INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING & COMPUTER SYSTEMS (IJSECS)
ISSN: 2289-8522 e-ISSN: 2180-0650
VOL. 9, ISSUE 2, 93 – 104
DOI: https://doi.org/10.15282/ijsecs.9.2.2023.3.0114

ORIGINAL ARTICLE

Sentiment Classification of Tweets with Explicit Word Negations and Emoji Using Deep Learning

Mdurvwa Usiju Ijairi[1], Mohammed Abdullahi[1,*], Ibrahim Hayatu Hassan[1,2]

[1]Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria
[2]Institute for Agricultural, Ahmadu Bello University, Zaria, Nigeria.

ABSTRACT – The widespread use of social media platforms such as Twitter, Instagram, Facebook, and LinkedIn have had a huge impact on daily human interactions and decision-making. Owing to Twitter's widespread acceptance, users can express their opinions/sentiments on nearly any issue, ranging from public opinion, a product/service, to even a specific group of people. Sharing these opinions/sentiments results in a massive production of user content known as tweets, which can be assessed to generate new knowledge. Corporate insights, government policy formation, decision-making, and brand identity monitoring all benefit from analyzing the opinions/sentiments expressed in these tweets. Even though several techniques have been created to analyze user sentiments from tweets, social media engagements include negation words and emoji elements that, if not properly pre-processed, would result in misclassification. The majority of available pre-processing techniques rely on clean data and machine learning algorithms to annotate sentiment in unlabeled texts. In this study, we propose a text pre-processing approach that takes into consideration negation words and emoji characteristics in text data by translating these features into single contextual words in tweets to minimize context loss. The proposed preprocessor was evaluated on benchmark Twitter datasets using four deep learning algorithms: Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Artificial Neural Network (ANN). The results showed that LSTM performed better than the approaches already discussed in the literature, with an accuracy of 96.36%, 88.41%, and 95.39%. The findings also suggest that pre-processing information like emoji and explicit word negations aids in the preservation of sentimental information. This appears to be the first study to classify sentiments in tweets while accounting for both explicit word negation conversion and emoji translation.

ARTICLE HISTORY
Received: 23 February 2023
Revised: 20 April 2023
Accepted: 27 June 2023
Published: 20 July 2023

KEYWORDS
Sentiment analysis
Long-short-term memory
Word embedding
Emoji

INTRODUCTION

Various social media platforms have contributed to the transformation of the day-to-day forms of human interactions and decision-making over the years. As a result, business organizations, government agencies, academic institutions, and consistent individuals have taken advantage of communicating to the public using this medium to reach out to different parts of the world. This unparalleled surge in the reception and penetration of social networking platforms such as Facebook, Twitter, and LinkedIn, has changed the pattern of online communications. In today's web, interactions can be taken out as a public discussion or topic among a wide range of individuals via public posts or discussion forums, so users can engage and share different ideas or opinions on a particular subject. As the interactions continue, large volumes of data are continuously generated because of users' exchanges and engagements online. According to Hassonah *et al.* [1], this large volume of data represents the sentiments/opinions or reviews of the users that are available for analysis. Analyzing the generated text data is known as sentiment analysis (SA) or opinion mining which involves extracting the sentiments/opinions about a particular topic from user-generated text data. Furthermore, the immense amount of text data available on social media platforms contains opinions or reviews on different subjects. Identifying sentiments can prove to be useful on different levels such as for business organizations/stakeholders in monitoring brand reputation and product sentiment in customer feedback and understanding customer needs. Governments can observe and describe the public opinion of their citizens on various policies. However, due to the prevalence of newer slang phrases, spelling errors, sarcasm, memes, and emojis, processing user-generated text data from social media to get useful insights have proven difficult [2, 3].

Recent research highlights the challenges in accurately analyzing sentiments in short-form text containing negation and emoji. Negation, in particular, can significantly impact the meaning of a phrase or word, and determining the sequence of words affected by negation is crucial for reliable sentiment analysis [4]. As such, appropriate processing techniques are necessary to avoid misclassification when evaluating sentiments on a document level. To ensure accurate extraction of detailed information from users' opinions, it is essential to handle the inclusive diversity of verbalizations, including

*CORRESPONDING AUTHOR |M. Abdullahi | abdullahilwafu@abu.edu.ng

elements such as emoji, short-form text expansion [5], and word negations. Negation words, in particular, have a direct impact on the meaning of the related word; for instance, "interesting" is a positive view, whereas "not interesting" is a negative one. However, during data cleaning, negation words are often eliminated, making them unclear to the classifier, and thus losing their contextual meaning. Therefore, it is crucial to handle negation words appropriately to obtain more meaningful documents for sentiment analysis. Most present approaches overlook this semantic combination, leading to the misclassification of sentiment [6]. According to Singh and Paul [4], identifying the scope of negations improves the classification accuracy of sentiment analysis.

Furthermore, a new set of emoticons known as emojis have been dominating social media interactions. An emoji can be defined as a graphical symbol or icon that is commonly used in digital communication to express emotions, ideas, or concepts. Emojis are often used to supplement or replace text-based communication to convey the tone, sentiment, or context of a message [7]. Emojis express strong sentiments in text data and add rich semantics and emotional information to online conversations in the form of facial expressions [8]. Emojis are frequently used in place of actual words by online users to express entire thoughts. They are facial expressions that have been repeatedly introduced in social interactions, and when they are ignored or unprocessed, they might harm the result of Sentiment classification. They are pictographs that are used in place of actual words in mainstream online conversations to describe facial emotions, places, persons, or objects (e.g., ☺). Because these characters are associated with texts, sentiments can be precisely communicated through them. Recently, individuals and corporate brands, are progressively communicating with customers using emojis [9]. According to Ko *et al.* [10], emojis have become an essential part of digital communication and cannot be ignored in sentiment analysis since they can significantly influence outcomes in brand-related user-generated content. Emojis can help convey emotions and provide context to a message, which can help determine the sentiment of the text. For instance, a smiley emoji in a text message can indicate a positive sentiment, whereas a crying emoji can indicate a negative sentiment. Thus, incorporating emojis into sentiment analysis can improve the accuracy and granularity of the analysis.. As a result, analyzing sentiment in documents without taking into account context information such as emojis could lead to sentiment misclassification. The use of the explicit negation word ("not") in sentences has a considerable effect on detecting feelings; yet, when such negation is improperly processed, sentiments are misclassified. Negation words in sentences have the effect of negative reversal, which changes the meaning of a word to its opposite.

