International Journal of Data Science and Analytics (2021) 12:355–367
https://doi.org/10.1007/s41060-021-00277-x

REGULAR PAPER

Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic

M. Sivakumar[1,2] · Srinivasulu Reddy Uyyala[2]

Received: 31 March 2021 / Accepted: 13 July 2021 / Published online: 5 August 2021

Abstract

Nowadays, reviews about the products on online shopping sites become an essential source to help the customer to take better decisions on buying a product and achieve good sales of the product. It becomes a habit for the consumers to share opinions about a recently purchased product immediately on online shopping sites and social media websites. So, there is a huge demand for an intelligent system to detect sentiments from customer reviews under specific aspects of a product posted on online shopping sites. In recent years, various machine learning techniques have been experimented on various benchmark datasets to analyze sentiments expressed by the consumer through online portals. But, consumers are still struggling to get aspect-based sentiments expressed by other consumers, and the accuracy of the existing model is not satisfactory. Hence, we proposed an intelligent system using long-term short memory with fuzzy logic to classify consumer review sentences under various aspects with four different labels, namely highly negative, negative, positive and highly positive. So, consumers who wish to buy a new product from online portal can see multi-label sentiments of the various aspects of the product quickly. The proposed system was experimented on Amazon cell phone review, Amazon video games review and consumer reviews of amazon products benchmark datasets and obtained the results with the accuracy of 96.93%, 83.82% and 90.92%, respectively. The proposed model outperforms in terms of accuracy when compared to the state-of-the-art-methods. The developed system also analyzed the product reviews based on the current trends and geographical location. The proposed system aids the manufacturers for improving the products based on customer complaints.

Keywords Mobile phone reviews · LSTM · Recurrent neural network · Sentiment analysis · Product review · Artificial intelligence · Fuzzy logic

1 Introduction

Customer opinions in online shopping portal help other customers to come out of confusion on choosing a right product. People wish to know about pros and cons of a product before purchasing, which removes their ambiguity about the product selection. This will take more time to get confident on a particular product by analyzing all the positive and negative sides of the product though it is a low price product. The products under various brands and specifications may confuse the customer by offering discounts and attractive gifts. On the other hand, there are vast amount of opinionated contents which are generated every second on social media and people are able to see more comments about a particular product. Customers will not be able to find time for reading all the opinions in a short span of time before buying a product. They like to have a system which shows the pros and cons of a selected product by analyzing huge comments in the same portal with various aspects of the product. Aspect-based sentiment analysis is useful to the customers to know the sentiments of various product aspects. There are various data mining techniques available to perform sentiment analysis, and these techniques are still struggling to provide better accuracy. So the usage of

✉ M. Sivakumar
sivakumarm.cse@krct.ac.in

Srinivasulu Reddy Uyyala
usreddy@nitt.edu

1 Department of Computer Science and Engineering, K. Ramakrishnan College of Technology, Tiruchirappalli 621112, India

2 Machine Learning and Data Analytics Lab, Department of Computer Applications, Centre of Excellence in Artificial Intelligence, National Institute of Technology, Tiruchirappalli 620015, India

artificial intelligence came into picture, in which algorithms are trained to make decision like human beings. There is a huge demand for artificial intelligence (AI) systems using machine learning and deep learning techniques which saves valuable human time for any decision-making tasks. AI systems can also be used to help the customers to know about the pros and cons of any product before buying it. We can create such kind of AI system which can analyze a huge amount of past customer opinions about a particular product and can fulfill the customer expectations at the time of purchase. The AI system would also predict the customer's feelings and suggest the product accordingly. In aspect-based sentiment analysis, the input review contains sequence of features. The proposed work utilizes the sequence of features in aspect-based sentiment analysis. Instead of classifying sentences as positive or negative, the proposed LSTM with fuzzy logic model based on sentimental score generates four labels (highly positive, positive, negative and highly negative) which makes customers decision more easier and helps the customer for choosing right product.

The rest of the paper is organized as follows: Section 2 describes the related work, In Sect. 3 preliminaries of the proposed model was highlighted. The proposed methodology is discussed in Sect. 4. The experimental results are discussed in Sect. 5. Finally, Sect. 6 gives the conclusion and the future enhancement.

