Multimedia Tools and Applications (2024) 83:3085–3110
https://doi.org/10.1007/s11042-022-13339-4

1203: APPLICATIONS OF ADVANCED ARTIFICIAL INTELLIGENCE IN MULTIMEDIA AND INFORMATION SECURITY

An efficient two-state GRU based on feature attention mechanism for sentiment analysis

Muhammad Zulqarnain[1,2] • Rozaida Ghazali[1] • Muhammad Aamir[1] • Yana Mazwin Mohmad Hassim[1]

Received: 26 July 2020 / Revised: 19 January 2022 / Accepted: 2 June 2022 / Published online: 18 June 2022

Abstract

Sentiment analysis is one of the most challenging tasks in natural language processing (NLP). The extensively used application of sentiment analysis is sentiment classification of reviews. The purpose of sentiment classification is to determine the sentiment polarity of user opinion, attitude, and emotions expressed in the form of text into positive, negative and neutral polarities. Many advanced deep learning approaches have been proposed to solve sentiment analysis problem. Recurrent neural network (RNN) is one of the popular deep learning architectures which is widely employed in sentiment analysis. In this paper, we proposed a Two State GRU (TS-GRU) based on feature attention mechanism that concentrates on identifying and categorization of the sentiment polarity using sequential modeling and word-feature seizing. The proposed approach integrates pre-feature attention in TS-GRU to associate the complex connection between words by sentence based sequential modeling and capturing the keywords using attention layer for sentiment polarity. Subsequently, a decoder function has been added in the post-feature attention GRU, in order to extract the predicted features during attention mechanism. The proposed approach has been evaluated on three benchmark datasets including IMDB, MR, and SST2. Experimental results conclude that the proposed TS-GRU model obtained higher sentiment analysis accuracy of 90.85%, 80.72%, and 86.51% on IMDB, MR, and SST2 datasets, respectively.

✉ Muhammad Zulqarnain
zulqarnainmalik321@gmail.com

Rozaida Ghazali
rozaida@uthm.edu.my

[1] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor, Malaysia

[2] Riphah College of Computing, Riphah International University Faisalabad Campus, Faisalabad, Pakistan

Keywords Recurrent neural network · Gated recurrent unit · Two State GRU · Attention mechanism · Sentiment analysis

1 Introduction

Natural Language Processing (NLP) is an important branch of Artificial Intelligence which deals the interaction between understanding human language and machine language such as text and speech [2]. Nowadays, the fast growing world of electronic devices generates a huge amount of text data. According to the survey, the 18.7 billion messages in the form of text are universally produced every day [13] and 350 million photos are uploading to Facebook daily, which is the world largest social network media with 2.2 billion active users, while Google performs 41,000 search queries every second [35]. In order to process all this data, researchers perform sentiment analysis, which has become one of the most essential tasks in NLP. The purpose of sentiment analysis is to predict the sentiment polarities as negative, positive or neutral classes on given text. Many research oriented industries need help to process and extract useful information from large amount of text data including business values such as brand monitoring, customer services, market research, and social services [16, 34]. For example, merchants' needs to be assist for adjusting their marketing scheme based on a complete consumer's preferences. Generally, the context of data classification can be categorized into two major types, "structured data" which has been comprised and organized into a clearly formatted repository data types and "unstructured data", which deals unorganized format. Unstructured data is more difficult to collect and process, usually data from social media platforms. According to estimate by experts, in 2022 the unstructured data would increase 93% of all data around the global [1].

In recent years, NLP is one of the interested areas for deep learning tasks due to fast growth in text data. The major steps of NLP are presented in Fig. 1 which is applied in several fields including text classification, sentiment analysis, name entity recognition, information extraction, question-answering, semantic, spam filtering, documents classification, relationship extraction, speech recognition, keywords categorization and text clustering [14, 24, 42, 48].

The essential purpose of this research is to examine the way of sentiment analysis linguistics information over deep learning approaches for obtaining the maximum potential in the term of sentiments polarities. In the last few decades, deep learning based techniques have been extensively applied in many data mining applications are included, part-of-speech tagging, text classification, keyword extraction and sentiment analysis have successfully achieved superior results in data mining tasks [8, 28]. The deep learning models have capability to present information from original text data without laborious features engineering method, and extract the long-term dependencies among the context words and sentence in an adjustable method efficiently as compared to other standard machine learning techniques [49].

