



OpExHAN: opinion extraction using hierarchical attention network from unstructured reviews

Ankur Ratmele¹ · Ramesh Thakur²

Received: 30 December 2021 / Revised: 12 September 2022 / Accepted: 14 September 2022 / Published online: 5 October 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2022

Abstract

In recent decades, sellers and merchants have asked their customers to share their opinion on the products at online marketplace. Analysis of the massive amounts of reviews for a potential customer is an immense challenge, to decide whether to purchase a product or not. In this paper, a hierarchical attention network-based framework is presented to resolve this challenge. In this proposed framework, the Amazon's Smartphone review dataset is preprocessed using NLP approaches and then applied Glove embedding to extract word vector representation of reviews which identifies contextual information of words. These word vectors are fed into the hierarchical attention network, which produces vectors at the word and sentence levels. Bi-GRU model encodes the words and sentences into hidden vectors. Finally, reviews are classified into five opinion classes like extremely positive, positive, extremely negative, negative and neutral. Furthermore, to perform the experiments with the proposed method, the dataset is divided into three parts such as 80% train, 10% validation and 10% test. Experiments reveal that the proposed framework outperforms baseline methods in terms of accuracy, precision and recall. The OpExHAN model achieved admirable results like 94.6% accuracy, 91% both precision and recall after a lot of hyper-parameter experimentation.

Keywords Opinion extraction (OE) · Deep learning (DL) · Natural language processing (NLP) · Glove · Gated recurrent network (GRU) · Hierarchical attention network (HAN)

1 Introduction

Opinion Extraction (OE) is a sort of information retrieval that facilitates in generating the proper opinion about a product by observing web data and analyzing the reviews. Sentiment analysis is another term for this type of analysis, which is a branch of natural language processing. It is commonly used to assess public opinion or emotions toward a subject or product, so that marketers may determine whether the public's attitude is favorable or unfavorable (Tan et al. 2019; Haque et al. 2018). This can assist them in improving their flaws and altering their techniques in order to achieve the best specific results. By understanding why a given product

achieved high ratings and what makes that product outstanding, any company or manufacturer can capitalize on this fact and create new things with similar attributes.

Opinion extraction, according to Pang and Lee (2008), is concerned with determining the orientation of opinion in a piece of text with relation to a specific topic. Opinion extraction is made up of three parts, including (1) Opinion Holder: It is a person who offers an opinion. (2) Opinion Object: The subject of the opinion; (3) Opinion Orientation: Determine whether an object's polarity is positive, neutral or negative (Qiu et al. 2011; Turney 2002). For example, "the camera quality of the Samsung mobile is superb." The user who has posted this review is the opinion holder in this review statement. The opinion object is the Samsung mobile's "camera quality," and the opinion term is "superb," which has a positive orientation.

The primary task of opinion extraction is identifying the semantic orientation of text at different levels. It takes place on three levels (Liu 2012): (1) document level: the entire document is classified as positive, neutral or negative; (2) sentence level: the entire sentence is evaluated as

✉ Ankur Ratmele
ankur22ratmele@gmail.com
Ramesh Thakur
r_thakur@rediffmail.com

¹ IET-DAVV Indore, Indore, MP, India

² IIPS-DAVV Indore (M.P.), Indore, MP, India

positive, neutral or negative; (3) aspect level/feature level: at this level, each feature present in the document/sentence is classified as positive, negative or neutral (Ratmele and Thakur 2020).

Different ways have been proposed, which depends on how rules were obtained to analyze textual data. Rule-based, statistical, and hybrid approaches (Wu et al. 2018) are the most commonly applied methodologies in opinion extraction. In this research work, statistical approach-based deep learning and HAN models are used to extract the accurate opinion from Amazon's Smartphone's reviews.

1.1 Need

Opinion extraction is needed in a variety of applications, including recommendation systems, search engines, web ad filtering, Email filtering and question–answering systems. It is convenient in everyday life because it can improve the interaction between human and computer, government intelligence, business intelligence and citation analysis. For opinion extraction from small amount of data, machine learning can be used but for large amount of data, deep learning produces good results. In this research work, a large dataset of 56,000 Amazon's Smartphone's reviews is used to learn the machine for extraction of opinions. Therefore, deep learning is suitable for this large amount of data.

1.2 Motivation

In the digital world, e-commerce platforms allow millions or billions of people to buy a variety of things. On e-commerce platforms, customers can openly submit their thoughts and reviews of specific products. These websites are regarded as a great source of information for determining consumer opinions and estimating product quality based on their opinions. Through the rapid development of online platforms, there are huge amount of reviews in terms of opinions. It is a very tedious task to interpret the general opinion of consumers on a particular product. Other customers can use this information to make decisions, company executives may improve their products, and governments can improve the product quality standards (Chen and Xie 2008; Abrahams et al. 2013). For example, lockdown was imposed across the country as a result of the COVID-19 pandemic. People were locked to their homes and were unable to purchase any products outside of their homes. Online portals were used to purchase goods and essentials at that time. If people want to purchase the mobile phone from online sites and they want to know about the mobile phone's quality based on the available reviews. However, there are numerous reviews available. Manually processing and evaluating the massive volume of review data is a difficult task. This motivates the issue of opinion extraction, which involves identifying the

polarity of reviews based on whether they are best, worst, or neutral. The requirement of interpreting statements provided in an unstructured form has increased the demand for opinion prediction. To extract the correct opinions from a vast number of reviews, deep learning and a semantic-based classification are used.

The work's main focus is on deciphering the opinions conveyed in Amazon's product reviews that are written in an unstructured format. The benefits of polarity identification allow users to classify their reviews into five classes such as neutral, positive, extremely positive, negative and extremely negative.

1.3 Challenges

Opinions are frequently written informally and are extremely different, so that facing a number of challenges (Balahur et al. 2012). Some challenges are discussed as follows-

1. *False review detection* Sometimes, some people post the reviews on product even if they did not buy the product. Reviews given by those people are false and it is difficult to identify the false review.
2. *Different writing style* Individual writing styles fluctuate, and a sentence might be written in a variety of ways. Both positive and negative characteristics can be found in an opinion. As a result, analyzing an opinion at the phrase level, which incorporates both positive and negative feelings, is challenging.
3. *Implication of polarity* If the implications of examining the true polarity of an individual word in relation to its context are neglected, the overall accuracy of the polarity recognition process may suffer. For ex-, in the review "Camera size of this mobile is small," the word small is used in a positive sense, while in "The battery duration is similarly small," here the term small is used in a negative sense (Dhokrat et al. 2015). One drawback of the opinion extraction was that it determines the polarities of words individually without taking into consideration the polarity assigned to surrounding words. In fact, the failure to use the meaning of opinions offered in the target context frequently resulted in poor performance.
4. *Effects of syntax on semantics* Review written in casual form even it is not written in syntactically correct manner. Structure of reviews, contextual opinions and heterogeneous reviews may present major challenges.
5. *Limitation of lexicon-based methods* Context-based words must be handled carefully in their true context for efficient opinion extraction. Lexicons-based opinion extraction methods are ineffective when dealing with context-dependent terms.

1.4 Problem statement

In general, people express their thoughts in a casual way; it might be regarded as a type of opinion. These casual/unstructured reviews contain a wealth of data that must be examined and extracted in order to acquire opinions or overall sentiments. The aim of this research is to figure out how to distinguish and summarize Amazon's Smartphone characteristics from large collections of consumer's reviews. It might be regarded as a type of opinion extraction research because the aim is to extract relevant knowledge from customer reviews.

1.5 Our contribution

In this research, a deep learning framework is offered for opinion extraction from unstructured product reviews. Proposed approach performs the word embedding and classification of opinions with the help of the HAN model. This paper's contribution can be outlined as follows:

- (a) To scrap the dataset of Amazon's Smartphone reviews.
- (b) To design a framework for opinion extraction that identifies the true polarity of customer reviews on their product.
- (c) To normalize and pre-process the noisy text collected from Amazon's website for appropriate sentiment classification.
- (d) To design a HAN (hierarchical attention network)-based framework that recognizes the opinions communicated in freestyle. In this framework, Glove (Pennington et al. 2014) is used for embedding and bi-directional GRU is used for semantic-based classification of opinions.
- (e) To assign a five-point polarity scale to the individual word/statement as well as a set of statements of the review that contained the product detail information as neutral, positive, negative, extremely positive and extremely negative statements.

