


An Enhanced Emotion Classification Scheme for Twits Based on Deep Learning Approach

Laman R. Sultan 

Department of Power Mechanics, Basra Technical Institute, Southern Technical University, Al-Basrah 61004, Iraq

Corresponding Author Email: laman.radi@stu.edu.iq



<https://doi.org/10.18280/ria.370512>

ABSTRACT

Received: 27 June 2023

Revised: 17 September 2023

Accepted: 24 September 2023

Available online: 31 October 2023

Keywords:

emotion analysis, wordnet library, natural language processing, TF-IDF, CNN

Given the widespread incorporation of social media into everyday existence, platforms such as Twitter have become crucial arenas for individuals to articulate their thoughts, emotions, and viewpoints. The ability to identify emotions in these facial expressions has a wide range of practical uses, including tailored marketing strategies and research on human behaviour. Nevertheless, the language used on these platforms is frequently filled with colloquialisms and vagueness, rendering the process of detecting emotions a challenging endeavour even for individuals. The difficulties in analysing emotions on Twitter are particularly noticeable because current natural language processing (NLP) techniques have limited ability to handle slang language, and earlier classifiers that rely on slang have produced unsatisfactory results. This paper presents a new model for categorising emotions in tweets that contain slang. The model combines multiple approaches and utilises the WordNET dataset. The WordNET library is used in the proposed model to create synonymous phrases for the sentences in the text. The text data is divided into segments, reduced to their root forms, then filtered to remove often used words using natural language processing (NLP) approaches. The textual properties are described by utilising Term Frequency - Inverse Document Frequency (TF-IDF) and n-gram based similarity approaches. Emotions are classified by the utilisation of a convolutional neural network (CNN). An analysis was conducted on the performance of this model based on the metrics of classification accuracy and processing speed. The findings revealed a remarkable level of accuracy in classifying emotions in slang language using the suggested model, achieving a precise categorization rate of 89.3%. This study represents a notable advancement in the field of emotion classification in social media writing that contains slang language.

1. INTRODUCTION

Emotions play a crucial role in shaping people's feelings and thoughts. Platforms for online social media like Twitter and Facebook have changed the language of communication. At present, individuals have the ability to express various types of information, including facts, opinions, emotions, and the intensity of emotions, through concise written messages on a wide range of topics [1]. The analysis of emotions expressed in social media content has garnered interest from researchers in the field of NLP technique [2]. Emotion analysis has a wide range of applications in various fields such as commerce, public health, social welfare, and policy [3]. Emotion analysis refers to the process of determining the attitude towards a specific target or topic. Attitude refers to polarity, which can be either positive or negative, or an emotional state such as joy, happy, fear, surprise, anger, or sadness [4]. The proposed study works with three kinds of emotions: sad, happy, and angry.

Numerous researchers have recently studied emotion analysis [5-7], and some techniques have been proposed for the quick analysis of English texts. The analysis of the emotional structure of a slang text is a crucial undertaking as it has the potential to reduce ambiguity to a certain degree. In order to automatically extract and classify the emotions expressed in the slang text, deep learning methods for emotion analysis in textual data are used [8-10].

One of the most challenging and yet important tasks in the text processing-based Twitter field is emotion analysis. This tool has the capability to analyse the contents present in social networks. The categorization of emotions has the potential to become a highly valuable tool within the field of social network analysis (SNA) [11]. The utilization of these techniques in social media is impeded by a significant issue, namely the presence of a diverse range of abbreviations, synonyms, and colloquialisms in user-generated content. The analysis of slang text –based emotional content in social networks poses a significant challenge due to the insufficiency of NLP resources and tools in handling slang texts [12, 13]. Also, Slang terms often have multiple meanings, and their interpretation can depend on context. This ambiguity can make it difficult for emotion classification models to accurately determine the intended emotion. Moreover, slang on Twitter is often highly context-dependent. To accurately interpret emotions, it's crucial to understand the context in which slang terms are used. This contextual understanding is vital for sentiment analysis and emotion classification tasks as the extraction of emotions solely through slang word processing is deemed impossible [14]. It is not possible to extract emotions in slang texts only through word processing. For this reason, Twitter presents unique challenges for NLP and DL models due to the brevity, informality, and idiosyncrasies of the language used. Tackling slang language on Twitter is essential for advancing NLP and DL technologies [15].

However, there is still a challenge with the specifics of the text that needs to be processed when it comes to matching texts with slang phrases [16]. The author of a given text may alter the original terminology by employing synonymous language within their sentences. In this instance, discerning this particular type of issue and conducting an analysis of the underlying emotional constructs within the text presents a greater challenge [7, 17].

In order to process and adapt texts to slang phrases and expressions, there is a problem of contradicting the details of the slang text. This means that the author of the text may change the main term by using synonyms in his / her sentences [18]. It is more difficult to distinguish this from the problem, and to understand it, one must analyse the concepts in the text. In this paper, an attempt is made to solve the mentioned problem by processing slang phrases in texts more efficiently and classifying emotions in social media using a WordNet dataset [19] consisting of idioms and slang phrases. The proposed study uses a combination of NLP [20] and WordNet datasets. In the proposed model, the WordNet dataset is used to produce a list of phrases synonymous with the sentences in the text. NLP technique has also been used for segmentation [21], stemming [22], and removing stop words from the text [23]. Then, TF-IDF [24] and n-gramme-based similarity [25] are used to describe the features of the texts. Finally, emotion classification is performed by CNN [26].