The negative reversing effect of negation words and the nonverbal contents that are expressed using emoji contains vital information that can be used to determine the sentiments being expressed in online text data. We lose sentimental information in text data when they are left unprocessed or dropped during text cleaning. To address this problem, this paper proposes a preprocessor that considers both emoji features and explicit word negations before classifying sentiments.

The contributions of this work are summarized as follows:

- Propose a tweet preprocessor that reduces the loss of semantic information by extracting word negations from contractions.
- Propose a negation tracking algorithm that converts explicit word negations to their single word representations using the WordNet lexicon dictionary.
- Improve the text data with appropriate word representation of emoji-based content.

The remainder of this paper is organized as follows: In section 2, we highlight the related works. Section 3 presents the architecture of the proposed framework. In Section 4, we summarize the results of the experiment and discuss them before Section 5 presents the concluding remarks.

RELATED WORK

The emergence of Web 2.0 technologies has led to a surge in the number of internet users and an increase in online services and content. This has been facilitated through social networks such as Facebook, Twitter, Instagram, and LinkedIn, where users from diverse backgrounds can build relationships and interact with each other [11]. As a result, social media platforms have become a persistent social structure, generating vast amounts of user-generated content, primarily in the form of textual data. The availability of user-generated content on social media has made it a valuable source of information for a wide range of applications, including marketing, politics, and individual decision-making [12]. Social media platforms provide a wealth of data that can be analyzed to deliver real-time insights to stakeholders, enabling them to make informed decisions. However, the vast amounts of user-generated content available on social media require sophisticated data processing techniques to extract meaningful insights [13].

Sentiment analysis (SA) is a prominent application of Natural Language Processing (NLP) [14] that uses text analysis [15] to extract sentiments from reviews, identifying favorable and unfavorable opinions about specific products and services from a large amount of textual data [16]. SA usually categorizes texts into positive, negative, and neutral sentiments. For example, a review stating, "Chicken Republic is the best recipe" would be classified as positive, while "This movie was too long, and I got bored halfway through" would be classified as negative. By evaluating comments and reviews on social media, this technique can help determine the general attitude toward business brands, enabling them to improve the services rendered. Researchers have proposed numerous methods for various SA tasks, including lexicon-based approaches [17] or supervised machine-learning approaches [18]. Combining the unsupervised lexicon-based approach and supervised machine learning methods is recommended to improve the outcome of sentiment classification, as they complement each other, providing better results compared to using a single approach [19, 20]. This approach can be useful in identifying phenomena and handling the unstructured nature of text data [21]. SA has been applied to various domains, such as politics [22], products [23], services [16], public opinions, lifestyle, and health [24] .

Pröllochs *et al.* [25] developed a novel method for detecting and interpreting negations in natural language for sentiment analysis. They argue that negation processing is crucial for accurate sentiment analysis and present a reinforcement learning-based approach for this task. The paper also highlights the significance of language processing and natural language understanding in various applications, including recommender systems, financial news analysis, and question-answering systems.

To classify sentiments without losing any contextual information, Chen *et al.* [6] proposed a model using negative and supplementary information that overcomes the shortcoming of Xu et al. [26]. The model exploits the linguistic features of negative and intensive words as well as contextual information to address the sentiment reversing the effect of negative words and the sentiment-shifting effect of intensive words. Pröllochs *et al.* [25] identified the difficulty in recognition and interpretation of negation in text data, they developed a reinforced learning framework for the task of inferring the meaning of narrative content in the presence of negations by learning a negation policy.

Ullah *et al.* [27] proposed an algorithm and method that handles both text and emoticons (bi-mode) in the data. Their study highlighted the impact of emoticons in sentiment classification tasks as well as deep learning systems' high classification accuracy. These studies also demonstrated the value of enhancing semantic relationships in text data. Singh and Paul [4] proposed a deep learning approach for negation handling in sentiment analysis. The authors highlighted the importance of negation handling in accurate sentiment analysis and discussed the limitations of existing rule-based approaches in handling negation in natural language. The proposed approach utilized a bidirectional LSTM to identify negation cues and extract the scope of the cue in sentences, along with word-level features to determine the correct polarity of the sentence. The model was trained and tested on a pre-annotated corpus of Conan Doyle stories, achieving better performance compared to traditional rule-based models

Mukherjee *et al.* [28] proposed a strategy for handling syntactic and morphological negations in sentences because their use has such a big impact on detecting the polarity of sentiment when improperly processed. Identifying negations, according to Mukherjee *et al.* [28], improves the classification of sentiments. Recently, deep learning algorithms have been widely used for sentiment classification tasks. Colón-Ruiz & Segura-Bedmar [29] compared various deep learning architectures with state-of-the-art language models for classifying sentiments on drug reviews. Other domain-specific application of deep learning techniques on SA includes movie reviews [30], and Hotel reviews [31]. Naseem *et al.* [32] proposed an intelligent framework known as $DICE_T$, a method for effectively handling complex attributes in words, the noisy context of tweets, and ambiguities. Extensive experiments based on the framework showed it outperformed start-of-the-art SA methods. To improve the performance of LSTM, Shobana & Murali [33] proposed an efficient sentiment classification method based on optimizing weight parameters using an adaptive PSO Algorithm. Their experiments showed that the proposed APSO-LSTM model outperformed traditional techniques in terms of accuracy.

Researchers have also offered several strategies to deal with real-world scenarios. In the case of the COVID-19 outbreak, on Twitter, Bangyal *et al.* [34] provided a highly accurate strategy for detecting fake news using deep learning approaches. Sunitha *et al.*[35] also proposed an ensemble-based deep learning model (Gated Recurrent Unit (GRU) and Capsule Neural Network (CapsNet)) for real-time sentiment classification of COVID-19 tweets across India and Europe. In light of the #BlackLivesMatter protest/movement on Twitter, Ankita *et al.* [36] proposed an end-to-end deep learning model using CNN-LSTM for efficient Sentiment detection of user tweets for individuals that are based in the USA (Minnesota and Washington DC). The results of the proposed method were shown to outperform existing machine learning models (Random Forrest), and deep learning models (CNN, LSTM, and BI-LSTM) thus validating their model. To perform sentiment classification on Bangla text, Bhowmik *et al.* [20] proposed an extended lexical dictionary and deep learning methods. They used a rule-based Bangla text sentiment score (BTSC) algorithm to extract polarity from large texts. These polarities, along with the preprocessed text, are sent into the neural network as the training set. Their proposed improved long short-term memory (LSTM) models showed high accuracy in performing SA tasks, according to experimental results.