1.6 Paper organization

The remaining paper is organized as follows: Section 2 outlines the important research related to opinion extraction techniques. Section 3 presents a proposed framework and methodology for opinion extraction on the basis of semantic classification using HAN. Section 4 presents the experiments and evaluation, Sect. 5 discusses the experimental results and evaluates the proposed methods, and Sect. 6 concludes the paper.

2 Literature survey

In recent years, with the expansion of online reviews, opinion extraction has received more attention to obtain the opinion of user. To enhance the online business, user's reviews cannot be under estimated. Therefore, many researchers have developed various systems to extract the opinion or information from reviews (Khan et al. 2014; Vo et al. 2018; Hussein 2018; Wang et al. 2019; Ratmele and Thakur 2019; Hajek et al. 2020). Some of the most relevant research is presented here.

Wei et al. (2020) presented an approach for sentiment analysis based on bi-directional LSTM with multi-polarity orthogonal attention. In this approach, to improve the Bi-LSTM model for implicit sentiment analysis, a novel multi-polarity orthogonal attention-based mechanism was introduced. They use a multi-polarity attention mechanism to initialize attentions by employing embeddings of words that differ with respect to sentiment polarity; as a result, their model can better capture the properties of each sentiment polarity. Abdalgader and Al Shibli (2020) suggested a lexicon-based word's polarity recognition approach on customer's reviews. In this approach, they compute the semantic relationship between the context expansion list of the target word and a synonym expansion list of all words around the target word inside the input text of reviews. Using this approach, they can get the highest score of semantic and sentimental knowledge, and people can easily understand the context of a large review. Yi and Liu (2020) designed a hybrid recommendation system (HRS) for customer sentiment analysis using a regression model of machine learning. This technique has been shown to be useful in identifying a customer's preferred shop based on the things they have purchased. The most notable characteristic of this HRS strategy is that there is no human intervention when it comes to determining customer shopping preferences. The mean absolute percentage error value for HRS was close to 98 percent, indicating a high level of accuracy. Kauffman et al. (2019) proposed a sentiment analysis on customer reviews to make a decision for marketing purpose. In this work, they used a quantity-based star score rating and sentiment analysis mechanism to check the subjective customer reviews and then classify the opinion of buyers. To make a decision for purchase of a product, they designed a framework in which NLP, sentiment analysis, data mining and clustering methods are used to calculate the sentiment score of a particular product's feature. The framework calculated the total score of the product on the basis of price and earlier mentioned sentiment score.

Lots of researches have been done in sentiment analysis for opinion extraction (Ratmele and Thakur 2021), and these are compared in terms of advantage and disadvantages. Do

et al. (2019) proposed a review study of 40 deep learning methods for sentiment analysis, in which they classified deep learning models into CNN, RNN, recursive NN and hybrid approach. They mention the advantage and weakness of all the three models. Advantage of CNN model is that it performs fast computation. It is capable of obtaining regional patterns from training data and produce nonlinear dynamics. The disadvantage of CNN is that it requires high demand of data. The advantage of RNN is that they need fewer amounts of data as well parameters. The disadvantage of RNN is that it cannot store the long-term dependencies and pick the last hidden state to indicate the statement that could result in an inaccurate prediction. The advantage of recurrent neural network is that it has basic architecture and easily able to understand tree structure but as a disadvantage it requires parser for tree structure that might be slow. It has been stated that RNN-based models outperform CNN-based models and that exploratory research into recurrent neural networks is needed. Park looked into the qualities that are incorporated in product reviews for five distinct categories of products and looked at how they affect review usefulness. Four data mining algorithms were tested on five real-life review datasets obtained from Amazon.com to see which one best predicts review helpfulness for every product type. The research shows that reviews for different kinds of products have different psychological and linguistic aspects, as well as different elements influencing review usefulness (Park 2018).

Jianqiang et al. (2018) proposed a deep CNN-based approach for sentiment analysis of twitter data. Initially the approach integrates with the word embedding features generated by the Glove on the basis of sentiments or polarity. The feature-based training data were processed through the deep CNN and then produced the model file for anticipating the sentiments by analyzing the twitter data. During the analysis of twitter data, text preprocessing plays an important role. Authors have compared and analyzed the various text preprocessing methods in another part (Jianqiang and Xiaolin 2017). For document classification, Yang et al. (2016) proposed a hierarchical attention network (HAN) model. The model has two features: (1) it has a hierarchical structure that relates to the hierarchy of documents. (2) When building the document representation, it has two levels of attention mechanisms used at the word and sentence levels, allowing it to attend differently to more and less relevant stuff. The proposed model gradually generates a document vector through integrating the important words into sentence vectors and subsequently aggregating important sentence vectors to document vectors. It is useful to extract important words and sentences from documents (Table 1).

3 Proposed work

3.1 OpExHAN framework

In this paper, OpExHAN framework is proposed for opinion extraction, which is based on the concept of word embedding and hierarchical attention mechanism. The opinions are extracted from Amazon Smartphone review dataset, which is scrapped from amazon.in. The overall framework of OpExHAN is shown in Fig. 1 which contains different phases for opinion extraction. The process starts from collecting real dataset through web scraping from amazon.in and then pre-processes the data for getting clean and normalized reviews. Then, we apply Glove embedding to extract word vector representation of reviews which identifies contextual information of words. These word vectors are fed into the hierarchical attention network and produce the vectors at word level and sentence level. Finally reviews are classified into five classes like extremely positive, positive, extremely negative, negative and neutral. Every phase of this framework is described in detail in further sections. In Fig. 1, OE represents Opinion Extraction, P , R , F represent Precision, Recall and F -Score, respectively, and Acc represents Accuracy.

3.2 Scraping the Amazon's dataset

Web scraping is performed by extracting data from the amazon.in website where the result page is "smartphone". For web scraping, the required URL is stored in a text file and then extracts the relevant information from the given URL. The extracted results are converted in columns and saved into CSV (comma separated) format. The extracted list of items are mentioned as 'mobile name', 'asin number', 'title review', 'user review' and 'star rating score'.

3.3 Preprocessing

The reviews are given by customers in a casual way on an online platform that's why it encloses a lot of noise such as informal words, multilingual words, hyperlinks and insignificant words. This noisy information increases the dimensionality of the issue. This problem is addressed by preprocessing, which is performed on each review, and generates the clean review file for further steps. In the proposed framework, preprocessing is performed with several steps which are mentioned below.

- Remove hyperlinks.
- Remove surplus spaces between words.
- Remove special characters.
- Convert informal words into its formal form like convert 'I've' to 'I have' etc.