This paper is structured in the following manner: The second section of the paper describes the methodology employed in the study being proposed. The findings that resulted from the experiment have been delineated in Section 3. Prior to the conclusion of the work in Section 4.

2. PRELIMINARIES

To achieve the goal of analysing and categorising emotions of slang text in tweets, the proposed model's details will be explained in this part. First, we explained the flowchart of the proposed study in the section 2.1. Furthermore, the methods employed for preparing the slang text and categorising the textual features were thoroughly explained. Third point. In this section, the process of generating synonym phrases using WordNet dataset has also been listed. Lately, there has been a listing of the processes involving the extraction of TF-IDF features and the classification of emotions.

2.1 Proposed methodology

One of the most difficult tasks in NLP is emotion analysis. This will be much more difficult for people to accomplish due to a lack of NLP resources and tools. It can't just use word processing to figure out how people feel about a text. As a result, it appears that using slang words and phrases in the text is a good way to achieve this goal. However, in order to process and adapt texts to slang phrases and expressions, we must change textual details, which can be confusing. This means that the text's author can change the main word by substituting a similar word. This is more difficult to distinguish from the problem, and understanding it requires a close examination of the ideas in the text. In this section of the paper, an attempt is made to solve the mentioned problem, and a new method for analyzing emotions in English texts is demonstrated using a database of idioms and slang phrases. A similarity criterion based on n-grams is also used to match the

phrases in the text with the database of idioms and slang phrases. We'll go over this in greater detail later in the section. The general proposed flowchart is depicted in Figure 1.

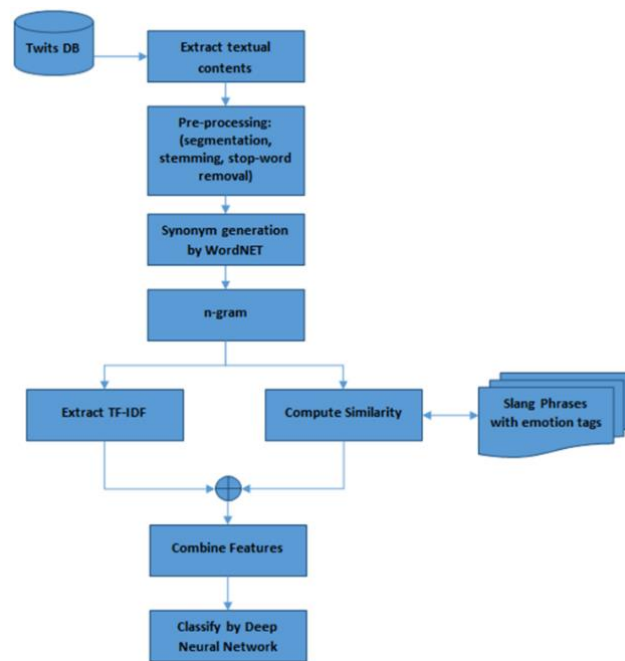


Figure 1. General proposed flowchart

According to Figure 1, the proposed method uses a database containing slang phrases to better analyze the content of users' messages on social networks. Each slang phrase in this database is tagged with an emotional class, e.g., "sad", "happy" "angry", etc. In this method, the content of each message is first preprocessed using segmentation, stemming, and stop-word removal operations; then, text fragments are extracted using the n-gram technique. In the next step, the similarity of each piece of text with the slang expressions in the database is computed, and this similarity is described as a numerical feature. Besides that, tweet contents will then be extracted using the TF-IDF technique and the combination of the extracted features will be processed by a CNN. The following sections will illustrate the stages shown in Figure 1.

2.1.1 Pre-processing stage

The purpose of pre-processing stage, is to remove unnecessary information and prepare data for correct processing in the next steps of the proposed method. The pre-processing process in the proposed model begins with text segmentation. Text segmentation is the process of decomposing a text into its constituent parts, such as paragraphs, sentences, and words. Given that the smallest acceptable unit for extracting emotions in a text document is a sentence; in the proposed model, sentences are considered as the purpose of segmentation (see Eq. (1)). To do this, each text document is first normalized. Text normalization [27] refers to the correction of blank spaces and signs in the text (see Eq. (2)). During this process, the consecutive characters of blank spaces, new line, and symbols in the text are refined. In the proposed model, each textual content is processed using the OpenNLP tool [28] and each textual content is described as an array of words with their labels. In the list of words generated, the end of the sentence is marked using the <sentenceSplitter> tag. Therefore, this tag will be used to segment input texts.

$$D = \sum_{k=1}^M \frac{\text{sum of ranks within segment } k}{\text{area within segment } k} \quad (1)$$

When k = the text needs to be segmented.

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

When the range of value (x) is (0, 1).

After decomposing the text to its sentences, the stemming and stop word removal operations are performed on the resulting text. To do this, we first identify the stop words in each piece of text and delete them. Stop words are words such as: “be”, “from”, “as”, etc., the presence of which in the text will not reflect any key concept [29]. Therefore, deleting these words in the sentences can improve the process of feature extraction.