With emojis becoming more ubiquitous in online interactions, studies have shown that when emojis replace words, sentence comprehension does not deteriorate [37]. Although emoji's meanings vary depending on where they are used, Shardlow et al. [7] pointed out that they are polysemous, and their experiment demonstrated that it is possible to predict an emoji's meaning in NLP tasks. Liu *et al.* [5] identified that emojis work well as additional features for enhancing the accuracy of sentiment analysis algorithms and that the algorithm's performance can be improved even more by considering varied emoji usages.

DATASET DESCRIPTION PROPOSED MODEL

This section presents the methodology used in the research comprising three main phases: (i) text preprocessing, (ii) feature representation, and (iii) modeling. The complete proposed conceptual framework is presented in Figure 1.

Dataset

A total of three (3) datasets were collected from Twitter and Kaggle. However, the dataset from Kaggle has its root in Twitter. As shown in Table 1, each dataset has its name and category. These categories public opinions on general topics.

The instance column describes the total number of documents/reviews (tweets) in the data. These datasets contain two (2) known classes: positive and negative classes.

Table 1. Dataset description

Dataset Name	Category	Instances	Classes
Reviews-2019092	General	82,815	2
Sentiment140	General	1,600,000	2
Twitter Dataset	General	2,000	2

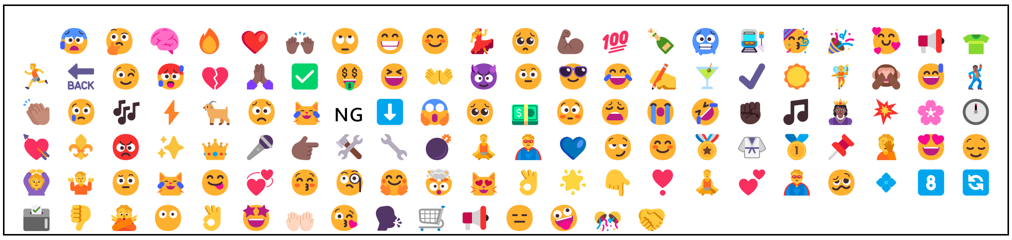
Text Preprocessing

To begin the initial phase of our methodology, we tokenize the document; it was utilized to handle specific entities of the text data by breaking down sentences into tokens as a feature extraction phase using the NLTK (Natural Language Toolkit) package. Words, symbols, and emojis are examples of tokens. The rest of the text preprocessing methods includes conversion to lowercase, stop word removal, lemmatization, part-of-speech tagging, and removal of other unwanted characters. These methods are carried out at a later stage in other to preserve the emoji and negation elements in the text data. Tokenization: this technique was used to separate the raw texts into words known as tokens. Tokens are employed to help in the comprehension of a word's context. Evaluating the word sequence, this phase will help in interpreting the meaning of the text in NLP tasks.

Emoji Identification

Similar to the methodology by Fernández-Gavilanes *et al.* [38] and Redmond *et al.* [39], our research considers EmojiPedia as the source for our emoji descriptions to build our emoji dictionary. This dictionary is used to replace emojis in the text data with relevant and contextual English words. Some of the emojis present in the dataset are shown in Table 2. The details for the emoji conversion algorithm are depicted in Algorithm 1.

Table 2. Emoji in datasets used



Algorithm 1: Emoji conversion algorithm

Input: review (texts and emoji tokens)
Output: new_review (text with emoji description)

```
emoji_dict = {
        "😜": "happy",
        …
        "😞": "sad",
        "😠": "angry"
        }
For token in word_tokens
    Convert all emoji keys to their value pairs in the dictionary
End for
return new_review
```

Negation Handling

Word contractions such as "can't" and "didn't" among others are known to have shortened word negations [40]. To handle such negations, we also build a contraction dictionary to accommodate such occurrences in the data. Subsequently, we added popular internet acronyms to this dictionary to further enrich the data instead of cleaning them out in further text cleaning. The contraction dictionary algorithm is depicted in Algorithm 2.

Algorithm 2: Contraction words vocabulary dictionary

```
Contraction_vocab_mapper = {
        "didn't": "did not",
        "wasn't": "was not",
        …
        "aren't": "are not"
        }
```

The contraction dictionary will help obtain the shortened negations in the text data to enable the negation algorithm to convert them into the appropriate antonyms. Negation handling in this work is based on the WordNet lexicon dictionary. A recent negation handling algorithm was proposed by Mukherjee et al. [28] where negations are identified by adding a "_NEG" suffix to the negated word. This means that each time a word is negated, the negation algorithm appends the negation suffix while also converting the word to its lemmatized form. Following up on this approach, the proposed negation algorithm identifies the negated word and returns its antonym. In contrast to this approach, we use the Lexicon dictionary. This dictionary contains both synonyms and synsets of English words. We further explore this synset to retrieve the antonyms of negated words such that phrases such as "don't like" would be converted to "dislike". In so doing, we can handle explicit word negations in the text data. Spelling errors were corrected in the event of misspelled words that might be lost yet relevant to the data. The negation handling algorithm is shown in Algorithm 3.

Algorithm 3: Negation handling algorithm

Input: review (text tokens)
Output: new_review (text with normalized negation)

```
temp_word ← "empty string"
For word in tokens
    If word == "not"
        Initialize a new variable for Negation word "not "
        For synonyms in WordNet
            Get the synsets of the negated word
            For synonym in synonyms
                Get lemmas of the synonym
                For antonym in synonym
                    Get and append antonym of negated word
        If length of antonyms >= 1
            word = antonyms(0)
        Else
            word = temp_word + word
    If word ≠ "not"
        Return review
Return new_review
```

Text Cleaning

We proceed to clean the text data; this step rids the text data of irrelevant (noise) data before feature representation. We begin by converting all characters to lowercase. Part-of-Speech tagging is done on the respective word tokens maintaining the grammatical contexts in the text data. Words such as "a", "is", "the", "do", etc. are common words in the English language that do not add any information to the classifier. They are known as stop words and since they do not contribute to the sentimental contents of the text, they are eliminated. Another preprocessing step is lemmatization. This text preprocessing technique removes all inflectional endings of a word and returns its dictionary form, also known as the lemma. For this process, we use the WordNet Lemmatizer. Finally, in the text cleaning phase, we removed unwanted characters in the text data. This includes characters such as punctuations and special characters (symbols), URLs, @username, multiple whitespaces, and numerals.