2 Related works

Aspect-based sentiment analysis is an important area of study in the current era. To know the sentiments of the product's aspects, artificial intelligence (AI) techniques help consumers for choosing right product. The following section describes various AI techniques used in aspect-based sentiment analysis. Kai Sheng Tai et al. [1] proposed sentiment classification using a Tree-LSTM network. The authors experimented the proposed model on movie reviews (MR) dataset and Stanford Sentiment Treebank (SST) dataset for fine grained (five classes: very negative, negative, neutral, positive, and very positive) and binary classification. The experimental results showed better performance on binary classification than fine grained classification with the accuracy of 88.0% and 51.0%, respectively. There is a wide scope for improvement of accuracy on the multi-label classification. Xi Ouyang et al. [2] developed a framework using Word2vec and convolution neural network (CNN) for sentiment analysis at sentence level on MR dataset. The Word2vec was used to extract features from sentences in vector representation and CNN with 3 convolutional layers and 3 pooling layers to perform fine grained five class (negative, somewhat negative, neutral, positive or somewhat positive) sentiment classification. The accuracy of the proposed framework was 45.4% on classifying sentences with multi-class labels. Karen Howells et al. [3] presented a system for sentiment analysis of consumer comments from social media using fuzzy logic on twitter dataset which categorize the review tweets into strongly positive, positive, neutral, negative and strongly negative. The developed framework did not concentrate on feature extraction, preprocessing and model accuracy. Ying Fang et al. [4] developed a framework for word level sentiment analysis on consumer product review (CPR) dataset using support vector machine (SVM) and enhanced SVM with 82.85% and 86.35% accuracy. Features were extracted using TF-IDF method and constructed sentiment lexicon for performing sentiment analysis. When increasing the vector dimension above 400, the model performance was not improved. Xianghua Fu et al. [5] presented a deep learning model for sentiment analysis with binary (positive and negative) classification using LSTM network. The features were extracted from review sentences using word embedding technique. The proposed model was experimented on publicly available MR dataset, IMDB dataset, Yelp2013 dataset, NB4000 dataset and book4000 dataset with the accuracy of 80.8%, 87.4%, 59.4%, 93.25% and 95.0%, respectively. When word embedding combined with sentiment embedding for sentiment analysis, the developed model becomes more complex. Can et al. [6] developed a multi-lingual sentiment analysis framework with binary classification using recurrent neural network (RNN). Word embedding was used as the feature extraction technique to perform sentiment analysis, and SentiWordNet was used for finding the sentiment score of the review sentences. The framework was experimented on English reviews (ER) dataset and Yelp restaurant review (RR) dataset of four languages datasets with the accuracy of 87.06% and 85.62%, respectively. Aziz et al. [7] created a supervised machine learning system for aspect-based sentiment analysis using support vector machine. Tree similarity index (TSI) calculation and term frequency (TF) approach were used for discovering relationship and feature extraction from the review sentences. They labeled the sentences with word, namely poor, fail, bad and great. The proposed system was experimented on random dataset from five domains with 75% accuracy which need further improvement. Dong et al. [8] developed a model for sentiment analysis using BiLSTM network. The pre-trained GloVe word embedding technique was used to convert the text into word vector with dimension 300. The feature of the sentence was extracted using n-gram convolution layer. BiLSTM with softmax activation function was used for performing deep learning on the output of the model. The developed model experimented on MR dataset, IMDB dataset and SST dataset with the model accuracy of 81.47%, 91.96% and 48.34%, respectively. The proposed model did not focus on the aspect-based sentiment analysis and accuracy of the model on SST dataset was not up to the mark. Zuo et al. [9] created a system for implicit sentiment analysis using context-specific heterogeneous graph convolutional network (CsHCNN). The authors created dependency tree for extracting features, but the system did not consider the edge information. Basiri et al. [10] presented a system for detecting sentiment polarity through target identification. Lexicon-based approach was used to extract features from the sentences. The proposed system adopted five policies such as most occurring first (MOF) with 89% accuracy, most specific first (MSF) with 83% accuracy, first occurring first (FOF) with 80% accuracy and last occurring first (LOF) with 84% accuracy. The proposed system requires labeled large corpus for training. Shams et al. [11] presented a method for aspect-based sentiment analysis using EM algorithm on SemEval2016 laptop product review (LPR) dataset. Aspects were extracted using latent Dirichlet allocation (LDA) method from the sentences. The proposed method classified the dataset according to 15 aspects of laptop product with binary classification with 42.0% precision, 89.0% recall and 57.0% F-measure. Waqar Ali et al. [12] proposed a framework for aspect level sentiment analysis on product reviews by incorporating social Internet of Things using bidirectional gated recurrent unit. Word embedding technique was used to extract features from the review sentences. The proposed framework experimented on Twitter, LPR and restaurant datasets with the accuracy of 74.56%, 77.11% and 82.05%, respectively. The proposed framework detects polarity based on single aspect only. Yong Bie et al. [13] presented a model for aspect-based sentiment analysis by adopting the ideas of multi-view learning and multi-task learning using LSTM, BiGRU and CNN. Word embedding was used for feature extraction from the sentences. The presented model experimented on three LPR, RR and ER benchmark datasets taken from SemEval2014 repository, and model was evaluated based on F1-scores with values 55.08%, 65.20% and 47.89%, respectively. When more than one aspect is present in a sentence, the proposed model suffers on predicting the polarity with conflict. Lin et al. [14] proposed a deep learning model for aspect-based sentiment analysis using BiLSTM network. Word embedding using GloVe was used for extracting features from sentences. The proposed model experimented on three benchmark LPR, RR and Twitter datasets from SemEval-2014 with the accuracy of 83.39%, 80.25% and 75.87%.

2.1 Research gap on the existing literature

From the existing literature, it is found that the location-based sentiment analysis was not carried out and timestamp information was not considered on the sentiment analysis task. It is also noted that when the number of aspects of the product increased, the performance of the proposed framework degrades. To address these issues, the current manuscript formulates aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic.

2.2 Contributions

The contributions of the aspect-based sentiment analysis framework for product reviews are as follows:

- The location-based sentiments of the customers were considered to treat opinions under various geographical locations by analyzing interest of customers
- The current trends were also considered on sentiment analysis, since the interest of customer changes on time to time
- Feature extraction was done using word embedding techniques based on bag of words techniques which will reduce time for searching unused features
- A hybrid model was constructed for classifying sentences based on aspects using LSTM and fuzzy logic, and then, the proposed model was experimented on three benchmark datasets: ACPR, AVGR and CRAP with better accuracy of 96.93%, 83.82%, and 90.92%, respectively.

3 Preliminaries

3.1 Deep learning using LSTM network

Neural network is a brain-stimulated system that might be supposed to copy the way that human beings study, and it consists of input, output and a hidden layer. It is a fantastic tool for locating patterns that can be too complicated for a human. RNN is a class of deep neural networks and also called feedback network which remembers the past output, and decisions are influenced by what it has learned from the past by allowing the use of previous output as input. In Fig. 1, each hidden unit of RNN is replaced with LSTM cell and an additional connection between LSTM cells. It diminishes the exploding gradient and vanishing problem.

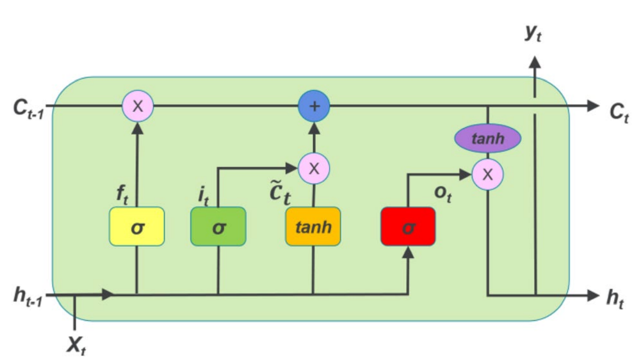

Fig. 1 General LSTM architecture

In this structure, a cell state vector value is maintained at each time step by each LSTM. Explicit gating mechanism is used in the next LSTM to read from the present LSTM. Each LSTM consists of binary gates, namely input gate (i_t), forget gate (f_t) and output gate (o_t).

The input gate controls the memory cell in the updating process, forget cell controls the setting of the memory cell to zero, and the output cell controls the visibility of current cell information. The entire LSTM function was carried out by considering the parameters listed in Table 1.