After removing stop words from each piece of text, we decompose the resulting text into its constituent words and perform the stemming operation. Because, using stem words will limit the scope of words. Accordingly, for each word of the text, we ignore suffixes and prefixes. It should be noted that in the proposed method, we have used the porter algorithm to perform stemming operations. The porter algorithm performs word stemming operations through the following steps:

1. Remove the plural (s-es) and time relevant suffixes such as “ed” and “ing” from each word.
2. Convert dual compound extensions such as “-ization” to single extensions such as “-ize” and “-ation”.
3. Delete extensions such as “ic”, “full”, “ness”, etc., which are obtained through the second step.
4. Remove suffixes such as “-ant”, “-able” and “-ence” from the end of words.
5. Delete “-e” at the end of words and convert “-ll” to “-l” if the remaining length of the word is more than one character.

After stemming, we delete the stop words in the resulting text again. Because words like “being” have suffixes but are stop words. This feature prevents the removal of stop words from being successful at first. Therefore, after stemming the words, the root of the word (be) will remain, and re-deleting stop words will lead to the successful deletion of the word. It should be noted that pre-processing step is performed on both tweets and slang phrases databases.

2.1.2 Generating synonym phrases by WordNET

WordNET dataset [19] is an English knowledge database that contains information about words and concepts, their syntactic information and semantic relations between them. The first version of WordNET contains more than 17,000 word entries from the categories of noun, verb and adjective [30]. The second version of WordNET includes more than 30,000 word entries from the categories of noun, verb, adjective and adverb [31]. In addition to the intra-category relationships in English WordNET (Version 2.1), the five inter-category relationships also connect concepts, and in addition to the features intended for words, syntactic, phonetic, and phonetic features have been added to structure of words [32]. In this study, the second version of WordNET has been used for conceptual processing of English texts.

To generate phrases synonymous with each sentence in the input text, the array of words is obtained first, and then converted into a list of unique words in the text. This list of

words is used as the input of the WordNET library. If we consider the list of unique words in a text as $W = \{w_1, w_2, \dots, w_n\}$, WordNET software for each word w_i will produce a list of synonyms such as $S_{w_i} = \{s_1, s_2, \dots, s_j\}$. Therefore, for a text document that has unique words W , the output of WordNET will be $S_W = \{S_{w_1}, S_{w_2}, \dots, S_{w_n}\}$. In this set, S_{w_i} is a list of words synonymous with the word w_i in the set W .

After creating the synonyms set S_W , each sentence in the query text is processed and using the synonyms set, all the combinations with the same sense with the current sentence will be generated. An example of the process of splitting text and generating synonyms of sentences is shown in Figure 2.

Query Text:					
Text mining is a method of processing textual information . In this way, very useful information can be extracted automatically from the texts.					
Normalized Sentence:					
Text mining is a method of processing textual information.					
Unique Words:					
Text	mining	method	processing	textual	information.
Synonym Words					
Text	mining	method	processing	textual	information.
“	excavation	procedure	handling	“	info
“	searching	strategy	dealing with	“	data
				
Synonym Sentences:					
Text excavation is a procedure of processing textual information					
Text mining is a strategy of dealing with textual information					
Text mining is a strategy of handling textual info					
And so on...					

Figure 2. An example of generating synonym phrases in proposed method

According to Figure 2, the input text is first decomposed into its constituent sentences. Each normalized sentence is then broken down into its unique words. Then a list of synonyms is generated for each word, and by replacing synonyms with the main words in the sentence, all synonyms will be generated with the input text.

In the proposed study, the utilized database consists of a collection of slang terms and English proverbs. This database consists of the following components:

- 40 proverbs in English
- 33 common slang terms in English

In total, the database contains 73 slang terms and proverbs. Each of these samples was rated positively or negatively by an expert. And the intensity of emotion (positive or negative) is set as a number between 0 and 100. It should be noted that MS. Excel software (2016) has been used to store database information. A summary of the properties of the collected database is shown in Table 1.

Table 1. Summary of the characteristics of the collected database

Parameter	Value
Number of English Proverbs	40
Number of Common Slang Terms	33
Number of Negative Proverbs	19
Number of Positive Proverbs	21
Number of Negative Slang Terms	15
Number of Positive Slang Terms	18
Average Number of Words in English Proverbs	10.48
Average Number of Words in Common Slang Terms	3.45

In order to evaluate the efficiency of the proposed model, a collection of 73 opinions of Twitter Sentiment Analysis database about US election has been used. The positive or negative feelings in each of these comments are known, and the purpose of the proposed model is to determine whether users' emotions are positive or negative by processing the texts of their comments.

2.1.3 Generating n-grams and computing similarity

According to the process described in the previous section, the query message D with n sentences will be described as a set like: $Q_D = \{\{Q_{(P_1)}\}, \{Q_{(P_2)}\}, \dots, \{Q_{(P_n)}\}\}$. In this set, $Q_{(P_i)}$ is a set of synonymous sentences with P_i . Each member of the set Q corresponds to a synonym phrase in message D. In order to identify the similarities, in the proposed model, each sentence in the input text is compared with idioms and slang phrases of the database, and a decision will be made about the similarity of each pair of sentences. This is done in the proposed model using n-gram method.

In the proposed model, the similarity between each sentence of message D and the idioms and slang phrases of database is calculated by producing 3-gram subsets of text document sentences. For example, 3-grams set of the sentence: "NLP can recognize emotions in texts." will be as follows:

{“NLP can recognize”, “can recognize emotions”, “recognize emotions in”, “emotions in texts”}.