The word embedding Word2Vec method proposed by Lee et al. [17], in which the authors convert each word with sparse vector identical to the number of particular terns and distributed the various semantic and syntax features of each word to each dimension in vector space. For word embedding module Zulqarnain et al. [45] has applied Word2Vec method for sentiment analysis tasks in order to facilitate feature modeling study. Kalchbrener et al. [12] applied CNN for modeling sentences to achieved superior results in text classification tasks through the convolutional computations process to efficiently classify sentiment polarity based on n-gram

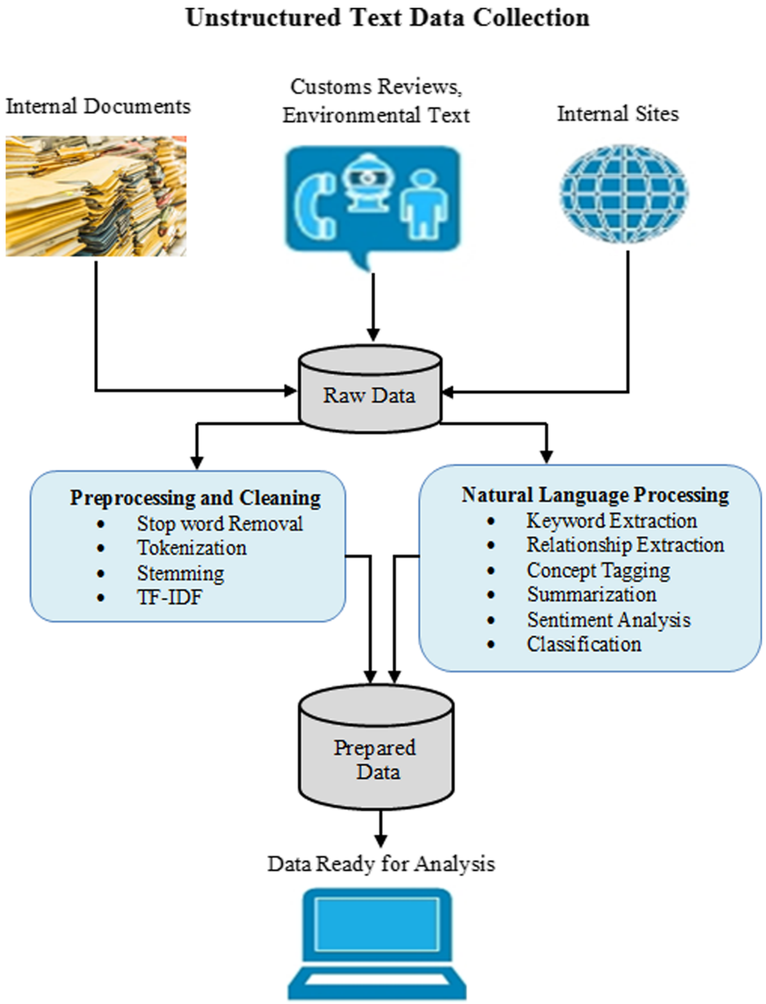

Fig. 1 Text mining process

features method from various position of a sentence. However, features context-based extraction CNN entirely excludes the sequence information while it frequently concentrates on the local features of a sentence. On the other hand, a GRU and LSTM models are better in learning of sequential correlation matrix among the context words and sentiment words by applying a versatile gating architecture [6]. However, the vanishing gradient and exploding issue [3] carried by RNN also observed by drawback. To addresses this kind of drawback, there are several deep learning approaches have been introduced by different researchers are included Long Short-Term Memory (LSTM) [11] and Gated Recurrent Unit (GRU) [5]. GRU is a simplified variant of LSTM; which has been broadly applied in many NLP tasks due to the more efficient execution process and also maintain the advantages of LSTM.

In this research, we investigate the effective functionality of the existing GRU for sentiment analysis through disseminated representation in public platform. One of the ability of GRU network is to solve the vanishing gradient or exploding issue in traditional RNNs.

Furthermore, we carried on the experiments on three benchmark datasets such as MR, SSTb2 and IMDB. Our proposed framework presented further enhancement in the existing GRU architecture based on Feature-Attention Two State GRU (TS-GRU) with attention mechanism in order to select the most information features for sentiment analysis. The main purpose of our research is to enhance the existing GRU architecture in order to improve accuracy of sentiment polarity and reduce the information loss. Therefore, the main contributions of this study are presented as follows:

- We proposed Feature Attention Two State GRU (TS-GRU) framework for solving sentiment analysis issues. This framework extracts informative features through feature mechanism.
- For vector initialization, we used a traditional and highly practiced unsupervised word embedding technique as known global vector (GloVe) [27].
- Our contribution concentrated on the development of a mechanism which provides an efficient computational process and robust performance with fewer parameters.
- Experimental results illustrate that our proposed TS-GRU performs well on all benchmark datasets except MR and focus to captures the long-term dependencies, while it presented excellent results over significantly less computational cost as compared to standard architecture [4, 9, 19–22, 29, 37, 39, 43, 44].