Table 1 Previous research on opinion extraction and sentiment analysis

Work	Proposed method	Dataset	Language and domain	Results/performance	Advantage	Disadvantage
Kardakis et al. (2021)	Attention-based model using RNN	MR	English and movie	Acc = 79.97%	Attention-based methods achieve highest accuracy with gains of up to 3.5%	Different dropout rates are not used at the time of experiments
Abdalgader and Al Shibli (2020)	Word polarity method using Wordnet and SemWordnet with the contextual information	SUBJ IMDB SemEval-2017, SemEval-2016, SemEval-2015	English and hotel	Acc = 91.55% Acc = 87.92% Capture the highest score-based sentiment score to recognize the exact polarity of the words according to context	Preserve the sense or meaning of reviews in terms of polarity	Domain-specific method
Khalid et al. (2020)	Gradient-boosted support vector machine (GBSVM)	Twitter dataset	English and social platforms	Accuracy of Proposed (GBSVM) with TF-IDF is 93.0%	Performance of GBSVM is compared with four similar models such as GBM, SVM, LR, and RF and then it gives higher accuracy	It is applicable only in some cases of tf-idf features such as uni, bi and trigram
Kaufmann et al. (2019)	NLP, sentiment analysis and clustering techniques	44,460 reviews from Amazon	English and cell phones	Price + Sentiment Score of product to make a decision	Approach will helpful to marketing manager to improve the quality of product using negative features	Features of product extraction are depending upon the designed ontology
Feng et al. (2019)	Deep CNN + Word embedding + POS + explicit aspects	Dataset from Amazon	Chinese and mobile phone	$P = 77.75\%$ $R = 72.61\%$ $F1 = 75.09\%$	Obtain the sentiment annotation of the sentence	For higher performance, more time and computational resources should be spent in training
Ma et al. (2019)	Feature-based integrating memory networks (RNN)	SemEval '14	English and restaurant	Acc = 82.03%	Get better results with less hops for Sentiment analysis	Select the last hidden state to produce the sentence, sometimes resulting in an inaccurate prediction
Li et al. (2018)	Truncated history attention (THA) and selective transformation network (STN) built on two LSTMs	SemEval '14, SemEval '15, SemEval '16	English and laptop	Acc = 73.94% $F1 = 79.52\%$	Detected the aspects and generate the opinion summary. Proposed method achieved 5.0%, 1.6%, 1.4%, 1.3% absolute gains	Restricted to use joint extraction approach in proposed method
Mai and Le (2018)	Bi-directional RNN + CRF	YouTube Reviews	English and restaurant	$F1 = 71.46\%$ $P = 68.12\%$ $R = 75.87\%$ $F1 = 71.79\%$	Extracted opinions with sentiment polarity and approach can easily applicable on other language	They use the latest hidden state to represent the phrase because it can't capture long-term dependencies
Tay et al. (2018)	Aspect Fusion LSTM	SemEval '14	English and laptop	Acc = 75.44%	Predict the overall polarity of sentence using associative relation between sentence and aspect	Require high demand of data
				Acc = 68.81%		

Table 1 (continued)

Work	Proposed method	Dataset	Language and domain	Results/performance	Advantage	Disadvantage
Cui et al. (2017)	Single CNN	Amazon's Dataset	English and smart-phone	Acc = 84.87%	Aspect-based opinion extracted	For higher performance, more time and computational resources should be spent in training
Wang et al. (2016)	LSTM + aspect attention + aspect embeddings (RNN)	SemEval '16	English and restaurant	Acc = 77.2%	Proposed sentiment classification produced admirable performance over the baseline models	Selects the latest hidden state to represent the sentence, sometimes resulting in an inaccurate prediction
		SemEval '15	English and laptop	Acc = 68.9%		

- Perform sentence tokenization and word tokenization.
- Convert to lower case.
- Remove stop words.
- Apply Lemmatization (Gupta et al. 2016) on word tokens to identify root words.

3.4 Word embedding

Word embedding is an approach to convert word into vector representation. In the proposed method, pretrained Glove is used for embedding of words with vectors (Pennington et al. 2014). Glove is already trained on dataset of one billion words with vocabulary of 400,000 words. Although Glove has different embedding vector sizes like 50, 100, 200 and 300 dimensions, in this work 100 dimensions are selected for embedding. In the proposed work, the review can be mentioned as a document which is denoted as d with m sentences like $\{s_1, s_2, s_3, \dots, s_m\}$ and the length of i th sentence is M_i . The i th sentence denoted as s_i consists of words, which are denoted as M_i . The pre-trained glove model creates a word level vector, which represents as $w_e w_{ij}$ and mentioned in Eq. (1).

$$w_e w_{ij} = \{w_{i1}, w_{i2}, w_{i3}, \dots, w_{Mij}\} \tag{1}$$

3.5 Hierarchical Attention Network (HAN)

In the proposed work, hierarchical attention network (HAN) is used for opinion extraction. Basically, HAN model is designed to take the hierarchical structure in the form of review \rightarrow sentences \rightarrow words, where each review considered as a single document. This model incorporates an attention mechanism which is used to identify the contextually related most important words and sentences in a review. Figure 2 depicts the overall architecture of the hierarchical attention network. This architecture contains several components like word level and sentence level, where word level includes word encoder and word attention and sentence level includes sentence encoder, sentence attention. HAN uses gated recurrent unit (GRU) for encoding at both word level and sentence level. Description of every component is presented in the following subsections. In Fig. 2, ExPos represents Extremely Positive, Pos represent Positive, Neu, Neg and ExNeg represent Neutral, Negative and Extremely Negative accordingly.

3.5.1 Bi-GRU

Bidirectional gated recurrent unit (Bi-GRU) is employed in the proposed work, which is used to map a sequence of word vectors of the review to opinion classes (Bahdanau et al. 2014). Bi-GRU is the one of approaches of the recurrent neural network (RNN) model in which gates of GRU are

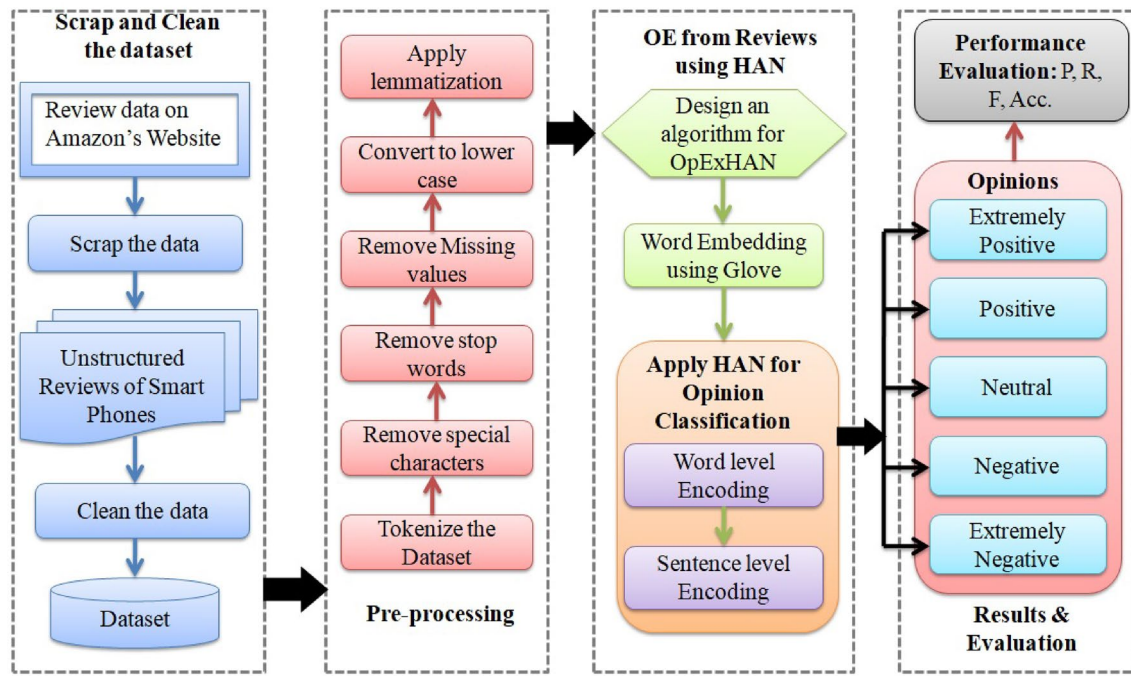


Fig. 1 Framework for OpExHAN

used to address the problem of gradient vanishing so that long-distance information can be preserved. GRU contains two gates: an update gate and reset gate, where update gate denoted as y_t and reset gate denoted as rs_t . The reset gate regulates how previous information affects the candidate state \tilde{hd}_t , and the update gate controls how old information is kept and new information is inserted. The GRU calculates the hidden new state hd_t at time t , mentioned in Eq. (2). This shows the linear interpolation based on the new sequence information between the previous state hd_{t-1} and the present new state hd_t . The gate y_t mentioned in Eq. (3) controls how much old data is retained and how much new data is added, where x_t is denoted as the sequence vector at time t .

$$hd_t = (1 - y_t) \odot hd_{t-1} + y_t \odot \tilde{hd}_t \tag{2}$$

$$y_t = \sigma(W_z x_t + U_z hd_{t-1} + b_z) \tag{3}$$

The candidate state \tilde{hd}_t is mentioned in Eq. (4), which is calculated in the same way as a typical recurrent neural network (RNN). When the reset gate rs_t equals to zero, then it fails to remember the previous state, in that case rs_t is updated, mentioned in Eq. (5).

$$\tilde{hd}_t = \tanh(W_h x_t + rs_t \odot (U_h hd_{t-1}) + b_h) \tag{4}$$

$$rs_t = \sigma(W_r x_t + U_r hd_{t-1} + b_r) \tag{5}$$

3.5.2 Hierarchical attention mechanism

HAN architecture consists of two levels: word level and sentence level, where GRU applied initially on word level subsequently on sentence level. The HAN model consists of the encoder which provides significant contexts and the attention mechanism which computes the important weights of these contexts as one vector.