In this paper, for each sentence (and its synonyms) in the query message D, the members of set Q_D are compared one by one with the database phrases, and the similarity percentage of each pair is calculated using n-grams. In order to calculate the similarity of the two sentences using the n-gram solution, the jacquard criterion [33] is used in the proposed model. Suppose we want to calculate the similarity of two sentences like A and B. To do this, we first extract all 3-grams of sentences A and B. Assume that the sets N_A and N_B are all possible 3-grams of A and B, respectively. In this case, the similarity of the two sentences is calculated using the following equation:

$$Similarity_{A,B} = 100 \times \frac{|N_A \cap N_B|}{|N_A \cup N_B|} \quad (3)$$

In the above relation, $|N_A \cap N_B|$ refers to the number of common 3-grams between N_A and N_B , and $|N_A \cup N_B|$ is the number of members resulted by the union of these two sets.

After calculating the similarity between each slang phrase of the database and the synonymous sentences of the input text, finally the most similarity obtained between the synonymous sentences with the current sentence and the database sentences is considered as the final similarity percentage for the current sentence in the input text. After calculating the similarity percentage for each sentence, if the calculated similarity for the current sentence is more than 65%, the current sentence is considered in accordance with the slang phrase. Then, emotional classes which are tagged with according slang phrase will be set as logical “TRUE” in the resulting vector.

2.1.4 Extracting TF-IDF features

In this step of the proposed model, the textual content of each message will be described as a vector. On the other hand, it is necessary to describe the content of all database texts as vectors of the same length. For this reason, the proposed model uses the TD-IDF standard. To do this, we will first generate a list of unique words for all the texts in the database. This list includes the entire set of database root words and is used to

describe each message. Equations number 4 and 5 expresses the used TF and IDF to determine the frequency with which a text appears in a document.

$$Term\ Frequency = \frac{no\ of\ times\ term\ (t)\ appears\ in\ the\ document}{Total\ no.\ of\ terms\ in\ the\ document} \quad (4)$$

$$Inverse\ document\ frequency = \log \frac{Total\ no.\ of\ documents}{no\ of\ documents\ with\ term\ (t)} \quad (5)$$

Suppose that by doing this, a vector such as $F = \{w_1, \dots, w_n\}$. Is generated for the database. Each text can be described by giving weight to a feature vector such as $W = \{t_1, \dots, t_n\}$. In this vector, t_i determines the weight of the word w_i in vector F. After extracting the list of unique words in the text, the TF-IDF criterion is used to describe the lexical content of each message. In this step, the content of each message is tracked based on the unique words in it. Then each word is weighed using the TF-IDF criterion. The following relation is used to weigh a word such as w:

$$TFIDF_w = \frac{T_w}{T_D} \times \left(\frac{N}{N_w}\right) \quad (6)$$

In the above relation T_w is the number of repetitions of the word w in the current message, T_D is the total number of words in the current message and N is the total number of messages in the database. N_w also specifies the number of messages in the database that contain the word w. In the proposed method, after calculating the weight of each word, the amount of weight obtained is stored in the W features. By doing this, a feature matrix will be obtained whose number of rows will be equal to the number of messages and the number of columns will be equal to the length of the feature vector. Therefore, each row of the matrix resulting from this step will describe a message.

2.1.5 Emotion classification using CNN

In the last step of the proposed model, similarity features (section 2.1.3) and TF-IDF features (section 2.1.4) are combined to construct the feature set. Then these features are fed to a CNN technique for slang text classification of emotion through them. The structure of the CNN used in the proposed model is shown in Figure 3. As shown in Figure 3, the CNN used in the proposed model consists of 14 consecutive layers. The components of this CNN are: an input layer, a classification layer, four ReLU layers, two convolution layers, two MaxPool layers, three fully connected layers, and one SoftMax layer.

Passing the properties of each sample from the input layer to the fully connected FC3 layer will result in a hierarchical abstraction of the data properties and reduce the dimensions of the features to 128 properties. Each layer of ReLU, after the convolution layers in this neural network, is acting as an activator function. Each two-dimensional layer of MaxPool will also be responsible for transferring data from each sample to the next phase of processing. The three fully connected layers defined at the end of the deep neural network structure will also be used to describe sample's properties and compress it. Eventually; The classification layer will be responsible for determining the output of the CNN, indicating the emotion class of the texts.

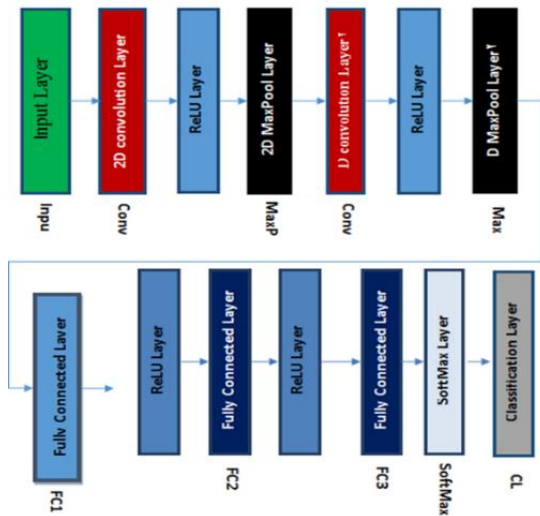


Figure 3. Structure of the CNN used in the proposed method

The primary justification for using CNN in the analysis and classification of slang text is that it can extract features from a wide range of data and take into account the relationships between those characteristics. The proposed model has the potential to significantly improve the accuracy of text analysis and classification. In the field of NLP, it is possible to extract features from text data individually and analyse their relationships. However, it is important to note that without considering the context or the entire sentence, there is a risk of misinterpreting the sentiment. Currently, CNN is considered one of the most effective methods for performing sentiment text classification. CNN utilises a convolutional layer to extract information from larger pieces of text. In our work, we focused on sentiment analysis using a convolutional neural network. We designed a simple CNN model and tested it on a benchmark dataset. The results demonstrated that our model achieved higher accuracy in twitter sentiment classification compared to traditional methods such as SVM, K Nearest Neighbours, and J48 decision tree.