Forget gate layer forgets the sentences which are not useful for long time. This is achieved by sigmoid activation function given in Eq. (1):

$$f_t = \sigma(W_f.[h_{t-1}, x_t]) + b_f \tag{1}$$

In every iteration, the previous vector is concatenated with the current input vector and then, it is multiplied with the weight vector and added with the bias vector. This gate decides which sentence should be discarded and which should be remembered using the sigmoid activation function. When the value of the forget gate is zero, the sentence will be discarded; otherwise, the sentence will be remembered. Input gate layer decides which portion of the input sentences are to be added with output cell state, and it is calculated using Eq. (2). This layer adds the current input sentence with existing information. For that current cell, state vector is calculated using Eq. (3). The final information to be passed to next LSTM will be available in C_t, which is calculated as per Eq. (4). So, C_t is called long-term memory in LSTM.

$$i_t = \sigma(W_i.[h_{t-1}, x_t]) + b_i \tag{2}$$

$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t]) + b_C \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

In the output gate layer, the output and hidden state values are calculated using Eqs. (5) and (6) which are based on current input and previous hidden state value. This value is called short-term memory.

$$o_t = \sigma(W_o.[h_{t-1}, x_t]) + b_o \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Figure 2 shows how LSTM processes the input review sentences and generates hidden state and cell state vector values. Based on these values, the polarity of the sentence is calculated. LSTM uses two activation sigmoid activation functions depicted in Eq. (7). This activation function is applied on all the three gates which help the node to generate output based on the given inputs.

Sigmoid Function:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \tag{7}$$

In the current study, aspect-wise sentence polarity score was calculated by LSTM (Fig. 5) and the fuzzy logic system generates multiple labels, namely highly positive (HP), positive (P), negative (N) and highly negative (HN). The proposed system was trained and tested using three benchmark datasets.

Table 1 Parameters of LSTM network

Parameter	Description
W_f	Weight vector of the forget gate layer
h_{t-1}	Previous hidden state vector
h_t	Output hidden state vector
x_t	Current input vector
b_f	Bias vector
i_t	Current input vector
o_t	Output vector
C_t	Output cell memory vector
C_{t-1}	Output cell memory vector
$\tilde{C}_t$	Current cell memory vector

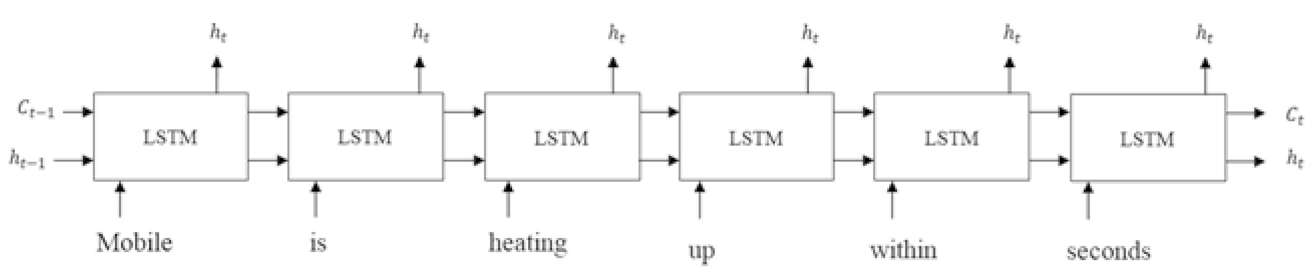

Fig. 2 LSTM processing a review sentence

3.2 TF-IDF

Each review paragraphs are considered as a document and that can be tabulated as term-document matrix. Each entry in the matrix is pointing to the frequency of a term in the given document. If the value of the term-document matrix is zero, the term does not have presence in the review paragraph. Term weighting allows a system to detect the dominance of any word in the paragraph. Term weighting assigns numerical values to terms that represent their dominance in the review with a purpose to improve retrieval effectiveness. Term frequency (tf) is the fraction between frequency (*f*) of a particular term and highest frequency value of a term (Eq. 8):

$$tf(t) = \frac{f_{ij}}{\max_i \left(f_{ij}\right)} \tag{8}$$

Inverse document frequency (idf) is the logarithmic ratio of the number of review paragraphs or documents in the page related to particular topic of interest to total the number of review documents the specific term or token present (Eq. 9).

$$idf_i = \log_2 \left(\frac{N}{df_i}\right) \tag{9}$$

When the text corpus is large, TF-IDF will generate a high-dimensional feature vector, which could potentially increase the over-fitting problem in the classification model. Owing to this, the model accuracy will be reduced, and also this method does not consider the semantic feature of the text. To address these issues, the authors adopted the word embedding feature extraction technique [12].

4 Proposed methodology

The proposed aspect-based sentiment analysis using long short-term memory and fuzzy logic (Fig. 3) generates class label for various aspects. The pre-processed product review sentences are fed to the LSTM-based deep learning network and generates positive and negative sentiment score. Then, these sentences along with generated sentiment score were sent to the fuzzy logic system. The fuzzy logic system processes these values and generates class label based on rule inference. There are four class labels, namely highly negative (HN), negative (N), positive (P) and highly positive (HP) which are generated based on membership function by the fuzzy logic system. By changing the input aspect set, the system can be used on any product review dataset to perform aspect-based sentiment analysis effectively. Eight aspects were identified for the mobile phone such as battery-life, performance, display, camera, storage, design, value-for-money and software. Though there are systems that carry out sentiment analysis under document level, sentence level, and word level, the proposed system detects sentiments for each sentence based on the specifies aspects of the product. This will make the customers easy to analyze the product pros and cons quickly before buying a product and also helps the manufactures to improve the product features which in turn improve the sales of the product. The proposed LSTM and fuzzy logic model was experimented on ACPR dataset, AVGR dataset and CRAP dataset. Initially, datasets were grouped according to the geographical location. The proposed aspect-based sentiment analysis model using LSTM and fuzzy logic was trained and tested separately for every dataset based on country. The proposed system can generate results according to geographical location. Recent three years records were only considered for training and testing the developed system. So, the current trends can also be considered for making decision on input customer review. The detailed process of the proposed framework (Fig. 3) is discussed in the following subsections.