Word level

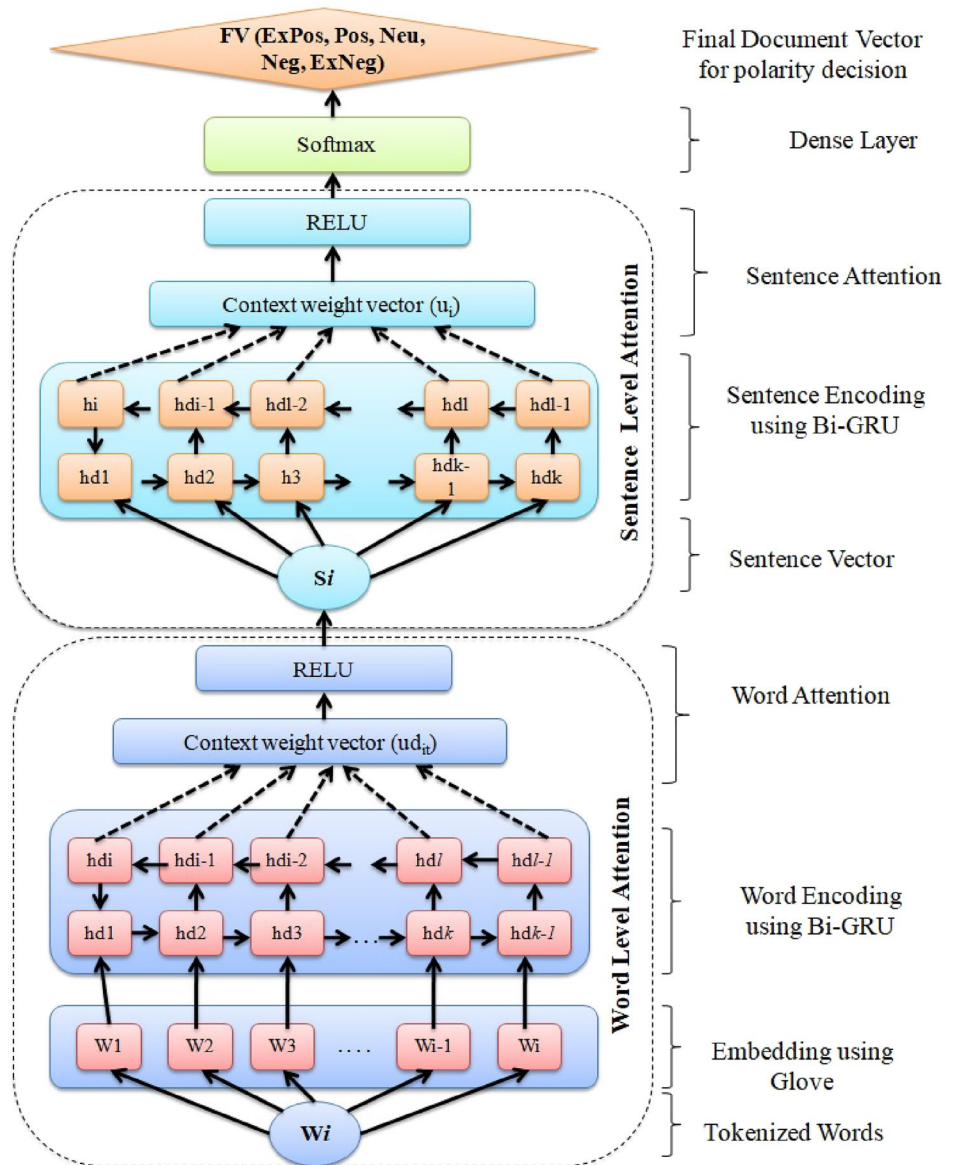
At word level, word tokens denoted as W_i are applied on Glove and perform embedding as $w_e w_{ij}$, shown in Eq. (1). Vectorized tokens are created after embedding, which are denoted as x_{im} , shown in Eq. (6), where d document contains M words and w_m depicts as the m th word in d and x_m depicts the m th word vector.

$$X_{im} = W_E W_{im}, m \in [1, M] \tag{6}$$

Word encoder

Vectorized tokens are applied as input to Bi-GRU (Yang et al. 2016) for encoding purpose. A GRU model contains the hidden state, which can be considered as a memory cell to transfer information. Bi-GRU is used to acquire word annotations by summarizing input from both sides of words and integrate contextual information into the annotation. GRU read the sentence s_i from w_{i1} to w_{iM} in the forward direction \longrightarrow and from w_{iM} to w_{i1} in the backward direction \longleftarrow

Fig. 2 Process of opinion extraction using hierarchical attention network



\overleftarrow{hd}_{im} , shown in Eqs. (7) and (8) accordingly. By concatenating the forward hidden state with the backward hidden state, get an annotation for a specific word w_{im} , mentioned in Eq. (9), which summarizes the content of the entire sentence focused around w_{im} . In implementation, 100 dimensions of embedding are taken as an input to the GRU. It considered 50 dimensions for forward direction and 50 for backward direction, because of the bidirectional nature of the model. Output dimensionality of GRU should be equal to 50, because it is running in forward and backward directions, which returns 100 dimensions which is the dimensionality of inputs.

$$\overrightarrow{hd}_{im} \xrightarrow{GRU} (x_{im}), m \in [1, M] \tag{7}$$

$$\overleftarrow{hd}_{im} \xrightarrow{GRU} (x_{im}), m \in [M, 1] \tag{8}$$

$$hd_{im} = \left[\overrightarrow{hd}_{im}, \overleftarrow{hd}_{im} \right] \tag{9}$$

Word attention

Word attention is used to extract those words which are important to create the sense of the sentence and combine the representations of these relevant words to produce a sentence vector. In the attention mechanism, the word annotation hd_{im} is given as input to hidden layer MLP (multilayer perceptron) (Sukhbaatar et al. 2015) and gets the hidden representation depicted as ud_{im} , shown in Eq. (10). Then, using a softmax function, calculate the importance of the word as

the similarity of ud_{im} with a word level context vector uv_w , and get a normalized significance weight α_{im} , mentioned in Eq. (11). Subsequently the sentence vector s_i is computed by concatenating the sum of these significance weights with earlier computed context annotations, shown in Eq. (12).

$$ud_{im} = \tanh(W_w hd_{im}) + b_w \tag{10}$$

$$\alpha_{im} = \frac{\exp(ud_{im}^M uv_w)}{\sum_t \exp(ud_{im}^M uv_w)} \tag{11}$$

$$s_i = \sum_t \alpha_{im} hd_{im} \tag{12}$$

Sentence level

The same network, which applied in word level, also applied in sentence level in progression, but focus is on sentence rather than word. Sentence vector s_i is already generated at word level that’s why there is no need to perform embedding at this step.