3. RESULTS AND DISCUSSION

3.1 Implementation of the proposed method

The methodology suggested in this study has been executed utilising a PC with Intel Core i5-2400, 3.1GHz, 6MB L2, L3 Cache, 4 GB PC3-10600 (DDR3-1333) RAM, 1 TB (7200 RPM) SATA-3G Hard Disk Microsoft Windows® 7 Home Premium Edition (64-bit). The experiments were conducted using Several Matlab libraries.

The OpenNLP library has been employed for the normalization and part-of-speech tagging of English sentences. The library encompasses a range of tools, such as part-of-speech tagging, text parsing, chunking, stemming, and converting informal language to formal language, among others. In addition to the tasks mentioned, we can also train and evaluate the proposed model for any of these tasks.

As previously stated, a set of text comments containing 75 opinions of Twitter users about the US election was used to evaluate the proposed method. Each of these comments will be treated as a test sample. Each comment also contains one of the positive or negative emotions that an expert has already determined. The target variable is the type of emotion

expressed in each comment. The proposed method's goal is to predict this target variable based on the comment text. The rest of this section will refer to texts that elicit positive emotions as "positive class samples" and texts that elicit negative emotions as "negative class samples." 35 of the samples evaluated fall into the negative category, while 40 falls into the positive category.

The proposed model's performance was evaluated using the cross-validation technique. As a result, we divide the database into five parts (without shared data), each of which contains 20% of the database samples. The training and testing operations are then repeated five times, with four parts (80% of samples) considered labelled and used to train the CNN classifier in the proposed method. The remaining 20% of samples are considered unlabeled and used to test the proposed method's performance in correct emotion classification. Following that, we will describe the results obtained from implementing this scenario in the proposed approach.

The results of the correct emotion classification by the proposed method have been showed in Figure 4, compared with the accuracy of other classification algorithms. In these experiments, the performance of the proposed method is compared with the other three learning algorithms. These algorithms are:

- a) K Nearest Neighbors.
- b) J48 decision tree.
- c) Support Vector Machine (SVM).

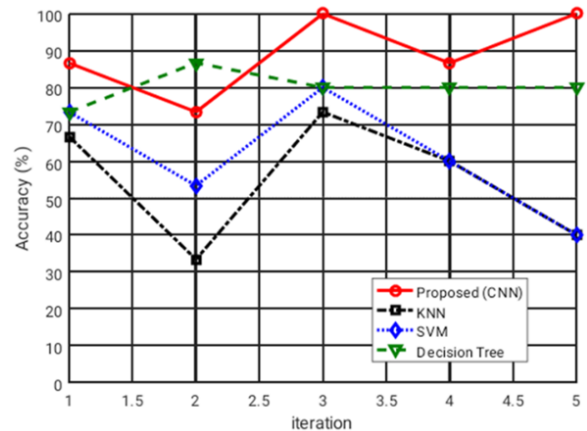


Figure 4. Accuracy of classification algorithms in emotion classification during 5 iterations

Figure 4 shows that the proposed method using CNN classifier has higher average accuracy than the compared algorithms. The confusion matrix resulting from the classification of emotions in the test samples by the proposed model has been illustrated in Figure 5. According to this figure, out of 35 samples that belong to negative emotion class, 34 samples were correctly classified by the proposed method and also 1 sample was incorrectly classified in the category of texts with positive emotions (positive class). Also, out of 40 samples that belong to the positive class, 33 samples were correctly classified in the category of comments with positive emotions and the remaining 7 samples were misclassified. These results show that using the proposed method, emotions in English texts can be correctly classified with an average accuracy of 89.3%.

As shown in Figure 6, if the WordNET tool is not used to generate synonymous sentences, the accuracy of correctly emotion classification from texts will be significantly reduced,

and emotions in only 70.7% of test samples will be classified correctly. This is because people often use synonyms of proverbs and slang terms in their expressions. As a result, it will not be possible to fully match the database terms with the query text. The WordNET tool used in the proposed method solves this problem by generating synonymous expressions for the query text sentences. That is why the use of this tool in the proposed method has increased the accuracy by 18.6%.

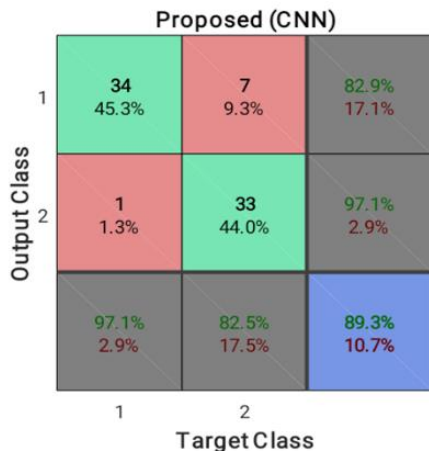


Figure 5. Confusion matrix of emotion classification by the proposed model

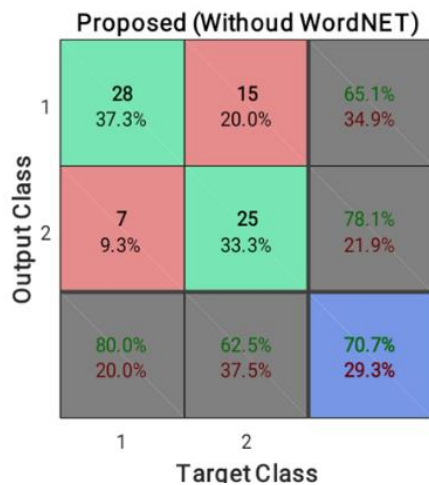


Figure 6. Confusion matrix of emotion classification by the proposed model (without WordNET)