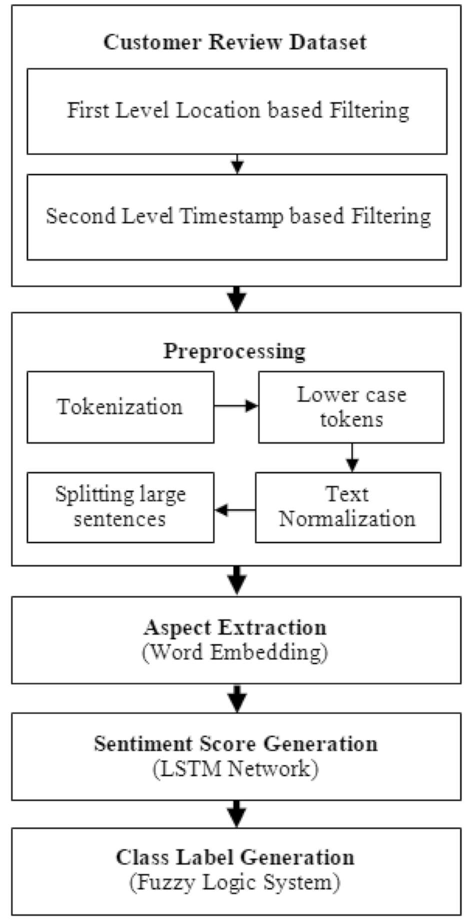

Fig. 3 Proposed framework using LSTM and fuzzy logic system

Algorithm Proposed Deep Learning Model

Result: Labeled Sentences based on Aspect

Step-1: Pre-processing

while end of the review paragraph **do**

- Tokenization
- Lowercase the tokens
- Spell checking
- Lemmatization

end{while}

- Split long sentence

while end of the review **do**
split the long sentence using ClausIE framework
end{while}

Step-2: Aspect Extraction using Word Embedding

Step-3: Calculate Polarity Score using LSTM

Step-4: Fuzzification

- Assign value for membership function using intuition
- Construct Fuzzy Set based on polarity score
- Use Linguistic variables {HN,N,P,HP}

Step-5: Decision Making using Rule Inference

Step-6: Deffuzzification using *Centre of Gravity (CoG)* method

Step-7: Clustering based on Aspects

Step-8: Calculate Accuracy using *precision, recall and F-score* measures

4.1 Pre-processing

The dataset contains reviewer-id, category, reviewer-name, review-text, rating, summary, and time. During pre-processing, the unwanted non-opinionated sentences are eliminated with the help of a dictionary by labeling subjective sentences and the unlabeled sentences would be treated as subjective sentences which will be discarded as non-opinionated sentences. Text normalization consists of various tasks such as statistical machine translation, dictionary mapping and spelling correction. In the proposed work, the preprocessing components such as tokenization, converting all text to lowercase, spelling correction, and lemmatization using dictionary mapping were applied to the product review dataset. Tokenization splits the given review paragraphs into words. Then, the words were converted into lower case. Spelling correction was carried out on tokens, and lemmatization converts the tokens into its standard form. The detailed preprocessing operations are as follows:

4.1.1 Sample AVGR dataset

"I bought this and the key didn’t work. It was a gift, and the recipient wasn’t able to solve the problem. It might have been a good game, but I never found out because the key failed."

4.1.2 Tokenization

‘I’, ‘bought’, ‘this’, ‘and’, ‘the’, ‘key’, ‘did’, "n’t", ‘work’, ‘It’, ‘was’, ‘a’, ‘gift’, ‘and’, ‘the’, ‘recipient’, ‘was’, "n’t", ‘able’, ‘to’, ‘solve’, ‘the’, ‘problem’, ‘It’, ‘might’, ‘have’, ‘been’, ‘a’, ‘good’, ‘game’, ‘but’, ‘I’, ‘never’, ‘found’, ‘out’, ‘because’, ‘the’, ‘key’.

4.1.3 Convert tokens into lowercase

‘i’, ‘bought’, ‘this’, ‘and’, ‘the’, ‘key’, ‘did’, “n’t”, ‘work’, ‘it’, ‘was’, ‘a’, ‘giffft’, ‘and’, ‘the’, ‘recipient’, ‘was’, “n’t”, ‘able’, ‘to’, ‘solve’, ‘the’, ‘probelem’, ‘it’, ‘might’, ‘have’, ‘been’, ‘a’, ‘good’, ‘game’, ‘but’, ‘i’, ‘never’, ‘found’, ‘out’, ‘because’, ‘the’, ‘key’.

4.1.4 Text normalization: spelling correction

‘i’, ‘bought’, ‘this’, ‘and’, ‘the’, ‘key’, ‘did’, ‘not’, ‘work’, ‘it’, ‘was’, ‘a’, ‘gift’, ‘and’, ‘the’, ‘recipient’, ‘was’, ‘not’, ’able’, ‘to’, ‘solve’, ‘the’, ‘problem’, ‘it’, ‘might’, ‘have’, ‘been’, ‘a’, ‘good’, ‘game’, ‘but’, ‘i’, ‘never’, ‘found’, ‘out’, ‘because’, ‘the’, ‘key’.

4.1.5 Text normalization: lemmatization (dictionary mapping)

‘i’, ‘**buy**’, ‘this’, ‘and’, ‘the’, ‘key’, ‘do’, ‘not’, ‘work’, ‘it’, ‘be’, ‘a’, ‘gift’, ‘and’, ‘the’, ‘recipient’, ‘be’, ‘not’, ‘able’, ‘to’, ‘solve’, ‘the’, ‘problem’, ‘it’, ‘might’, ‘have’, ‘be’, ‘a’, ‘good’, ‘game’, ‘but’, ‘i’, ‘never’, ‘**find**’, ‘out’, ‘because’, ‘the’, ’key’.

4.1.6 Splitting long sentences

The current student mainly focuses on splitting large sentence connected by conjunction into meaningful sentences using ClausIE framework [13]. The detailed working procedure of the ClausIE framework is described in Fig. 4.

Let CR_i be the input customer review from www.amazon.in.

CR_i = {"Brilliant camera, huge battery life and brilliant display in addition with the premium feeling of SAMSUNG. It's a shame that this phone's launch was delayed due to lockdown."}.

After preprocessing, the given input review sentence [14] was split into the following five sentences as follows:

CR_{i1} = {"Phone has Brilliant camera"}.
CR_{i2} = {"Phone has huge battery life"}.
CR_{i3} = {"Phone has brilliant display"}.
CR_{i4} = {"Phone has premium feeling of SAMSUNG"}.
CR_{i5} = {"Phone's launch was delayed due to lockdown"}.