Word embedding models

To initialize the classifier, we need a feature representation model that will serve as an input. For this work, the model was trained using Global Vectors (GloVe). It is a count-based language model that derives semantic relationships between words from a co-occurrence matrix. Unlike language models like word2vec which only capture local context, GloVe can capture both the local and global context of a word. To represent the data using the GloVe language model, each text is denoted as a matrix of word embedding. Since each of the reviews in the data are of different lengths because some of the reviews are longer while others are short, we applied padding and/or truncating to obtain equal lengths for the entire review. This length defines the sequence length for the input layer of the LSTM classifier

The long-short-term memory classifier

Numerous literature has recorded the success of applying Recurrent Neural Networks (RNN) in the classification of sequential data [41, 42], although it suffers from the vanishing gradient problem (where the gradient comes close to zero). LSTM is a variation of the traditional RNN that was designed to capture long-term dependencies in sequential data. It is built around a memory cell that uses output, input, and forget gates to control the read, write, and reset operations of its internal state. The forget gate will decide which information to keep and which to offload at one time, t, using the current input, x_t, and output from the previous state h_{t-1}, and then update the memory cell.

This is achieved by the three gates in the recurrently connected memory block. These gates are the forget, input, and output gates that regulate the flow of information in the current time step, consequently, solving the vanishing gradient problem of the traditional RNN, the LSTM cell is expressed as follows:

$$f_t = \sigma_g(W_f \times x_t + U_f + h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i \times x_t + U_i + h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o \times x_t + U_o + h_{t-1} + b_o)$$
$$c'_t = \sigma_c(W_c \times x_t + U_c + h_{t-1} + b_c)$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$$
$$h_t = o_t \cdot \sigma_c(c_t)$$

where, f_t, forget gate, i_t, input gate, o_t, input gate, c_t, cell state, and h_t is the hidden state. Also, σ_g, is the sigmoid, and σ_c, is the tanh activation function.

Table 3 describes the parameter settings used for the LSTM classifier.

Table 3. Parameter values for the classifier

Parameter	Value
Embedding dimension	150D
Optimizer	Adam
Epochs	15
Dropout	0.4
Activation	Sigmoid
Learning Rate	0.001
Training set	70%
Testing set	30%

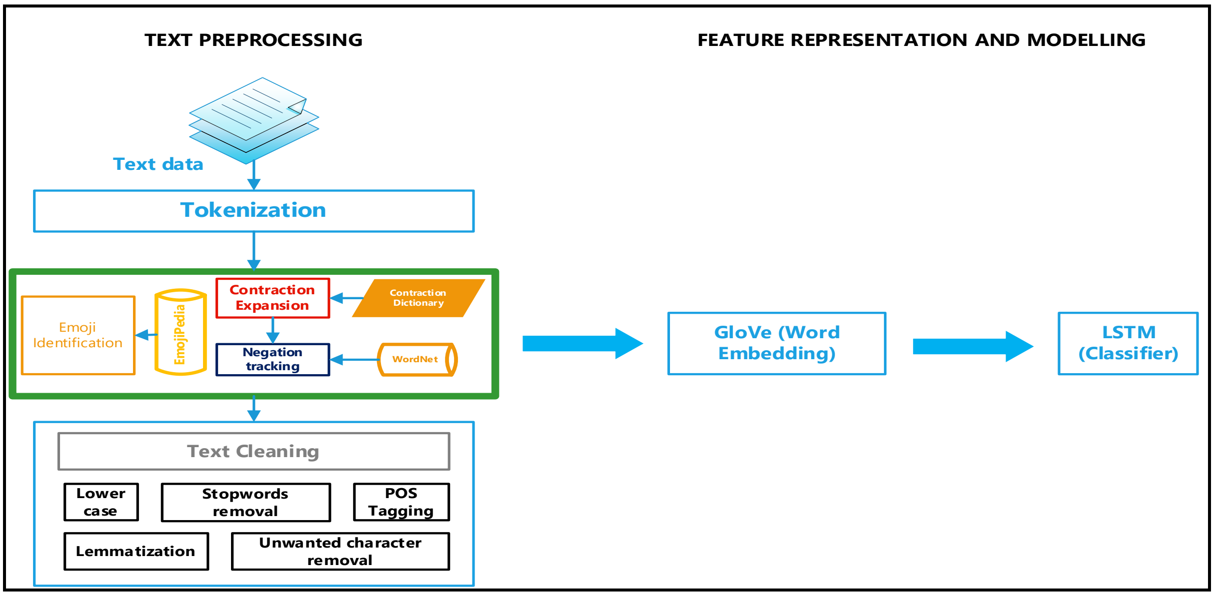


Figure 1. A proposed conceptual framework of SA for negation words and emoji-based contents based on long short-term memory.

RESULTS AND DISCUSSION

The results obtained from the implementation of the methodology in Figure 1 are presented and discussed in this section. It is comprised of the performance evaluation metrics, ablation analysis, and performance evaluation of the proposed RNN, and ANN, and finally, its comparison with similar techniques.

Evaluation metrics

To evaluate the performance of the proposed framework, four commonly used evaluation metrics for sentiment analysis are used, these include; recall, precision, accuracy, and F1-score. Here's an explanation of each of these metrics and their formulas:

Recall: Recall measures the ability of the model to identify all relevant instances of a particular sentiment. The formula for recall is:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

where True Positives are the instances where the model correctly identifies a particular sentiment and False Negatives are the instances where the model incorrectly identifies a different sentiment.

Recall is an important metric for sentiment analysis as it helps to ensure that the model is not missing any relevant instances of a particular sentiment.

Precision: Precision measures the ability of the model to correctly identify a particular sentiment. The formula for precision is:

$$Precision = \frac{True\ Positives}{True\ Positive + False\ Positive}$$

where False Positives are the instances where the model incorrectly identifies a particular sentiment.

Precision is an important metric for sentiment analysis as it helps to ensure that the model is not falsely identifying irrelevant instances as a particular sentiment.