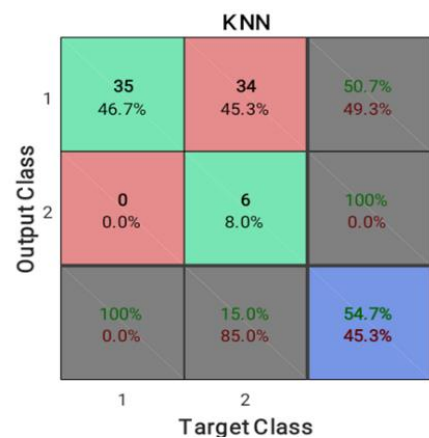


Figure 7. Confusion matrix of emotion classification by KNN

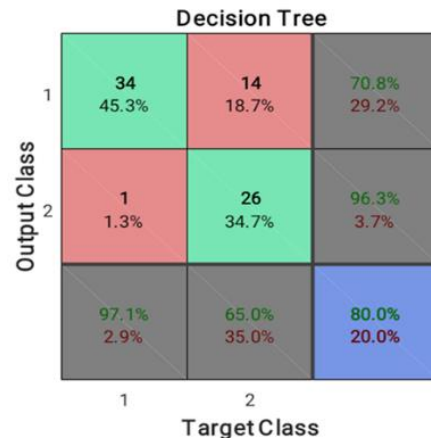


Figure 8. Confusion matrix of emotion classification by J48 decision tree

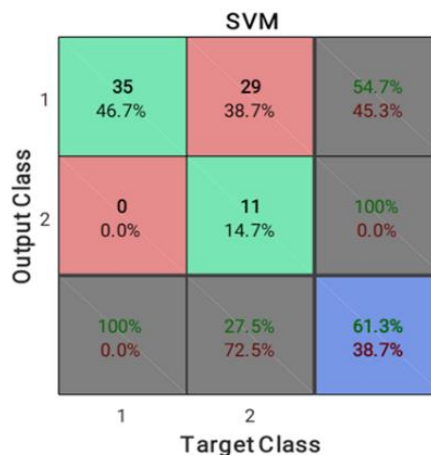


Figure 9. Confusion matrix of emotion classification by SVM

Figure 6 shows the accuracy of correct emotion classification through texts in the case that the WordNET tool is not used.

As described in section 2.1.5, the proposed method uses CNN neural network to classify database features. In order to evaluate the performance of the CNN neural network in the classification of emotions, we will compare its performance with other classification models. In this section, the accuracy of the proposed method in emotion classification will be compared with K-Nearest Neighbors, J48 decision tree and Support Vector Machine classification models. Figures 7, 8 and 9 show the confusion matrices of the KNN, J48 decision tree, and SVM classifiers, respectively.

As shown in Figures 7, 8 and 9, the accuracy of the proposed method will be higher than all the compared cases. Therefore, it can be concluded that by using the combination of CNN and WordNET, higher classification accuracy can be achieved than other compared algorithms. Based on experiments performed, if the KNN classifier is used to classify and recognize emotions in texts; the accuracy rate will be 54.7%. Also, if the J48 decision tree is used to classify emotions in the evaluated texts; The accuracy will be 80.0%. Finally, using the support vector machine, the accuracy of emotion classification in the evaluated texts will be 61.3%. This is while using the proposed method, the average accuracy of 89.3% can be achieved. These results show that the combination of WordNET tool and CNN neural network in the proposed

method will increase the accuracy, compared to these algorithms, by at least 9.3%.

The experiment results of the proposed algorithm for correct emotion classification of test samples have been illustrated in Table 1. In this table, the sensitivity and specificity criteria are compared. The sensitivity criterion is used to measure the ratio of the total positive samples that have been correctly classified. This criterion is calculated as follows:

$$Sensitivity = 100 \times \frac{TP}{TP + FN} \quad (7)$$

In the above equation, TP is the number of positive samples that have been correctly classified (the value located in the second row and column of the confusion matrix) and FN refers to the number of positive samples that have been classified in

negative class. (The value in the first row and the second column of the confusion matrix). Specificity criteria is used to measure correctly classified negative samples. This criterion is calculated as follows:

$$Specificity = 100 \times \frac{TN}{TN + FP} \quad (8)$$

In the above equation, TN is the number of negative samples which have been correctly classified (the value located in the first row and column of the confusion matrix) and FP is the number of negative samples that are incorrectly classified as positive samples (the value in the second row and the first column of the confusion matrix). Also, in Table 2, the results of the proposed method have been compared with other cases.