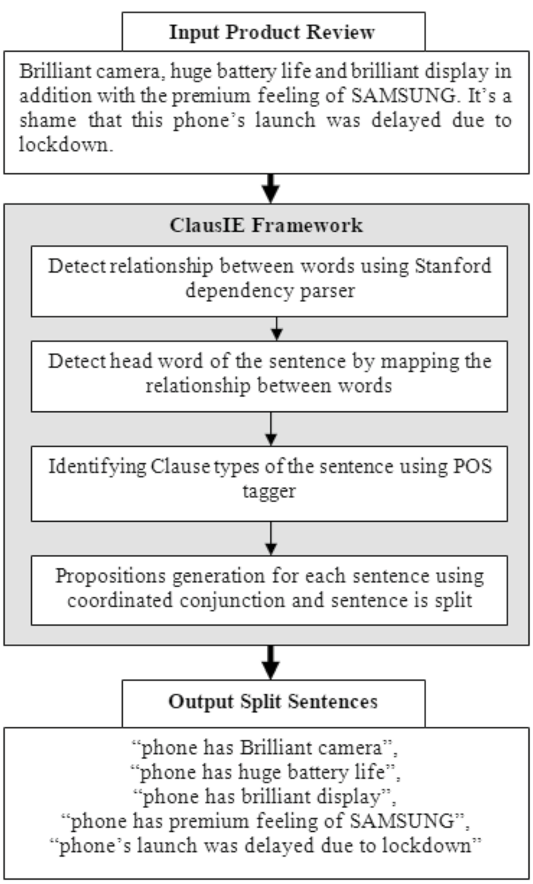

Fig. 4 ClausIE framework for long sentence splitting

4.2 Feature extraction

Feature extraction [15] from the review text becomes a crucial task of sentiment analysis. Bag-of-Words, TF-IDF [16], and Word Embedding [17] are the most popular feature extraction techniques used in literature. Bag of words are well suitable for document classification in which each word is considered as a feature and does not maintain the order of the features. The order of the features is important to retain the semantic nature of the sentences in aspect-based sentiment analysis. Hence, bag of words are not a suitable approach for feature extraction. Even though TF-IDF performs better than bag of words for the feature extraction, when text corpus is large, TF-IDF will generate a high-dimensional feature vector, which could potentially increase the over-fitting problem in the classification model. Owing to this, the model accuracy will be reduced, and also this method does not consider the semantic nature of the review sentences. The current work focuses on aspect-based sentiment analysis in which the order and semantic nature of the review sentence are important to extract the viable features. The word embedding feature extraction technique is a well-known method. It maintains the order and as well as the semantic nature of the review sentences. Hence, the word embedding technique is a suitable method for extracting the features in aspect-based sentiment analysis.

4.3 Word embedding

Word embedding [18] is a vector space model for representing words as real-valued vectors for finding the relationship between a word and its synonymous words. In this work, this approach is used to extract aspect in sentences. Word embedding works with two methods which includes the Continuous Bag of Words (CBOW) as well as Skip-gram technique. We used the CBOW technique by which a word or context or feature is predicted. It is performed using three-layer neural network structures where the first layer represents input words, the second layer represents the projection layer, and the third layer represents the output layer.

Let S be the input words in a sentence represented as follows (Eq. 10):

$$S = \{W_{i-2} + W_{i-1} + W_{i+1} + W_{i+2}\} \tag{10}$$

where W_i is the feature word to be predicted.

Let V be word size in a sentence and N refers to the size of the hidden layer. The weight matrix (W_t) is calculated as follows (Eq. 11):

$$W_t = V * N \tag{11}$$

Weighted sum (WS) is calculated as follows (Eq. 12):

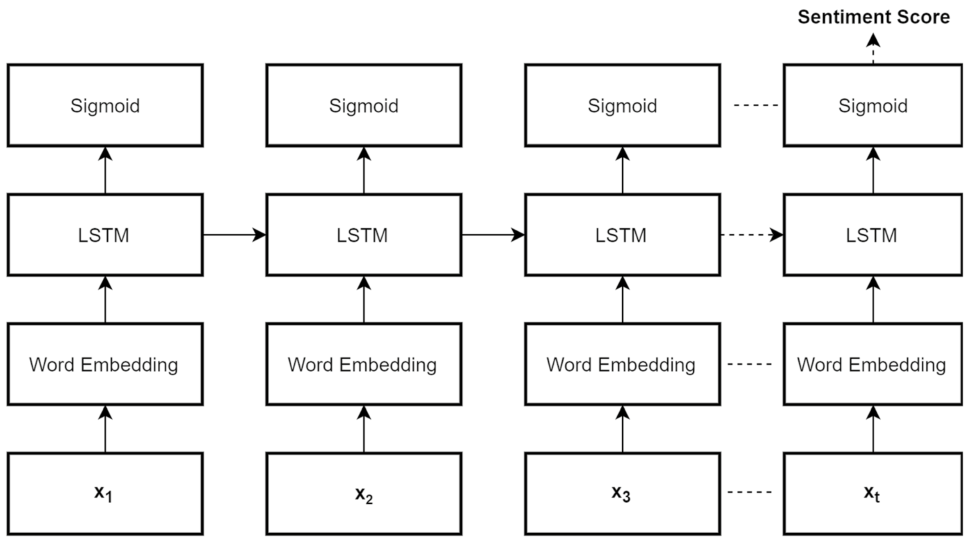

Fig.5 Sentence sentiment score generation by LSTM

$$WS = X_i * W_t \tag{12}$$

where X_i represents the words in LSTM input layer and *Wt* represents the weight of the LSTM.

Let PR be the target feature vector representation. The relationship (R) between the feature and a word is calculated as follows (Eq. 13):

$$R = PR * WS \tag{13}$$

Based on *R* value, each sentence in a review is labeled with a feature of a product and then it is sent to recurrent neural network to generate sentiment score. This score is compared with target polarity score and error is calculated; this error value is send back to the previous layer for updating its weights.

4.4 Sentiment score generation

Rule-based method and lexicon-based approaches are dominantly used by most of the sentiment analysis work. In our work, we used lexicon-based approach to detect polarity of a sentence. In the lexicon-based approach, a bag of words with sentimental words are used to check the presence of the sentimental word using word embedding in a sentence in the LSTM. Figure 5 shows sentence polarity score generation for each input review sentences by applying sigmoid activation function in the hidden layer and output layer.