Accuracy: Accuracy measures the overall performance of the model in terms of correctly identifying all sentiments. The formula for accuracy is:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + True\ Negatives + False\ Negatives}$$

where True Negatives are the instances where the model correctly identifies a different sentiment.

Accuracy is an important metric for sentiment analysis as it provides an overall measure of the model's performance.

F1-score: F1-score is a harmonic mean of precision and recall and provides a balanced measure of the model's performance. The formula for F1-score is:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

F1-score is an important metric for sentiment analysis as it provides a balanced measure of both precision and recall.

These evaluation metrics are commonly used in sentiment analysis experiments and can be justified by their ability to measure different aspects of the model's performance. Recall and precision are important metrics for ensuring that the model is correctly identifying relevant instances and avoiding false positives. Accuracy provides an overall measure of the model's performance, while the F1 score provides a balanced measure of precision and recall.

Ablation analysis

We first perform an ablation breakdown of the proposed framework on benchmark Twitter datasets to investigate the performance gain in terms of classification accuracy. Data imbalance was handled using up-sampling to create a well-adjusted dataset. This is because we can retain all the information in the training set as opposed to down-sampling which tends to drop significant information from the data. Five experiments were conducted repeatedly with different combinations. The results are summarized and presented in Table 4. In experiment 1, we use the LSTM classifier to build the sentiment classification model using text_to_sequences to transform each word in our text into a sequence of integers where each integer corresponds to a value from the word_index dictionary.

Table 4. Ablation Analysis of the Proposed Model

Experiment	Model	Datasets		
		Reviews2019092	Sentiment140	Twitter dataset
1	LSTM	68.38%	68.75%	72.95%
2	GloVe + LSTM	93.56%	83.98%	89.40%
3	Emoji Preprocessor + GloVe + LSTM	93.59%	85.27%	92.85%
4	Contraction Dictionary + Negation Tracker + GloVe + LSTM	95.39%	86.59%	94.42%
5	(Emoji + Contraction + Negation) preprocessor + GloVe + LSTM	96.36%	88.41%	95.39%

In experiment 2, we introduce the GloVe word embedding language model without any of the preprocessors to examine the performance of the model. This model uses the similarity between words as an invariant, it attempts to produce a vector representation of words. The model makes use of two different model techniques, notably the Skip-gram and Continuous Bag of Words Models (CBOW). The former has a computational time issue (despite strong accuracy), whereas the latter has a low accuracy issue (although computationally efficient). GloVe seeks to integrate the approaches of two models, and it has proven to be more accurate and efficient than the other two [43]. We choose this language model because, unlike models like word2vec and TF-IDF, it can capture word co-occurrence (global statistics) to generate word vectors. In adopting this model, we used an embedding dimension of 150D to train the classifier.

After introducing the GloVe language model, we introduce our Emoji preprocessor in experiment 3 to convert all emojis into their word representations. Emojis are often used to augment, clarify, and even enhance the meanings of tweets, and they are widely shared on Twitter. Research has been conducted on the impact of emojis on social media interaction [37, 7, 44]. We use the emoji preprocessor in this experiment to augment the vocabulary with word representations of each emoji in the text input. In experiment 4, we introduced a contraction dictionary to retrieve the negation words that are part of forming a contraction through shortening and/or merging two words to become a single word. Most of these contractions are negative verbs. This preprocessing step helps in obtaining the opposite of words that are negated in text data using the word "not" as a prefix. Negations are a crucial component of expressing sentiments and thoughts, thus this is a necessary stage [45].

Finally, in experiment 5, we combine both preprocessors from experiments 4 and 5 to see the impact on the sentiment classification model. As we progress from the initial experiment, there is a significant increase in the classification accuracies, there is an increase in accuracy when the emoji is introduced, this can be attributed to the presence of either a few or more emoji in some of the datasets. The overall strength of the proposed model as seen in the classification accuracy results in Table 4 (Experiment 5) can be attributed to the preprocessor that was introduced to the sentiment classification task.

Performance comparison with another Negation algorithm(s)

We perform another comparison with a negation algorithm in terms of accuracy on a product review dataset (Reviews-2019092) using the Recurrent Neural Network (RNN) and Artificial Neural Network (ANN). We first perform a comparison between our proposed model (LSTM + Negation) with two baseline models (RNN + Negation, and ANN + Negation) that use the same negation algorithm. It can be observed that the ANN + Negation model outperforms the RNN + Negation with 0.37% and outperforms our model (LSTM + Negation i.e., experiment 4) with 0.28%. However, our proposed model outperforms the RNN + Negation model with higher accuracy of 0.09%. The results of this comparison are illustrated in Figure 2.

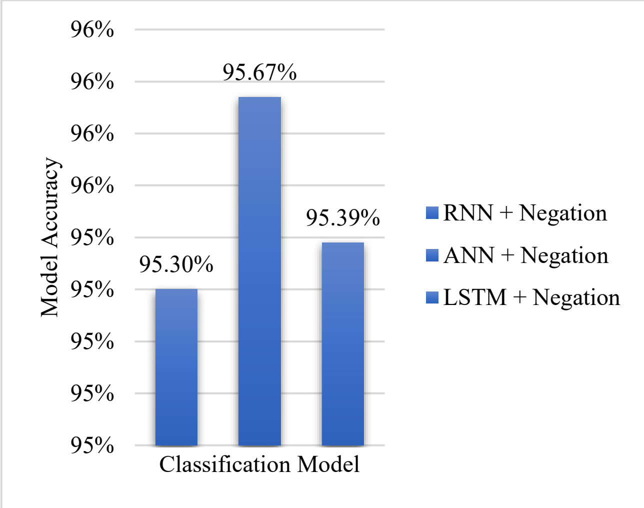


Figure 2: Performance comparison of the proposed model with other Negation Preprocessors without emoji pre-processor

We perform another comparison where we included the emoji pre-processor in the model to compare the performance of the proposed model with the existing models. From this comparison, we can see that the proposed model outperforms the RNN + Negation model by 1.06% and ANN + Negation model by 0.69%. Figure 3 illustrates the results of this comparison. From these comparisons, we can conclude that handling negations words and emoji in the dataset helps include more contents that carry information in the text data, thus, improving the performance of the SA.