Table 2. Efficiency of the proposed method in classification of emotions

Method	Accuracy	Sensitivity	Specificity	FN	FP	TN	TP
Proposed Method	89.3	82.5	97.1	7	1	34	33
Proposed Method (Without WordNET)	70.7	62.5	80.0	15	7	28	25
KNN	54.7	15.0	100.0	34	0	35	6
Decision Tree	80.0	65.0	97.1	14	1	34	26
SVM	61.3	27.5	100.0	29	0	35	11

As the results in Table 1, the proposed method has better performance than other compared conditions both in terms of classification accuracy and sensitivity criteria. The average processing time of each sample in various conditions, have been compared in Figure 10.

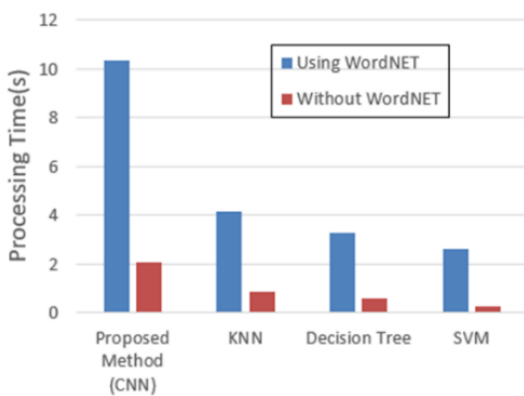


Figure 10. Average processing time of each sample in various conditions

According to Figure 10, it can be concluded that using WordNET, the average processing time of each file is 10.33 seconds. Whereas, without WordNET, the average processing time of each sample is 2.04 seconds. As mentioned, the reason for increasing processing time is to spend more time processing synonymous sentences. Considering that the use of WordNET increases the accuracy of classification by 30%; therefore, the added time will be negligible.

4. CONCLUSION

In this paper, a new model for emotion classification of slang text in tweets using deep learning techniques was proposed. In the proposed method, a combination of NLP techniques, WordNET library and CNN technique is used.

Also, in order to match the phrases in the text with the database of idioms and slang phrases, n-gram based similarity criterion is used. The suggested methodology is a good effort that aims to annotate slang texts' emotional content more quickly and inexpensively. Promoting the expansion of corpora for sentiment analysis in the field of natural language processing could be a good starting point. The results obtained by the present research can be useful for social network developers and researchers in SNA or text mining areas. The proposed model can be used experimentally in the following organizations: Social networks, online opinion expression websites, and online markets. The results show that the proposed model can analyze and classify emotions in texts with high accuracy and speed. In future works, we can try to improve the efficiency of the proposed model by using other feature extraction schemes.

REFERENCES

- [1] Liu, X., Zhou, G., Kong, M., Yin, Z., Li, X., Yin, L., Zheng, W. (2023). Developing multi-labelled corpus of twitter short texts: A semi-automatic method. *Systems*, 11(8): 390. <https://doi.org/10.3390/systems11080390>
- [2] Zhang, T., Yang, K., Ji, S., Ananiadou, S. (2023). Emotion fusion for mental illness detection from social media: A survey. *Information Fusion*, 92: 231-246. <https://doi.org/10.1016/j.inffus.2022.11.031>
- [3] Hou, Q., Han, M., Cai, Z. (2020). Survey on data analysis in social media: A practical application aspect. *Big Data Mining and Analytics*, 3(4): 259-279. <https://doi.org/10.26599/BDMA.2020.9020006>
- [4] Wang, Z., Ho, S.B., Cambria, E. (2020). A review of emotion sensing: Categorization models and algorithms. *Multimedia Tools and Applications*, 79: 35553-35582. <https://doi.org/10.1007/s11042-019-08328-z>
- [5] Goularas, D., Kamis, S. (2019). Evaluation of deep learning techniques in sentiment analysis from Twitter data. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-*