4.5 Sentiment labeling using fuzzy logic system

LSTM generates crisp output as polarity score for each input sentences, and it is listed in Table 2. LSTM was continued, and it is learning till the expected sentiment score generated.

This score was given to the fuzzy logic system as input (Fig. 6), and a fuzzy set was created to classify sentence with four categories. There were sentences which could not be classified completely as positive or negative, and these sentences might impact for making decision. So, those sentences will be labeled with positive or negative, and other

Table 2 Sentiment score generated by LSTM

Sample sentence	Negative	Neutral	Positive
CR_{i1}	0.12	0.04	0.89
CR_{i2}	0.11	0.03	0.82
CR_{i3}	0.12	0.04	0.89
CR_{i4}	0.23	0.31	0.78
CR_{i5}	0.19	0.32	0.79

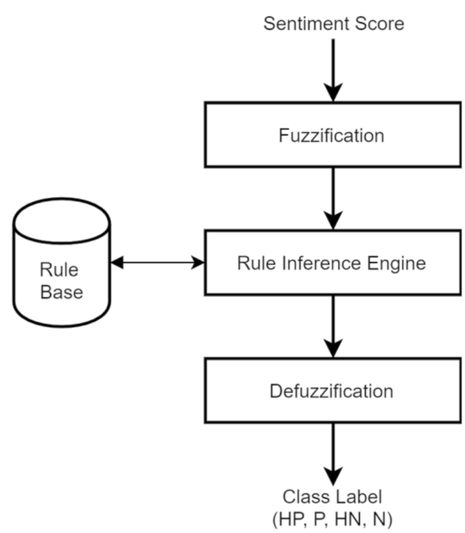

Fig.6 Class label generated by fuzzy logic system based on sentiment score

sentences are labeled with either highly positive or highly negative.

Fuzzification module converts the output sentiment score values of each word into fuzzy set, whereas each fuzzy set was represented using pair of member function and linguistic variables. Generally, a normalized fuzzy set Ã review in the domain of discourse $X_{\text{review-set}}$ was defined as a set of ordered pairs:

$$\tilde{A} = \left\{ \left(score, \mu_{\tilde{A}}(score) \right) | x \in X \right\} \tag{14}$$

where μ_A(word) is called membership function of A. The membership value ranges in the interval [0, 1].

Let the resultant fuzzy set for a input review sentence be

$$\tilde{S} = \left\{ \left(score, \mu_{\tilde{A}}(score) \right) | x \in X \right\} \tag{15}$$

where $X = \{\text{HN, N, P, HP}\}$ and membership function (μ_A) defines a fuzzy set ($\tilde{A}$), which are discrete or continuous and solve complex problems with experience rather than knowledge. Though there are various methods for assigning values of membership function, we used intuition method in our proposed work. Decision making is done by referring set of rules in the rule base to categorize the sentences based on threshold values into HN, N, P and HP classes. Defuzzification is a process of converting resultant fuzzy set into crisp set of values where these values can be processed by other systems. There are various approaches employed for converting fuzzy set values into crisp values. In our work, we used center of gravity (CoG) method for defuzzification.

5 Experimental results and discussions

5.1 Description of the dataset

The proposed LSTM with fuzzy logic model was evaluated using the following three publicly available benchmark datasets, and sample records are shown in Table 3.

5.1.1 Amazon Cell Phones Review (ACPR)

Amazon Cell Phones Reviews dataset was downloaded from https://www.kaggle.com/ [19]. The dataset contains reviews about ASUS, Apple, Google, HUAWEI, Motorola, Nokia, OnePlus, Samsung, Sony and Xiaomi cell phone products. The 'body' column contains the review paragraph, and 'date' column contains the review posted date. The Amazon Cell Phones Reviews dataset contains 67,986 records posted, and 61,225 records were labeled with verified = "TRUE".

5.1.2 Amazon Video Games Review (AVGR)

Amazon Video Games Review data were taken from https://jmcauley.ucsd.edu/ [20]. The dataset contains reviews about various video games product. It contains 231,780 reviews records in JSON file format. The 'ReviewText' column contains the customer reviews, and 'reviewTime' is having review posted time.

Table 3 Sample records from three benchmark datasets

Data Set	ID	Sample Review Text
Amazon cell phone review dataset	AVpf2cQm1cnluZ0-sb5y	Since the details for the items are a little sparse I thought I would provide the specs listed on the power adapter.AC Input 100–240 Volts, 0.15 Amps, 50/60 HzDC Output 4.9 Volts, 0.85 Amps. This adapter has a Flextronics part number of 09500043-200, Model A00810
	AVpf2cQm1cnluZ0-sb5y	I was initially happy that my Kindle replacement charging cord arrived so promptly, it seemed to work fine. But after only one use, the outer covering of the cord started cracking falling off in bits. It now is almost completely covered by electrical tape, as the bits fall off w/ each every use. I will end up having to buy a new one after all
Amazon video games review	A30TL5EWN6DFXT	They look good and stick good! I just don't like the rounded shape because I was always bumping it and Siri kept popping up and it was irritating. I just won't buy a product like this again
	ASY55RVNIL0UD	These stickers work like the review says they do. They stick on great and they stay on the phone. They are super stylish and I can share them with my sister
Amazon consumer product review	A248LSBZT4P38V	I bought this and the key didn't work. It was a gift, and the recipient wasn't able to solve the problem. It might have been a good game, but I never found out because the key failed
	AFS6WERAP409A	Crashed in Vista. Codemasters told me they don't support it in Windows 8. Couldn't get it to work even after looking on the Internet

5.1.3 Consumer Review of Amazon Products (CRAP)

CRAP dataset was collected from https://www.kaggle.com/ [21]. It contains over 34,000 consumer reviews for Amazon products like the Kindle, Fire TV Stick and more provided by Datafiniti's Product Database. The dataset includes basic product information, review text, rating and more for each product.

5.2 Experimental setup

The proposed model experimented on Intel core i5 7$^{\text{th}}$ Gen GPU processor and 8GB RAM, Windows 10 operating system. The proposed model was trained and tested using Anaconda framework with python 3.8 version. The Jupyter module of the anaconda framework was used with WordNetLemmatizer, wordtokenize, and NumPy packages for preprocessing. Word2Vec, pandas, and PCA libraries were used for feature extraction. The proposed LSTM and fuzzy logic model was constructed using the Keras library.