Sentence encoder

The sentence vector s_i is fed to Bi-GRU where GRU performs encoding of sentences and creates a final document vector for a review, which can be used as feature for opinion classification. Bi-GRU summarized the contexts of sentences by traversing the document forward and backward directions, which is mentioned in Eqs. (13) and (14) subsequently. Afterward, concatenate $\xrightarrow{hd_i}$ and $\xleftarrow{hd_i}$ to obtain an annotation of sentence i , mentioned in Eq. (15), where sentence i is the center point and hd_i summarizes the neighbor sentences around sentence i .

$$\xrightarrow{hd_i} = \xrightarrow{GRU} (s_i), i \in [1, K] \tag{13}$$

$$\xleftarrow{hd_i} = \xleftarrow{GRU} (s_i), i \in [K, 1] \tag{14}$$

$$hd_i = \left[\xleftarrow{hd_i}, \xrightarrow{hd_i} \right] \tag{15}$$

Sentence attention

The attention mechanism is again used to reward sentences that are indications to correctly classify a document. Sentence level context vector u_s is developed and we apply the vector to quantify the significance of the sentences. This concept is mentioned in Eqs. (16), (17) and (18), where FV is the document vector which summarizes complete information of sentences in the review.

$$u_i = \tanh(W_s hd_i) + b_s \tag{16}$$

$$\alpha_i = \frac{\exp(ud_i^M u_s)}{\sum_i \exp(ud_i^M u_s)} \tag{17}$$

$$FV = \sum_i \alpha_i hd_i \tag{18}$$

3.5.3 Opinion extraction

The probability of review depicted as pr mentioned in Eq. (19) belongs to each opinion class is computed using the final vector FV. This final vector is calculated as the feature representation of the document. The negative log likelihood of the correct labels is computed as training loss, which is presented in Eq. (20), where g is the label of document d .

$$pr = \text{softmax}(W_c FV + b_c) \tag{19}$$

$$\text{loss} = - \sum_d \log pr_{dg} \tag{20}$$

3.6 Proposed algorithm

The algorithm is proposed for opinion extraction which contains basic steps such as word embedding using GLOVE and then applies Bi-GRU with hierarchical attention mechanism. The input for the procedure is train and test dataset, and output is opinion classes with its accuracy, precision, recall and F -score. Different hyper-parameter values are set at the time of experimentations. Deep learning model is built at two levels: first word level and then sentence level using one Bi-GRU layer and one hidden layer with relu activation function at each level and one drop-out layer after sentence level. Initially, embedding layer performs word embedding on words and creates word vectors of similar words. Then, at word level Bi-GRU is applied on word vector, which is followed by identification of context similarity using attention mechanism and generate sentence vector. At sentence level, time-distributed function is applied to all word level layers on each sentence for sentence linking. The sentence vector is applied to Bi-GRU with attention mechanism to identify contextual similar sentences. After getting sentence sequences, drop-out layer is applied to prevent overfitting. Final document vector is created followed by extracting opinion classes at softmax layer using sigmoid activation function. After building the model, model is compiled using Adam optimizer and categorical cross-entropy loss. After compilation, model is validated with number of epochs and batch size on accuracy and loss. Then, performance of model is evaluated using precision, recall, accuracy and f -score measures.

Algorithm: Pseudo code for Opinion extraction Using Word Embedding + Bi-GRU + HAN Model

Input: Training Reviews Dataset (tnX), Training values such as Extremely Positive, Positive, Extremely Negative, Negative, Neutral (tnY)
 Testing Reviews Dataset, Testing values such as Extremely Positive, Positive, Extremely Negative, Negative, Neutral

Output: Accuracy, Precision, Recall, F-Score of extracted opinions.

1. **procedure** WordEmbedding + Bi-GRU + HAN Model (**tnX, tnY**)
 2. **Hyper parameters with Arguments**
 batchsize = 16; epochs = 25; max_features = 200000; embed_dim = 100; classes=5; input_length=56000; units=100
 3. **Build a Deep Learning Model**
 model = Sequential ()
 4. **Embedding Layer (Using Pre-trained Glove)**
 model.add (Embedding (max_features, embed_dim, input_length))
 5. **Bi-GRU Layer at word level**
 model.add (inputDim, word_sequence)
 6. **Dense Layer at word level**
 model.add (inputDim, activation='relu', name='word_dense') (word_gru)
 7. **Attention Layer at word level**
 model.add (EMBED_SIZE, name='word_att')(word_dense)
 8. **Create sentence vector**
 wordEncoder = Model(inputs = word_input, outputs = word_att)
 9. **TimeDistributed applies to all word level layers on each sentence for sentence linking**
 sent_encoder = TimeDistributed(wordEncoder,name='sent_linking')(sent_input)
 10. **Bi-GRU Layer at sentence level**
 model.add (inputDim, sent_encoder)
 11. **Dense Layer at sentence level**
 model.add (inputDim, activation='relu', name='Sent_dense') (Sent_gru)
 12. **Attention Layer at sentence level**
 model.add (EMBED_SIZE, name='sent_att') (sent_dense)
 13. **Dropout layer to prevent overfitting**
 model.add (Dropout (0.5, name='sent_dropout') (sent_att))
 14. **Softmax Layer**
 model.add (Dense (classes, activation='softmax'))
 15. **Model Compilation**
 model.compile (loss = 'categorical_crossentropy', optimizer = 'adam', metrics= [accuracy])
 16. **Model Summary**
 print (model.summary())
 17. **Fit a Model**
 for all epochs in (1: epochs) do
 model.fit (tnX , tnY , epochs, validation_data (ttX , ttY, batch_size=batchsize))
 18. **Model Evaluation**
 Acc = model.evaluate (ttX , ttY, batch_size = batchsize)
 Acc_total.append(Acc)
 End for
 19. return Acc_total
 Precision = model.evaluate (ttX , ttY)
 Recall = model.evaluate (ttX , ttY)
 F-Score = Geometric mean (Precision, Recall)
 20. End Procedure
-

4 Experiments and evaluation

4.1 Dataset

OpExHAN model is applied on Amazon Smartphone review dataset, which is scrapped from amazon.in web pages. 150,000 reviews are scrapped from web pages from which 56,000 Smartphone reviews are collected after preprocessing. From 56,000 reviews, 80% data, i.e., 44,800 reviews, are used as training, 10%, i.e., 5,600 reviews, for validation and rest 10%, i.e., 5600 reviews, for testing purpose. Maximum word features that are taken in experiments are 200,000, maximum sentences are 40, maximum word number is 50 and embedding dimension size is 100.

4.2 Evaluation metrics

The performance of OpExHAN model is evaluated using accuracy, precision, recall and F -Score measures. Recall can be represented as the ratio of number of correctly predicted classes of opinions with the cumulative number of reference opinion classes. Precision can be represented as the ratio of number of correct predicted classes of opinions with all predicted opinion classes. $F1$ score is the harmonic of recall and precision. The recall, precision and F -score are mentioned in Eqs. (21), (22), (23), respectively, where $S_{reference}$ is represented as reference opinion classes, $S_{predicted}$ is system-generated opinions. It is best to calculate both precision and recall before computing the F -measure.

$$R = \frac{|S_{reference} \cap S_{predicted}|}{S_{reference}} \quad (21)$$

$$P = \frac{|S_{reference} \cap S_{predicted}|}{S_{predicted}} \quad (22)$$

$$F = \frac{2 * P * R}{P + R} \quad (23)$$

4.3 Experimental setup

The proposed model is implemented in Python using high-level neural network API Keras for deep learning functions, NLTK for preprocessing and BeautifulSoup for web scraping. Keras conforming form is used to perform fast and efficient process. Table 2 presents the setup of various hyper-parameters with its different values and the best value that is selected for the final result. Model is experimented on validation dataset for hyper-parameters tuning with different values such as batch size, epochs and drops out, which

is mentioned in Table 2. Hyper-parameters are tuned using Adam optimizer and categorical cross-entropy as loss function. Categorical cross-entropy converts the integer sequence to a one-hot vector, which overcomes any memory issues. The OpExHAN model is experimented on test a dataset using selected best hyper-parameter values and gives best results.

5 Results and discussion

The OpExHAN model is trained and experimented on Amazon Smartphone dataset and compared the performance with the state-of-the-art methods and models on the same dataset. The performance of the proposed model is also compared with different hyper-parameters mentioned in Table 2. Performance comparisons of different batch sizes of OpExHAN are presented in Table 3. It is observed that small batch sizes give the greatest training consistency and generalization performance. Single batch acquires a lot of computation, which is too expensive and time-consuming or also it can be seen like higher batch size works inefficiently due to noise of the samples. If the large batch will be selected and noise will be smaller, then it can give better accuracy on training data, but the validation and test accuracy can be lower because of worst generalization and lost regularizing effect. To generate effective results and reduce noise, more information is required during training which is possible through smaller batch sizes. OpExHAN model gave best result with batch size 16, and the performance is outstanding in terms of accuracy, precision, recall and F -score and in another direction with batch size 128 results are inappropriate. From Table 3, it can be detected that the optimal batch size is 16 for the proposed model.