- ML) Istanbul, Turkey, pp. 12-17. <https://doi.org/10.1109/Deep-ML.2019.00011>
- [6] Xue, J., Chen, J., Hu, R., Chen, C., Zheng, C., Su, Y., Zhu, T. (2020). Twitter discussions and emotions about the COVID-19 pandemic: Machine learning approach. *Journal of medical Internet research*, 22(11): e20550. <https://doi.org/10.2196/20550>
- [7] Dini, L., Bittar, A. (2016). Emotion analysis on twitter: The hidden challenge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia, pp. 3953-3958. <https://aclanthology.org/L16-1624>.
- [8] Hung, L.P., Alias, S. (2023). Beyond sentiment analysis: A review of recent trends in text based sentiment analysis and emotion detection. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 27(1): 84-95. <https://doi.org/10.20965/jaciii.2023.p0084>
- [9] Shelke, N., Chaudhury, S., Chakrabarti, S., Bangare, S. L., Yogapriya, G., Pandey, P. (2022). An efficient way of text-based emotion analysis from social media using LRA-DNN. *Neuroscience Informatics*, 2(3): 100048. <https://doi.org/10.1016/j.neuri.2022.100048>
- [10] Guo, J. (2022). Deep learning approach to text analysis for human emotion detection from big data. *Journal of Intelligent Systems*, 31(1): 113-126. <https://doi.org/10.1515/jisys-2022-0001>
- [11] Babu, N.V., Kanaga, E.G.M. (2022). Sentiment analysis in social media data for depression detection using artificial intelligence: A review. *SN Computer Science*, 3: 1-20. <https://doi.org/10.1007/s42979-021-00958-1>
- [12] Krommyda, M., Rigos, A., Bouklas, K., Amditis, A. (2021). An experimental analysis of data annotation methodologies for emotion detection in short text posted on social media. In *Informatics*, 8(1): 19. <https://doi.org/10.3390/informatics8010019>
- [13] Patton, D.U., Frey, W.R., McGregor, K.A., Lee, F.T., McKeown, K., Moss, E. (2020). Contextual analysis of social media: The promise and challenge of eliciting context in social media posts with natural language processing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, New York, USA, pp. 337-342. <https://doi.org/10.1145/3375627.3375841>
- [14] Gandhi, A., Adhvaryu, K., Poria, S., Cambria, E., Hussain, A. (2023). Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. *Information Fusion*, 91: 424-444. <https://doi.org/10.1016/j.inffus.2022.09.025>
- [15] Naseem, U., Razzak, I., Eklund, P. W. (2021). A survey of pre-processing techniques to improve short-text quality: A case study on hate speech detection on twitter. *Multimedia Tools and Applications*, 80: 35239-35266. <https://doi.org/10.1007/s11042-020-10082-6>
- [16] Katz, D.M., Hartung, D., Gerlach, L., Jana, A., Bommarito II, M.J. (2023). Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039*.
- [17] Torregrosa, J., Bello-Orgaz, G., Martínez-Cámara, E., Ser, J.D., Camacho, D. (2023). A survey on extremism analysis using natural language processing: Definitions, literature review, trends and challenges. *Journal of Ambient Intelligence and Humanized Computing*, 14(8): 9869-9905.
- [18] Ahanin, Z., Ismail, M.A., Singh, N.S.S., AL-Ashmori, A. (2023). Hybrid feature extraction for multi-label emotion classification in English text messages. *Sustainability*, 15(16): 12539. <https://doi.org/10.3390/su151612539>
- [19] Beckwith, R., Fellbaum, C., Gross, D., Miller, G.A. (1991). WordNet: A lexical database organized on psycholinguistic principles. *Lexical acquisition: Exploiting on-line resources to build a lexicon*, 211-232.
- [20] Kang, Y., Cai, Z., Tan, C.W., Huang, Q., Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, 7(2): 139-172. <https://doi.org/10.1080/23270012.2020.1756939>
- [21] Azeraf, E., Monfrini, E., Vignon, E., Pieczynski, W. (2021). Highly fast text segmentation with pairwise markov chains. In *2020 6th IEEE Congress on Information Science and Technology (CiSt) Agadir - Essaouira, Morocco*, pp. 361-366. <https://doi.org/10.1109/CiSt49399.2021.9357304>
- [22] Singh, J., Gupta, V. (2016). Text stemming: Approaches, applications, and challenges. *ACM Computing Surveys (CSUR)*, 49(3): 1-46. <https://doi.org/10.1145/2975608>
- [23] Smelyakov, K., Karachevtsev, D., Kulemza, D., Samoilenko, Y., Patlan, O., Chupryna, A. (2020). Effectiveness of preprocessing algorithms for natural language processing applications. In *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T) Kharkiv, Ukraine*, pp. 187-191. <https://doi.org/10.1109/PICST51311.2020.9467919>
- [24] Singh, S., Kumar, K., Kumar, B. (2022). Sentiment analysis of Twitter data using TF-IDF and machine learning techniques. In *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) Faridabad, India*, 1: 252-255. <https://doi.org/10.1109/COM-IT-CON54601.2022.9850477>
- [25] Chahidi, H., Omara, H., Lazaar, M., Al Achhab, M. (2019). Impact of neural network architectures on arabic sentiment analysis. In *Proceedings of the 4th International Conference on Big Data and Internet of Things, Rabat, Morocco*, pp. 1-6. <https://doi.org/10.1145/3372938.3372950>
- [26] Liao, S., Wang, J., Yu, R., Sato, K., Cheng, Z. (2017). CNN for situations understanding based on sentiment analysis of Twitter data. *Procedia computer science*, 111: 376-381. <https://doi.org/10.1016/j.procs.2017.06.037>
- [27] Taha, M.S., Rahem, M.S.M., Hashim, M.M., Khalid, H.N. (2022). High payload image steganography scheme with minimum distortion based on distinction grade value method. *Multimedia Tools and Applications*, 81(18): 25913-25946.
- [28] Jayanthi, S.M., Nerella, K., Chandu, K.R., Black, A.W. (2021). Codemixednlp: An extensible and open nlp toolkit for code-mixing. *arXiv preprint arXiv:2106.06004*.
- [29] Neelakandan, S., Paulraj, D., Ezhumalai, P., Prakash, M. (2022). A deep learning modified neural network (DLMNN) based proficient sentiment analysis technique on Twitter data. *Journal of Experimental & Theoretical Artificial Intelligence*. <https://doi.org/10.1080/0952813X.2022.2093405>
- [30] Kilgarriff, A. (2000). Wordnet: An electronic lexical database. *JSTOR*. <https://doi.org/10.2307/417141>
- [31] Finlayson, M. (2014). Java libraries for accessing the princeton wordnet: Comparison and evaluation. In

Proceedings of the Seventh Global Wordnet Conference, 78-85.

[32] Zafar, A., Mahmood, A., Shams, S., Hussain, S. (2014). Structural analysis of linking urdu wordnet to pwn 2.1. In the Proceedings of Conference on Language and Technology.

[33] Osochkin, A., Piotrowska, X., Fomin, V. (2021). Automatic identification of authors' stylistics and gender on the basis of the corpus of Russian fiction using extended set-theoretic model with collocation extraction. *Glottometrics*, 50: 76-89.