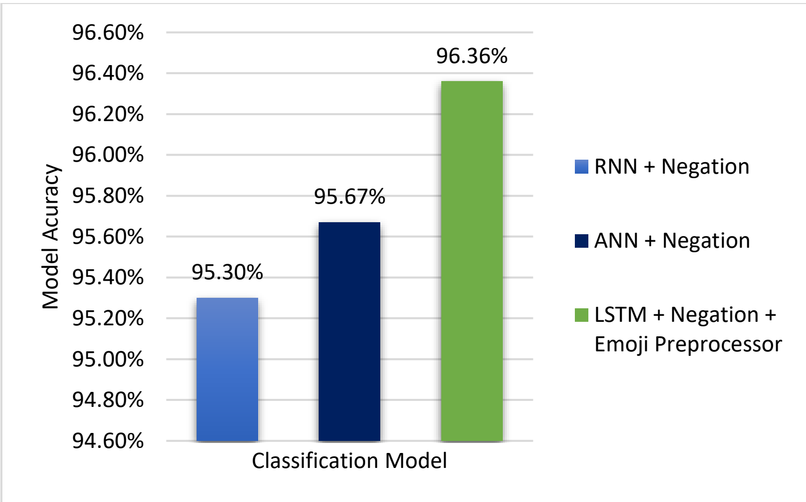


Figure 3: Performance comparison of the proposed model with other Negation Preprocessors with emoji processor

Finally, the proposed initial preprocessors are introduced before building the RNN and ANN models to evaluate the performance, and the results obtained are illustrated in Figure 4. The LSTM model outperforms the RNN and ANN models by 0.47% and 0.55% respectively. Consequently, the RNN model outperformed the ANN model by 0.08%.

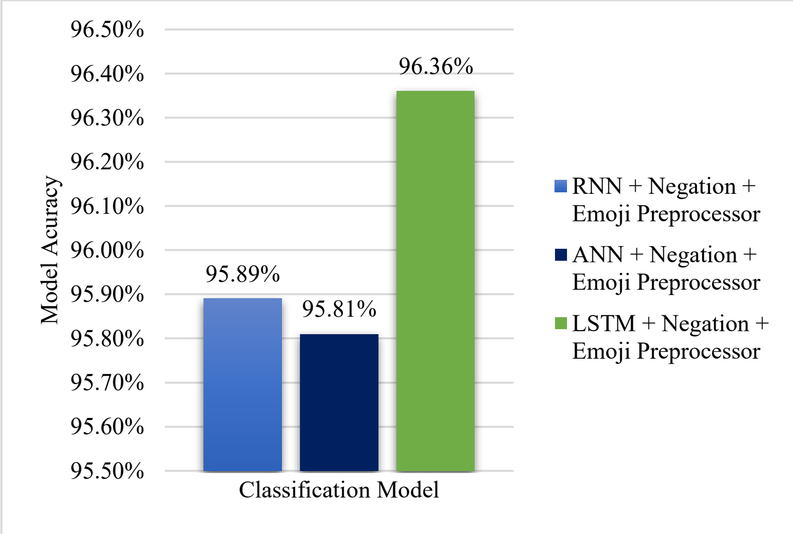


Figure 4. Comparison with ANN and RNN

In Table 5 and Table 6, we show the results obtained from the experiment on the three used benchmark datasets. Evaluations of the model's performance were based on the Recall, Precision, Accuracy, and F1-Score metric. The result obtained from the experiment using the Amazon Reviews-2019092 dataset as shown in Table 4 was compared to the work of Mukherjee et al. [28]. Both approaches show competing results such as the case of the recall (an increase of 0.23%), however, in the cases of precision, accuracy, and F1-Score, we see a performance increase of 3.14%, 0.69%, and 1.62% respectively. These performance increases can be attributed to the emoji preprocessor as shown in Figure 2 which helps preprocess emoji characters in the data.

Table 5. Performance comparison of the proposed method with that of Mukherjee *et al.* [28] on the Reviews-2019092 dataset

Proposed Approach				**Compared: Mukherjee et al. [28]**			
Recall	Precision	Accuracy	F1-Score	Recall	Precision	Accuracy	F1-Score
94.61%	99.98%	96.36%	97.22%	94.38%	96.84%	95.67%	95.60%

Table 6 shows the performance of the proposed model in terms of classification accuracy on Twitter datasets; Testdata.manual.2009.06.14, and Twitter dataset and compared to the results from Chandra *et al.* [46].

Table 6. Performance comparison of the proposed method with that of Chandra *et al.* [46] on Testdata.manual.2009.06.14 and Twitter datasets

Related Papers	**Dataset**	**Accuracy**
Chandra et al. [4]	Testdata.manual.2009.06.14	78.17%
	Twitter dataset	67.45%
Proposed Approach	Testdata.manual.2009.06.14	88.41%
	Twitter dataset	95.39%

The results obtained from Table 6 show that our model outperforms the compared model in terms of the available evaluation metric; accuracy. Chandra *et al.* [46] used K-means with Cuckoo Search for SA. However, at the feature extraction stage, similar features have been put into consideration such as Emojis and Negations using similar datasets. The high classification accuracy of the proposed model is due to the effectiveness of the Emoji dictionary (EmojiPedia) and the Lexicon dictionary (WordNet) to handle the features that are likely to be dropped during preprocessing while they contain sentimental contents and converted into vectors using GloVe for efficient Sentiment Classification.

CONCLUSION

A word representational framework for sentiment classification tasks was proposed in this study. While preprocessing the text data before classification, the proposed framework takes explicit negations words and emojis into account. This is accomplished by first dealing with word negations, extending contractions to recover reduced negation terms, and then using the Negation algorithm to track prefixed negations. To avoid this loss of information and the possibility of misclassification, we used the WordNet Lexicon dictionary to transform these words into their antonyms. Furthermore, an emoji dictionary was created and linked to a contextual word description using EmojiPedia. As a result, a framework was developed for translating emojis to words before text cleaning. GloVe was used to convert the text data into dense vectors, which were then utilized as input for the LSTM classifier to experiment. On the three benchmark datasets employed, we attained an accuracy of 96.36 percent, 88.41 percent, and 95.39 percent after running the tests. The findings of this study demonstrate that while assessing sentiments, preprocessing information such as emojis and explicit word negations helps to keep their emotive meaning.