The model is also experimented using different epochs like 15, 25 and 50 with respect to batch size 16 and 32, and the performance results are shown in Table 4. It is observed that the model obtained excellent performance at epoch 25 with batch size of 16. At epoch 50, accuracy is only 1.48% less than epoch 25, that's why no more experiments are performed beyond epoch 50. Basically, high epoch size does not always mean that results will be more accurate. Epoch sizes can improve accuracy up to a certain point, but after that overfitting can occur in the model. Underfitting is also a result of having an extremely low epoch. When the batch size is large, the number of times the weight is updated every epoch is low, as compared to when the batch size is small. At small batch size, the reason for a high number of weight updates every epoch is because the loss value is calculated using fewer data points. There are more batches each epoch, because an epoch requires going through all of the training data points in a dataset. As a result, for the same number of epochs, the smaller the batch size, the greater the accuracy.

Table 2 Hyper-parameter setup

Hyperparameter	Values	Best value
Optimizer	Adam	Adam
Loss function	Categorical cross-entropy	Categorical cross-entropy
Activation function	Relu	Relu
Batch size	8, 16, 32, 64, 128	16
Epochs	15, 25, 50	25
Dropout rate	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.5

Table 3 Performance comparisons of different batch sizes of OpExHAN with epoch 25 in terms of evaluation measures

Batch size	Accuracy	Precision	Recall	F-score
8	88.13	90.46	90.31	90.35
16	94.68	91.67	91.25	91.14
32	90.00	84.46	84.15	83.79
64	85.62	84.87	85.00	84.33
128	78.44	78.70	79.06	78.73

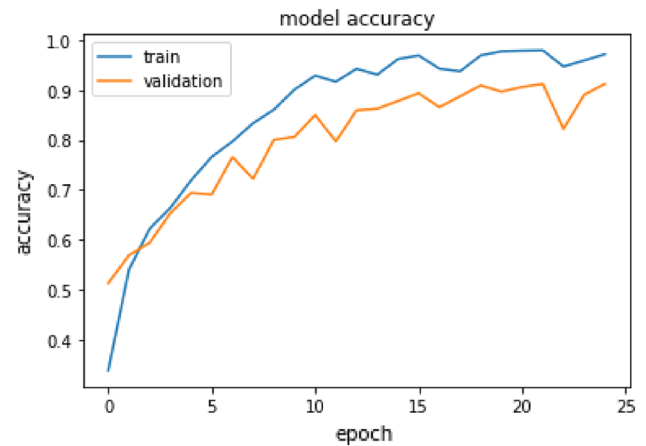
Bold values indicate best result are coming using Batch size = 16

The performance of the model with its accuracy and loss for train and validation data can also be represented. Model accuracy and model loss for batch size 16 are represented in Figs. 3 and 4, respectively. Model accuracy and loss for batch size 32 are also represented in Figs. 5 and 6 accordingly. Fundamentally through research it is identified that overfitting can be detected by the difference between training and validation accuracy, which shows the fact that the larger the gap, the higher the overfitting. The proposed model is performed best at epoch 25 with batch size 16, which is shown in Figs. 7 and 8 separately. Optimal accuracy is 94.69% and loss is 30.25%. Some facts identified by the experimentations that low accuracy with huge loss exist when huge errors are present on large data, low accuracy with low loss exist when few errors are present on large data and good accuracy with low loss exist when few errors are present on few data which can be considered as the ideal case.

Table 4 Performance comparisons at different epochs with batch size 16 and 32 in terms of evaluation measures

Epoch	Batch size-16				Batch size-32			
	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score
15	90.00	89.83	89.68	89.57	90.62	90.43	90.18	90.27
25	94.69	91.67	91.25	91.14	90.00	84.46	84.15	83.79
50	93.12	93.08	93.12	93.00	92.18	94.23	94.87	94.54

Bold values indicate best result are coming using Batch size = 16

**Fig. 3** Model accuracy at epoch 25 with batch size 16

The performance of the OpExHAN model is compared with existing methods, which are proposed by different authors. These existing methods are first applied on scrapped dataset, which is mentioned in this paper and then compared with the proposed algorithm. The comparison is mentioned in Fig. 9. Extensive research is done in the field of review opinion classification, but in this paper, few methods and model are focused for comparison, which performed best in previous research. Table 5 presents the performance comparison using precision, recall, F-score, and accuracy against recent existing methods of opinion extraction on the same scrapped Amazon review dataset. It is observed that OpExHAN model used the hierarchical attention using Bi-GRU with Glove-based word embedding and performed outstanding on state-of-the-art methods. These state-of-the-art methods are applied on scrapped dataset and then compared with OpExHAN model. The accuracy is 94.6%, which is better than other methods. In Alharbi et al. (2021), authors proposed Glove with GRU and Glove with LSTM approaches on Amazon review dataset, but performance is inadequate. Another method, LSTM with word2vec, is also proposed (Zhang et al. 2021), and when this method is applied on scrapped dataset, then the resultant accuracy is better but precision and recall are unsatisfactory. LSTM is applied with Glove and word2vec both, where Glove performed better than word2vec on Amazon Smartphone review dataset. Two different methods in a sequential manner in a single model are also applied (Shrestha and Nasoz 2019) on the same