5.3 Proposed LSTM and FL model hyper parameters

Different hyperparameters are adjusted for performance improvement in the proposed LSTM and fuzzy logic model. Hyperparameters impact the proposed model learning, accuracy and stability.

5.3.1 LSTM layers

The word embedding converts input review text into word vector. The proposed model performance was analyzed by training the input embedding layer sizes with 64 and 128. The proposed model gives better accuracy for 64 length vector input embedding layer on three benchmark datasets. Memory units are used to remember the words from input review sentence. The proposed LSTM was designed with 100 memory units that was sui of remembering words for understanding a lengthy review paragraph. Three nodes were chosen for output layer to generate sentiment scores (positive, negative and neutral classes).

5.3.2 Activation function

Aspect-based sentiments analysis can be viewed as multi-label classification problem. Sigmoid activation function was suitable for handling multi-label classification. For multi-label classification problem, the sum of the sentiment score of all three nodes in the output layer should not be equal to one. Hence, the proposed model was designed with sigmoid activation function.

5.3.3 Number of epochs

Number of iterations for the LSTM network on input dataset was decided by the number of epochs. Root mean square error (RMSE) was calculated on each epoch. The proposed model was trained by varying number of epochs (100, 200, 300, 400 and 500) and checked the performance of the RMSE. The proposed model obtained good accuracy when the number of epochs was 500.

5.3.4 Batch Size

Batch size determines the number of input review sentences to be processed by the LSTM network. In the proposed model, LSTM was trained with batch size of 4, 8, 16 and 32 on three benchmark datasets. The accuracy of the proposed model was good at the batch size of 4.

5.3.5 Optimizer

The default Adam optimizer was used for LSTM network to tune the learning rate. The accuracy of the proposed model was improved by adopting Adam optimizer.

5.3.6 Dropout

To avoid the over-fitting problem in LSTM network while training the proposed model on review sentences, two dropout layers were attached in the network. One dropout layer was added between embedding layer and LSTM layer. The second dropout layer was added between dense output layer and LSTM layer in the proposed model.

5.4 Overall classification accuracy of the proposed model

The proposed model was evaluated by using the following standard statistical measures. Precision, recall and F-score were calculated by using Eqs. (16, 17 and 18), respectively. Accuracy was computed by using Eq. (19):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

$$F\text{ - score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

where TP (true positives) is the number positive review sentences correctly classified as positive, TN (true negatives) is the number of negative review sentences correctly classified as negative, FP (false positives) is the number of negative review sentences wrongly classified as positive, and FN (false negative) is the number of positive review sentences wrongly classified as negative.

The performance of the proposed model was evaluated on three publicly available benchmark datasets by considering the standard statistical measures. The experimental results were compared with well-known state-of-the-art methods (SVM, LSTM, BiLSTM, CNN, Tree-LSTM, and RNN), and it is reported in Table 4. From the results, we can observe that the proposed model gives better accuracy of 96.93% on ACPR dataset and 83.83% on AVGR dataset when compared with the existing models, whereas for CRAP dataset, the proposed model reported 88.13% of precision, 95.33% of recall, 82.21% of F-score and 90.92% of accuracy. When compared with the existing models, the proposed model had 2.34% lesser precision and 6.63% lesser F-score than the existing RNN model, 1.77% lesser recall and 0.28% lesser accuracy than the existing BiLSTM model. The predominant values were highlighted in bold (Table 4). To get a better performance, one should create a separate model for constructing an aspect corpus by considering aspects of all products.

The performance of the proposed model was also experimented without word embedding feature extraction method on three benchmark datasets and obtained the accuracy of 63.68%, 68.15% and 77.56%, respectively. The results were compared with the proposed model and are reported in Fig. 7. From the results, we can observe that the word embedding feature extraction method has higher accuracy of 33.25%, 15.67% and 13.36%, respectively, on three

Table 4 Experimental results comparison of proposed model with state-of-the-art methods

Method	Amazon cell phones reviews (ACPR)				Amazon video games review (AVGR)				Consumer review of Amazon products (CRAP)			
	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score	Accuracy
SVM	64.04	78.88	80.06	75.93	89.72	90.34	76.38	76.97	89.47	93.11	85.74	88.88
LSTM	90.01	93.72	94.07	91.39	86.80	85.47	91.86	74.44	88.41	86.54	85.90	89.01
BiLSTM	13.75	82.13	82.89	79.43	86.73	83.79	82.12	74.66	83.80	84.01	82.92	90.33
CNN	46.84	84.54	85.91	81.87	91.98	93.29	85.06	77.53	89.76	97.06	86.60	90.94
Tree-LSTM	57.61	79.86	82.73	79.63	91.23	93.34	85.74	78.11	89.90	97.20	88.74	91.20
RNN	59.01	70.29	71.56	67.85	90.78	94.12	70.73	79.91	**90.47**	**96.47**	**88.84**	88.14
LSTM + FL	**95.33**	**95.29**	**94.99**	**96.93**	**93.66**	**95.29**	**92.12**	**83.82**	88.13	95.33	82.21	**90.92**

The predominant values were highlighted in bold

Fig. 7 Performance comparison between without word embedding and with word embedding

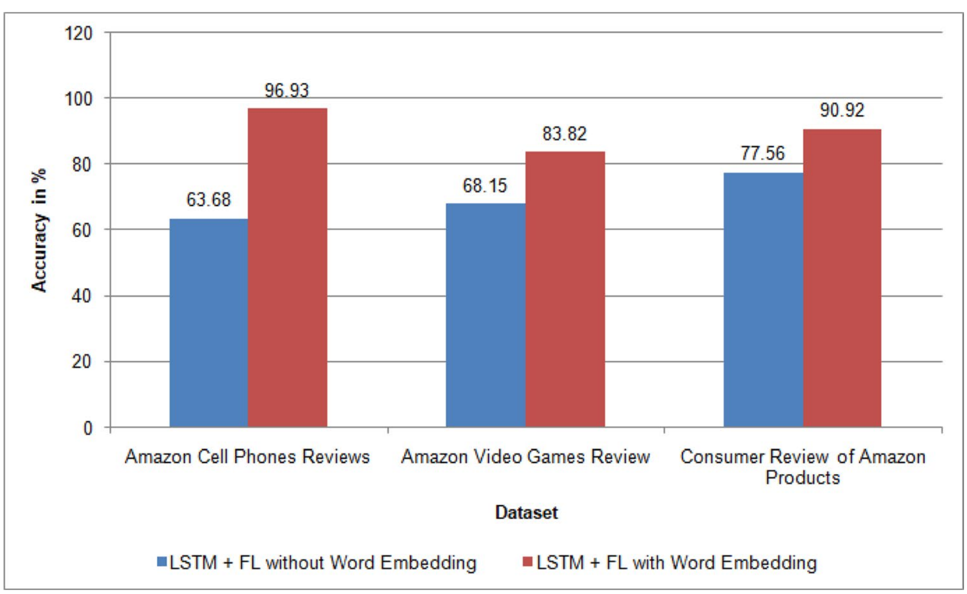

Table 5 Comparison of the proposed LSTM + FL hybrid method with SVM, LSTM, BiLSTM, CNN, Tree-LSTM, and RNN for location and trends