REFERENCES

[1] Hassonah, M. A., Al-Sayyed, R., Rodan, A., Al-Zoubi, A. M., Aljarah, I., & Faris, H. (2020). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. Knowledge-Based Systems, 192, 105353. https://doi.org/10.1016/j.knosys.2019.105353

[2] Appel, G., Grewal, L., Hadi, R., & Stephen, A. T. (2020). The future of social media in marketing. Journal of the Academy of Marketing Science, 48(1), 79–95. https://doi.org/10.1007/s11747-019-00695-1

[3] Dwivedi, Y. K., Ismagilova, E., Hughes, D. L., Carlson, J., Filieri, R., Jacobson, J., Jain, V., Karjaluoto, H., Kefi, H., Krishen, A. S., Kumar, V., Rahman, M. M., Raman, R., Rauschnabel, P. A., Rowley, J., Salo, J., Tran, G. A., & Wang, Y. (2021). Setting the future of digital and social media marketing research: Perspectives and research propositions. International Journal of Information Management, 59. https://doi.org/10.1016/j.ijinfomgt.2020.102168

[4] Singh, P. K., & Paul, S. (2021). Deep Learning Approach for Negation Handling in Sentiment Analysis. IEEE Access, 9, 102579–102592. https://doi.org/10.1109/ACCESS.2021.3095412

[5] Liu, C., Fang, F., Lin, X., Cai, T., Tan, X., Liu, J., & Lu, X. (2021). Improving sentiment analysis accuracy with emoji embedding. Journal of Safety Science and Resilience, 2(4), 246–252. https://doi.org/10.1016/J.JNLSSR.2021.10.003

[6] Chen, X., Rao, Y., Xie, H., Wang, F. L., Zhao, Y., & Yin, J. (2019). Sentiment Classification Using Negative and Intensive Sentiment Supplement Information. Data Science and Engineering 2019 4:2, 4(2), 109–118. https://doi.org/10.1007/S41019-019-0094-8

[7] Shardlow, M., Gerber, L., & Nawaz, R. (2022). One emoji, many meanings: A corpus for the prediction and disambiguation of emoji sense. Expert Systems with Applications, 198, 116862. https://doi.org/10.1016/J.ESWA.2022.116862

[8] Sun, X., Li, H., Sun, G., & Zhu, M. (2020). Fine-grained emoji sentiment analysis based on attributes of Twitter users. Proceedings - 2020 IEEE International Conference on Smart Cloud, SmartCloud 2020, 134–139. https://doi.org/10.1109/SMARTCLOUD49737.2020.00033

[9] McShane, L., Pancer, E., Poole, M., & Deng, Q. (2021). Emoji, Playfulness, and Brand Engagement on Twitter. Journal of Interactive Marketing, 53, 96–110. https://doi.org/10.1016/J.INTMAR.2020.06.002

[10] Ko, E. (Emily), Kim, D., & Kim, G. (2022). Influence of emojis on user engagement in brand-related user generated content. Computers in Human Behavior, 136, 107387. https://doi.org/10.1016/J.CHB.2022.107387

[11] Rahmanti, A. R., Chien, C.-H., Nursetyo, A. A., Husnayain, A., Wiratama, B. S., Fuad, A., Yang, H.-C., & Li, Y.-C. J. (2022). Social media sentiment analysis to monitor the performance of vaccination coverage during the early phase of the national COVID-19 vaccine rollout. Computer Methods and Programs in Biomedicine, 221, 106838. https://doi.org/10.1016/J.CMPB.2022.106838

[12] Xu, Q. A., Chang, V., & Jayne, C. (2022). A systematic review of social media-based sentiment analysis: Emerging trends and challenges. Decision Analytics Journal, 3, 100073. https://doi.org/10.1016/J.DAJOUR.2022.100073

[13] Qian, C., Mathur, N., Zakaria, N. H., Arora, R., Gupta, V., & Ali, M. (2022). Understanding public opinions on social media for financial sentiment analysis using AI-based techniques. Information Processing & Management, 59(6), 103098. https://doi.org/10.1016/J.IPM.2022.103098

[14] Hiriyannaiah, S., Srinivas, A. M. D., Shetty, G. K., G.M., S., & Srinivasa, K. G. (2020). A computationally intelligent agent for detecting fake news using generative adversarial networks. Hybrid Computational Intelligence, 69–96. https://doi.org/10.1016/B978-0-12-818699-2.00004-4

[15] Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (n.d.). Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. Retrieved August 26, 2021, from www.cnet.com.

[16] Al-Qudah, D. A., Al-Zoubi, A. M., Castillo-Valdivieso, P. A., & Faris, H. (2020). Sentiment analysis for e-payment service providers using evolutionary extreme gradient boosting. IEEE Access, 8, 189930–189944. https://doi.org/10.1109/ACCESS.2020.3032216

[17] Catelli, R., Pelosi, S., Comito, C., Pizzuti, C., & Esposito, M. (2023). Lexicon-based sentiment analysis to detect opinions and attitude towards COVID-19 vaccines on Twitter in Italy. Computers in Biology and Medicine, 158, 106876. https://doi.org/10.1016/J.COMPBIOMED.2023.106876

[18] Sarkar, D. (2019). Natural Language Processing Basics. Text Analytics with Python, 1–68. https://doi.org/10.1007/978-1-4842-4354-1_1

[19] Drus, Z., & Khalid, H. (2019). Sentiment analysis in social media and its application: Systematic literature review. Procedia Computer Science, 161, 707–714. https://doi.org/10.1016/j.procs.2019.11.174

[20] Bhowmik, N. R., Arifuzzaman, M., & Mondal, M. R. H. (2022). Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. Array, 13, 100123. https://doi.org/10.1016/J.ARRAY.2021.100123

[21] El Rahman, S. A., Alotaibi, F. A., & Alshehri, W. A. (2019). Sentiment Analysis of Twitter Data. 2019 International Conference on Computer and Information Sciences, ICCIS 2019. https://doi.org/10.1109/ICCISCI.2019.8716464

[22] Oyewola, D. O., Oladimeji, L. A., Julius, S. O., Kachalla, L. B., & Dada, E. G. (2023). Optimizing sentiment analysis of Nigerian 2023 presidential election using two-stage residual long short term memory. Heliyon, 9(4), e14836. https://doi.org/10.1016/J.HELIYON.2023.E14836

[23] Suhartono, D., Purwandari, K., Jeremy, N. H., Philip, S., Arisaputra, P., & Parmonangan, I. H. (2023). Deep neural networks and weighted word embeddings for sentiment analysis of drug product reviews. Procedia Computer Science, 216, 664–671. https://doi.org/10.1016/J.PROCS.2022.12.182

