Engineering from the University of Engineering and Technology, Peshawar, Pakistan. Currently, he is pursuing Ph.D. degree in Electronics and Information Engineering from the Korea Aerospace University, Goyang, South Korea. His research interests include Social Networks Analysis, Modeling, Frameworks and its applications.

[ORCID:0000-0002-9971-1149]

이 성 창 (Sungchang Lee)

SUNGCHANG LEE received his BS degree from the Kyungpook National University in 1983, received M.S. degree in Electrical Engineering from KAIST (Korea Advanced Institute of Science and Technology) in 1985, and received Ph.D. degree in Electrical Engineering from Texas A&M University in 1991. From 1985 to 1987, he joined KAIST as researcher, where he worked on Image Processing and Pattern Recognition projects. He was Senior Researcher in ETRI (Electronics and Telecommunications Research Institute), Korea, from 1992 to 1993. Since 1993, he has joined the faculty at Korea Aerospace University, Goyang, Korea, where he is currently a Professor in School of Electronics, Telecommunication & Computer Engineering. During 2004-2009, he was the Director of government Project on Intelligent Smart Home Security & Automation Service Technology. In 2009, he was the vice President of IEIE (The Institute of Electronic and Information Engineers), Korea as the Director of Telecommunications Society.

[ORCID:0000-0002-4698-6018]