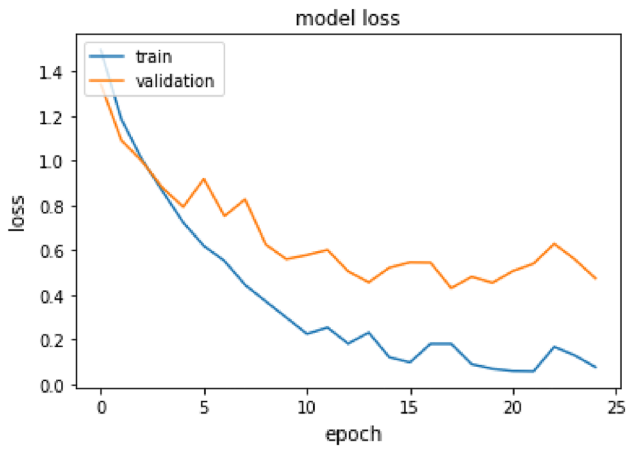


Fig. 4 Model loss at epoch 25 with batch size 16

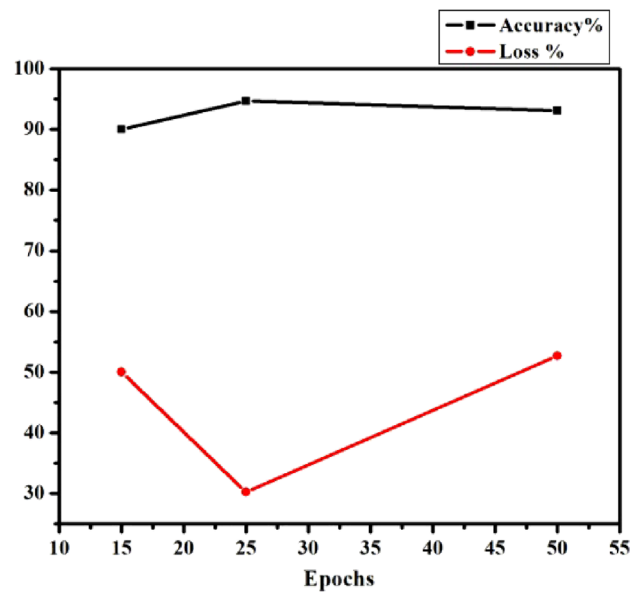


Fig. 7 Comparison of model accuracy and loss at epoch 15, 25, 50

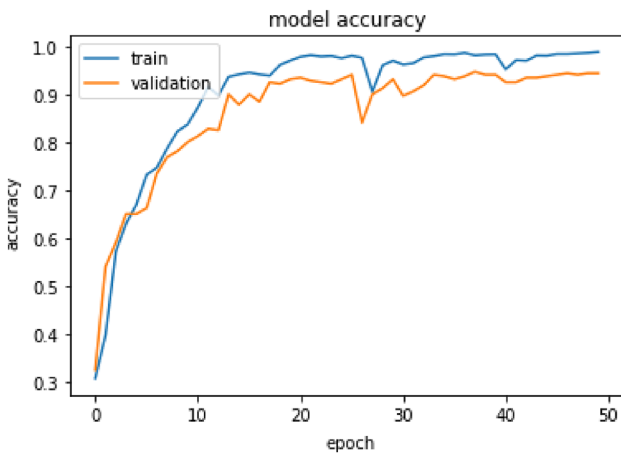


Fig. 5 Model accuracy at epoch 25 with batch size 32

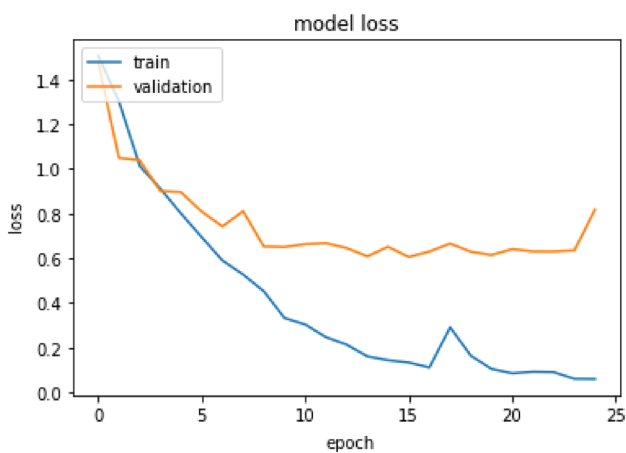


Fig. 6 Model loss at epoch 25 with batch size 32

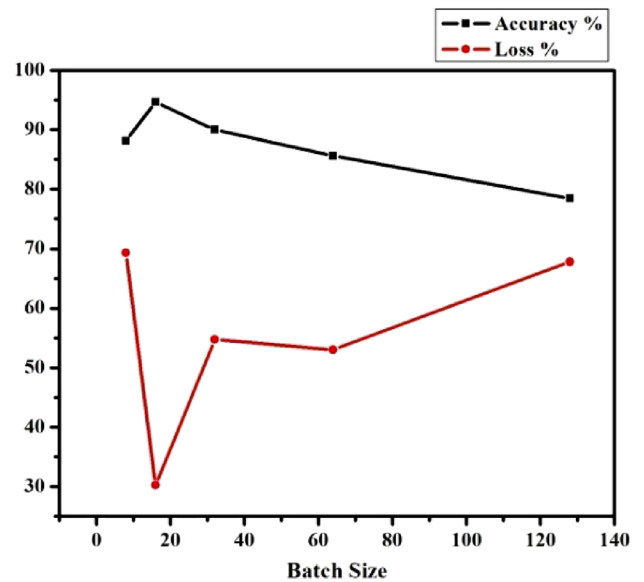


Fig. 8 Comparison of model accuracy and loss at batch sizes 8, 16, 32, 64 and 128

dataset, where GRU is used for feature extraction and SVM is used for classification of opinions. The performance of this model is also unsatisfactory. In the proposed model, the use of the attention mechanism with hierarchical approach has improved the performance of GRU. Analysis of review is performed first at word level and then at sentence level, through which contextual information is identified properly at both levels. The Bi-GRU model with attention is better

Table 5 Performance comparison of opinion extraction against existing methods on scrapped Amazon Reviews Dataset

Authors	Accuracy	Precision	Recall	F-Score
GRU + Glove	56.04	68.00	69.06	68.53
LSTM + Glove	73.06	80.59	81.16	80.87
GRU + SVM	75.82	59.00	52.12	55.34
LSTM + Word2Vec	85.00	74.00	79.00	76.41
OpExHAN (Bi-GRU + Glove)	94.69	91.67	91.25	91.14

able to find a long-term dependency between explicit words in capturing contextual information of previous and next words. In the proposed model, lemmatization is performed in pre-processing, which gives the root words and astonishingly is able to identify contextually similar words through glove, which is also the reason for excellent performance.

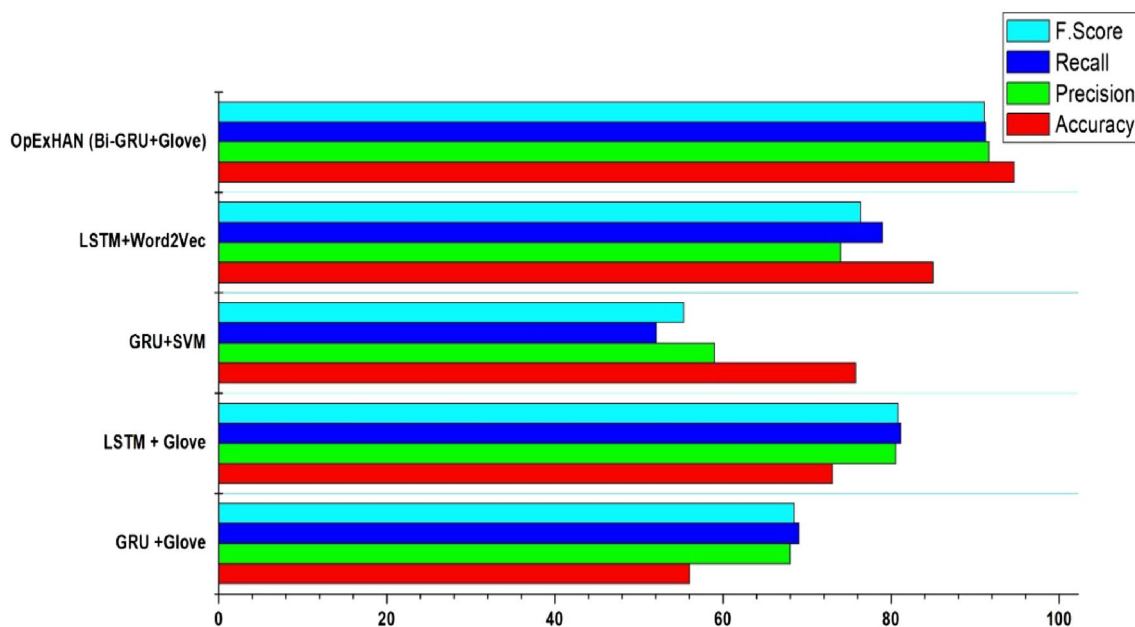
The precision and recall of the proposed model are both around 91%, which is admirable value but 8–9% is still required to improve the performance. This deficiency is because of some major points, which are identified at error analysis. The OpExHAN model is unable to identify valid Smartphone features because of co-referencing between sentences of reviews, which degrades the recall value. Few reviews contain multiple categories of product features with its polarity in a single sentence, which are failed to notice and affect the recall value. But these types of reviews are very less in the Amazon dataset, so the performance is not too much affected. The precision value lowers because the

model is also identified few terms as polarity of Smartphone features, which are contextually irrelevant in sentences. These major issues can be resolved by using product feature-based opinion prediction. Grammar and spell checking are also required to include in the proposed model for improving the performance. Comparative opinions are also not identified in this research. Few examples of reviews from Amazon review dataset are identified, which also degrade the performance.

Review 1: "Samsung Galaxy M02 (Blue, 3 GB RAM, 32 GB Storage)", Undoubtedly- High Specs. at such Low and Very Affordable Cost. Full Value for Money. Loving it..., Battery Life: Considerably Good... Value for Money: 100% Paisa Vasool...Sheerness: Considerably High Specs. at such Low and Affordable Cost...Improvement Suggestion: Charger could have been C-Type. This may be considered for Updated Version or New Production., Extremely Positive.

ISSUE: Multilingual issue which is not identified by Glove. Glove has only a dictionary of English words. In the above review, "Paisa Vasool" term belongs to the Hindi language so that it is not detected by Glove.

Review 2: Dont expect much from this phone., " Phone is only for Calling, WhatsApp and Youtube. I gifted this phone to my mother, so as per her usage i felt its okay. Worst part is the camera.... I even had Redmi Note 4.... I seriously felt Note 4 camera is much better that redmi 9.I got this phone in 7500/-", Negative.

**Fig. 9** Comparison of OpExHAN model with baseline exiting models

ISSUE: One part of review shows the positive polarity and another shows the negative polarity. So that overall polarity should be neutral. But it is quoted as negative polarity. There is a mismatch between polarities and it is the ambiguous review.