[24] Kaur, H., Ahsaan, S. U., Alankar, B., & Chang, V. (2021). A Proposed Sentiment Analysis Deep Learning Algorithm for Analyzing COVID-19 Tweets. Information Systems Frontiers, 1–13. https://doi.org/10.1007/s10796-021-10135-7

[25] Pröllochs, N., Feuerriegel, S., Lutz, B., & Neumann, D. (2020). Negation scope detection for sentiment analysis: A reinforcement learning framework for replicating human interpretations. Undefined, 536, 205–221. https://doi.org/10.1016/J.INS.2020.05.022

[26] Xu, Z., Fu, Y., Chen, X., Rao, Y., Xie, H., Wang, F. L., & Peng, Y. (2018). Sentiment classification via supplementary information modeling. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10987 LNCS, 54–62. https://doi.org/10.1007/978-3-319-96890-2_5

[27] Ullah, M. A., Marium, S. M., Begum, S. A., & Dipa, N. S. (2020). An algorithm and method for sentiment analysis using the text and emoticon. ICT Express, xxxx, 10–13. https://doi.org/10.1016/j.icte.2020.07.003

[28] Mukherjee, P., Badr, Y., Doppalapudi, S., Srinivasan, S. M., Sangwan, R. S., & Sharma, R. (2021). Effect of Negation in Sentences on Sentiment Analysis and Polarity Detection. Procedia Computer Science, 185, 370–379. https://doi.org/10.1016/J.PROCS.2021.05.038

[29] Colón-Ruiz, C., & Segura-Bedmar, I. (2020). Comparing deep learning architectures for sentiment analysis on drug reviews. Journal of Biomedical Informatics, 110, 103539. https://doi.org/10.1016/J.JBI.2020.103539

[30] Chakraborty, K., Bhattacharyya, S., Bag, R., & Hassanien, A. A. (2019). Sentiment Analysis on a Set of Movie Reviews Using Deep Learning Techniques. Social Network Analytics, 127–147. https://doi.org/10.1016/B978-0-12-815458-8.00007-4

[31] Muhammad, P. F., Kusumaningrum, R., & Wibowo, A. (2021). Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews. Procedia Computer Science, 179, 728–735. https://doi.org/10.1016/j.procs.2021.01.061

[32] Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis. Future Generation Computer Systems, 113, 58–69. https://doi.org/10.1016/J.FUTURE.2020.06.050

[33] Shobana, J., & Murali, M. (2021). An efficient sentiment analysis methodology based on long short-term memory networks. Complex & Intelligent Systems, 7(5), 2485–2501. https://doi.org/10.1007/s40747-021-00436-4

[34] Bangyal, W. H., Qasim, R., Rehman, N. U., Ahmad, Z., Dar, H., Rukhsar, L., Aman, Z., & Ahmad, J. (2021). Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches. Computational and Mathematical Methods in Medicine, 2021. https://doi.org/10.1155/2021/5514220

[35] Sunitha, D., Patra, R. K., Babu, N. V., Suresh, A., & Gupta, S. C. (2022). Twitter sentiment analysis using ensemble based deep learning model towards COVID-19 in India and European countries. Pattern Recognition Letters, 158, 164–170. https://doi.org/10.1016/J.PATREC.2022.04.027

[36] Ankita, Rani, S., Bashir, A. K., Alhudhaif, A., Koundal, D., & Gunduz, E. S. (2022). An efficient CNN-LSTM model for sentiment detection in #BlackLivesMatter. Expert Systems with Applications, 193, 116256. https://doi.org/10.1016/J.ESWA.2021.116256

[37] Scheffler, T., Brandt, L., Fuente, M. de la, & Nenchev, I. (2022). The processing of emoji-word substitutions: A self-paced-reading study. Computers in Human Behavior, 127, 107076. https://doi.org/10.1016/J.CHB.2021.107076

[38] Fernández-Gavilanes, M., Costa-Montenegro, E., García-Méndez, S., González-Castaño, F. J., & Juncal-Martínez, J. (2021). Evaluation of online emoji description resources for sentiment analysis purposes. Expert Systems with Applications, 184, 115279. https://doi.org/10.1016/J.ESWA.2021.115279

[39] Redmond, M., Salesi, S., & Cosma, G. (2017). A Novel Approach Based on an Extended Cuckoo Search Algorithm for the Classification of Tweets which contain Emoticon and Emoji.

[40] Sharif, W., Samsudin, N. A., Deris, M. M., & Naseem, R. (2017). Effect of negation in sentiment analysis. 2016 6th International Conference on Innovative Computing Technology, INTECH 2016, 718–723. https://doi.org/10.1109/INTECH.2016.7845119

[41] Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. Future Generation Computer Systems, 115, 279–294. https://doi.org/10.1016/j.future.2020.08.005

[42] Usama, M., Ahmad, B., Song, E., Hossain, M. S., Alrashoud, M., & Muhammad, G. (2020). Attention-based sentiment analysis using convolutional and recurrent neural network. Future Generation Computer Systems, 113, 571–578. https://doi.org/10.1016/j.future.2020.07.022

[43] Sharma, Y., Agrawal, G., Jain, P., & Kumar, T. (2018). Vector representation of words for sentiment analysis using GloVe. ICCT 2017 - International Conference on Intelligent Communication and Computational Techniques, 2018-January, 279–284. https://doi.org/10.1109/INTELCCT.2017.8324059

[44] Wu, R., Chen, J., Lu Wang, C., & Zhou, L. (2022). The influence of emoji meaning multipleness on perceived online review helpfulness: The mediating role of processing fluency. Journal of Business Research, 141, 299–307. https://doi.org/10.1016/J.JBUSRES.2021.12.037

[45] Mansour, R., Hady, M. F. A., Hosam, E., Amr, H., & Ashour, A. (2015). Feature selection for twitter sentiment analysis: An experimental study. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9042, 92–103. https://doi.org/10.1007/978-3-319-18117-2_7/COVER/

[46] Chandra, P. A., Singh Rajpoot, D., & Saraswat, M. (2017). Twitter sentiment analysis using hybrid cuckoo search method. Information Processing and Management, 53(4), 764–779. https://doi.org/10.1016/j.ipm.2017.02.004