Approach	Accuracy (%)	Location	Trend
Tree-LSTM [1]	88.00	No	No
CNN [2]	45.40	No	No
SVM [7]	75.00	No	No
LSTM [5]	95.00	No	No
RNN [6]	87.06	No	No
BiLSTM [8]	91.96	No	No
LSTM + FL [ours]	**96.93**	**Yes**	**Yes**

The predominant values were highlighted in bold

benchmark datasets than without word embedding feature extraction method. Hence, the word embedding technique is well suitable for the aspect-based sentiment analysis.

The location-based sentiment analysis with timestamp information aids the customer to get the sentiments of a product with respect to the current trends and geographical locations. The proposed LSTM and fuzzy logic model was experimented on the ACPR, AVGR and CPAP datasets. Customer review records were grouped based on geographical locations. The proposed model was experimented separately for every country, and results can be generated based on locations. The proposed model was designed to consider recent three years customer review records to make decisions on current trends.

The proposed model and existing models were experimented on three benchmark datasets, and accuracy, location and trends are reported in Table 5. The proposed model was able to filter the customer product reviews according to the geographical locations and trends with high accuracy when compared to the existing models and the same was highlighted in Table 5.

6 Conclusion

The proposed model for aspect-based sentiment analysis using LSTM with FL adopts the features of ClausIE framework for splitting long sentences into meaningful small sentences. The performance of the developed model was experimented with and without word embedding technique for feature extraction. From the results, we observed that the word embedding technique was well suited for aspect-based sentiment analysis. Instead of classifying consumer review sentences as positive and negative, the proposed LSTM with fuzzy logic model classified consumer product review sentences as highly negative, negative, positive and highly positive. The developed model was experimented on three publicly available datasets with 96.93% accuracy on ACPR, 83.82% accuracy on AVGR and 90.92% of accuracy on CRAP datasets. The proposed model also classified consumer review sentences according to the location of consumers and the current trends. The proposed model can be further extended for aspect-based sentiment analysis with a highly complex aspect corpus on live datasets from various online shopping portals.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Pranjal, S.: Bogawar: soft computing approaches to classification of emails for sentiment analysis. In: ICIA-16: Proceedings of the International Conference on Informatics and Analytics, pp. 1–7 (2016)
2. Karen, H.: Applying fuzzy logic for sentiment analysis of social media network data in marketing. In: 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, pp. 24–25 (2017)
3. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: International Joint Conference on Natural Language Processing, vol. 1, pp. 1556–1566 (2015)
4. Ouyang, X., Zhou, P., Li, C.H., Liu, L.: Sentiment analysis using convolutional neural network. In: IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 2359–2364 (2015)
5. Karen, H., Ahmet E.: Applying fuzzy logic for sentiment analysis of social media net-work data in marketing. In: International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, pp. 664–670 (2017)
6. Fang, Y., Tan, H., Zhang, J.: Multi-strategy sentiment analysis of consumer reviews based on semantic fuzziness. IEEE Access **6**, 20625–20631 (2018)
7. Fu, X., Yang, J., Li, J., Fang, M., Wang, H.: Lexicon-enhanced LSTM with attention for general sentiment analysis. IEEE Access **6**, 71884–71891 (2018)
8. Can, E.F., Ezen-Can, A., Can, F.: Multilingual sentiment analysis: an RNN-based framework for limited data. In: ACM SIGIR 2018 Workshop on Learning from Limited or Noisy Data, vol. abs/1806.04511 (2018)
9. Aziz, A., Starkey, A.: Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches. In: IEEE Access, vol. 8, pp. 17722–17733 (2020)
10. Dong, Y., Fu, Y., Wang, L., Chen, Y., Dong, Y., Li, J.: A sentiment analysis method of capsule network based on BiLSTM. IEEE Access **8**, 37014–37020 (2020)
11. Zuo, E., Zhao, H., Chen, B., Chen, Q.: Context-specific heterogeneous graph convolutional net-work for implicit sentiment analysis. IEEE Access **8**, 37967–37975 (2020)
12. Basiri, M.E., et al.: Improving sentiment polarity detection through target identification. IEEE Trans. Comput. Soc. Syst. **7**(1), 113–128 (2020)

13. Shams, M., Khoshavi, N., Baraani-Dastjerdi, A.: LISA: language-independent method for aspect-based sentiment analysis. IEEE Access **8**, 31034–31044 (2020)
14. Del Corro, L., Gemulla, R.: ClausIE: clause-based open information extraction. In: Proceedings of the 22nd international conference on World Wide Web, pp. 355–366 (2013)
15. Bie, Y., Yang, Y.: A multitask multiview neural network for end-to-end aspect-based sentiment analysis. Big Data Min. Anal. **4**(3), 195–207 (2021)
16. Dzisevič, R., Šešok, D.: Text classification using different feature extraction approaches. In: Open Conference of Electrical, Electronic and Information Sciences (eStream), pp. 1–4 (2019)
17. Zhang, W., Xu, W., Chen, G., Guo, J: A feature extraction method based on word embedding for word similarity computing. In: NLPCC 2014, CCIS 496, pp. 160–167 (2014)
18. Ali, W., Yang, Y., Qiu, X., Ke, Y., Wang, Y.: Aspect-level sentiment analysis based on bidirectional-GRU in SIoT. IEEE Access **9**, 69938–69950 (2021)
19. https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews
20. https://jmcauley.ucsd.edu/data/amazon/
21. https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products