Review 3: front camera, use front cem I am not happy, Neutral.

ISSUE: Camera is written as ‘cem’, it is not identified correctly by the proposed method.

6 Conclusion and future work

In this paper, HAN-based Bi-GRU approach is used to extract the opinions from large amount of Amazon’s Smart phone’s review dataset. OpExHAN model used the hierarchical attention using Bi-GRU with Glove-based word embedding and performed outstanding on state-of-the-art methods. Performance of the proposed method is compared with several baseline existing methods by conducting various experiments on Amazon’s dataset. Hence, the effectiveness of the proposed method is illustrated through various evaluation measures. The precision and recall of the proposed model are both around 91% and accuracy is 94.6%, which are admirable numbers. In future research, the extraction of aspect-based opinions will be focused, in order to make a purchasing decision with product feature-based opinion prediction. The OpExHAN model is unable to resolve some issues, which can target in future research, that are mentioned as:

1. Co-referencing between sentences of the review.
2. Multiple categories of product with its features that are mentioned in a single sentence.
3. Pruning of contextually irrelevant features and words from reviews.
4. Grammar and spell checking is also important before lemmatization and word embedding.
5. Comparative opinion identification in a review.
6. Multilingual reviews analysis for correct opinion identification.

Funding There is no funding for this manuscript.

Data availability The datasets analyzed during the current study are not publicly available but are available from the corresponding author on reasonable request.

Declarations

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

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Abdalgader K, Al Shibli A (2020) Experimental results on customer reviews using lexicon-based word polarity identification method. *IEEE Access* 8:179955–179969
- Abrahams AS, Jiao J, Fan W, Wang GA, Zhang Z (2013) What’s buzzing in the blizzard of buzz? Automotive component isolation in social media postings. *Decis Support Syst* 55:871–882
- Alharbi NM, Alghamdi NS, Alkhamash EH, Al Amri JF (2021) Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews. *Math Probl Eng* 2021:1–10
- Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. [arXiv:1409.0473](https://arxiv.org/abs/1409.0473)
- Balahur A, Kabadjov M, Steinberger J, Steinberger R, Montoyo A (2012) Challenges and solutions in the opinion summarization of user-generated content. *J Intell Inf Syst* 39:375–398
- Chen Y, Xie J (2008) Online consumer review: word-of-mouth as a new element of marketing communication mix. *Manag Sci* 54:477–491
- Dhokrat A, Khillare S, Mahender CN (2015) Review on techniques and tools used for opinion mining. *Int J Comput Appl Technol Res* 4:419–424
- Do HH, Prasad P, Maag A, Alsadoon A (2019) Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Syst Appl* 118:272–299
- Feng J, Cai S, Ma X (2019) Enhanced sentiment labeling and implicit aspect identification by integration of deep convolution neural network and sequential algorithm. *Clust Comput* 22:5839–5857
- Gu X, Gu Y, Wu H (2017) Cascaded convolutional neural networks for aspect-based opinion summary. *Neural Process Lett* 46:581–594
- Gupta V, Joshi N, Mathur I (2016) Design and development of a rule-based Urdu lemmatizer. In: *Proceedings of international conference on ICT for sustainable development*. Springer, pp 161–169
- Hajek P, Barushka A, Munk M (2020) Opinion mining of consumer reviews using deep neural networks with word-sentiment associations. In: *IFIP international conference on artificial intelligence applications and innovations*. Springer, pp 419–429
- Haque TU, Saber NN, Shah FM (2018) Sentiment analysis on large scale Amazon product reviews. In: *2018 IEEE international conference on innovative research and development (ICIRD)*. IEEE, pp 1–6
- Hussein DME-DM (2018) A survey on sentiment analysis challenges. *J King Saud Univ Eng Sci* 30:330–338
- Jianqiang Z, Xiaolin G (2017) Comparison research on text preprocessing methods on twitter sentiment analysis. *IEEE Access* 5:2870–2879
- Jianqiang Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for twitter sentiment analysis. *IEEE Access* 6:23253–23260
- Kardakis S, Perikos I, Grivokostopoulou F, Hatzilygeroudis I (2021) Examining attention mechanisms in deep learning models for sentiment analysis. *Appl Sci* 11:3883
- Kauffmann E, Peral J, Gil D, Ferrández A, Sellers R, Mora H (2019) Managing marketing decision-making with sentiment analysis: an evaluation of the main product features using text data mining. *Sustainability* 11:4235

- Khalid M, Ashraf I, Mehmood A, Ullah S, Ahmad M, Choi GS (2020) GBSVM: sentiment classification from unstructured reviews using ensemble classifier. *Appl Sci* 10:2788
- Khan K, Baharudin B, Khan A, Ullah A (2014) Mining opinion components from unstructured reviews: a review. *J King Saud Univ Comput Inf Sci* 26:258–275
- Li X, Bing L, Li P, Lam W, Yang Z (2018) Aspect term extraction with history attention and selective transformation. *arXiv preprint arXiv:1805.00760*
- Liu B (2012) Sentiment analysis and opinion mining. In: *Synthesis lectures on human language technologies*, vol 5, no 1, p 167
- Ma R, Wang K, Qiu T, Sangaiah AK, Lin D, Liaqat HB (2019) Feature-based compositing memory networks for aspect-based sentiment classification in social internet of things. *Futur Gener Comput Syst* 92:879–888
- Mai L, Le B (2018) Aspect-based sentiment analysis of vietnamese texts with deep learning. In: *Asian conference on intelligent information and database systems*. Springer, pp 149–158
- Pang B, Lee L (2008) Opinion mining and sentiment analysis. In: *Foundations and Trends® in Information Retrieval*, vol 2, pp 1–135
- Park Y-J (2018) Predicting the helpfulness of online customer reviews across different product types. *Sustainability* 10:1735
- Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp 1532–1543
- Qiu G, Liu B, Bu J, Chen C (2011) Opinion word expansion and target extraction through double propagation. *Comput Linguist* 37:9–27
- Ratmele A, Thakur R (2019) Statistical analysis & survey of research work in opinion mining. In: *Proceedings of Recent Advances in Interdisciplinary Trends in Engineering & Applications (RAITEA)*.
- Ratmele A, Thakur R (2020) Feature based opinion classification (FBOC) of customer reviews. *Indian J Comput Sci Eng* 11(6):892–898
- Ratmele A, Thakur R (2021) Feature extraction of unstructured product reviews. In: *Innovative data communication technologies and application*. Springer
- Shrestha N, Nasoz F (2019) Deep learning sentiment analysis of amazon.com reviews and ratings. *arXiv preprint arXiv:1904.04096*
- Sukhbaatar S, Szlama A, Weston J, Fergus R (2015) End-to-end memory networks. *arXiv preprint arXiv:1503.08895*
- Tan W, Wang X, & Xu X (2019) Sentiment analysis for Amazon reviews. *Reseachgate*, 0-9.
- Tay Y, Tuan LA, Hui SC (2018) Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In: *Proceedings of the AAAI conference on artificial intelligence*
- Turney PD (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032*
- Vo A-D, Nguyen Q-P, Ock C-Y (2018) Opinion–aspect relations in cognizing customer feelings via reviews. *IEEE Access* 6:5415–5426
- Wang Y, Huang M, Zhu X, Zhao L (2016) Attention-based LSTM for aspect-level sentiment classification. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp 606–615
- Wang R, Zhou D, Jiang M, Si J, Yang Y (2019) A survey on opinion mining: from stance to product aspect. *IEEE Access* 7:41101–41124
- Wei J, Liao J, Yang Z, Wang S, Zhao Q (2020) BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing* 383:165–173
- Wu C, Wu F, Wu S, Yuan S, Huang Y (2018) A hybrid unsupervised method for aspect term and opinion target extraction. *Knowledge-Based Syst* 148:66–73
- Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp 1480–1489
- Yi S, Liu X (2020) Machine learning based customer sentiment analysis for recommending shoppers, shops based on customers' review. *Complex Intell Syst* 6:621–634
- Zhang M, Fan B, Zhang N, Wang W, Fan W (2021) Mining product innovation ideas from online reviews. *Inf Process Manag* 58:102389

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.