2 Preliminaries

This section describes the preliminaries study of recurrent neural networks (RNNS) that have been widely used in NLP tasks. The RNN is a type of artificial neural network which performs the sequential tasks between the units in a cycle connection. The standard RNNs contain fully connected layers, and they compute input layer to hidden layer and hidden to output layer using recurrent connection as presented in Fig. 2. The Chung et al. [38] used three layered network architecture with the addition of a “context unit” set. The connection between the nodes of hidden layer and the nodes of context layer is based on fixed weights. The RNN is also a deep neural network that is added to sequential data, resultantly RNN is extremely expressive [18].

RNN keep a vector of activations for each time step, which makes the RNN a deep neural network. The decision of a recurrent network at time step $t-1$ depend the next decision at time step t. Therefore, RNN has two sources of input, one is the present and second is the

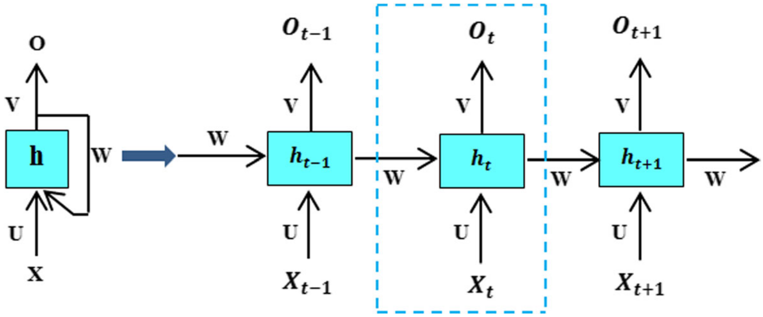

Fig. 2 Computational architecture of RNN [30]

previous, which combines to decide how it will respond to new data. However, the standard RNNs has the capability of learning sequential time series data patterns, and vanishing gradient issue that affects its performance. However, RNN is a powerful network, when handling with sequential issues; while it is difficult to train with the gradient descent technique due to basic vanishing gradient and exploding issue [32]. In contrast, this issue is addressed by two advance variants of standard RNN such as Long Short-Term Memory (LSTM) and GRU. However, GRU contain less parameters than LSTM and prevents overfitting issue, as well as saves training time.

Given sequential input of word vectors (X_1, X_2, $X_{3...}X_T$), it generates a sequential hidden state (h_1, $h_{2,}$ $h_3 \ldots h_T$), which is calculated at time step t, thus the output can be calculating using RNN as follow:

$$O_t = \phi(W_x x_t + U_o h_{t-1}) \tag{1}$$

$$H_t^l = \phi(W_x h_{t-1}^l + U_H h_{t-1}^l) \tag{2}$$

where U_o, U_H is the recurrent weights matrix, W_x is the input to-hidden weights matrix, and φ is an arbitrary activation function. The Eqs. 1 and 2 shows the hidden layer activity h_t with its previous hidden layer activity h_{t-1}

On the other hand, the GRU is an advance and simplified variant of LSTM was initially proposed by Chung et al. [7], on statistical machine translation. GRU inspired by LSTM, which control the information flow inside the unit through update gate z_t and reset gate r_t without separate memory. Therefore, GRU have a capability of captures the mapping relationship between time series data [10, 46] while it also has attractive advantages such as less complexity and efficient computational process. The architecture of GRU, which illustrates the relationship among update and reset gate is presented in Fig. 3.

Similar to the functioning of LSTM, the GRU has gating units that manage the movement of data inside the unit without making utilize of any additional separate memory cells [40]. However, GRU store and filter information through internal memory capability and combine the input gate and forget gate into a single update gate with previous activation $\boldsymbol{h}_{t-1}$ and the candidate state represented by $\boldsymbol{\hbar}_t$. There are three major components of GRU are included update gate, reset gate and candidate state and its Eqs. (3 to 6) are follows:

$$\boldsymbol{z}_t = \phi(\boldsymbol{V}_{xz}\boldsymbol{x}_t + \boldsymbol{U}_{hz}\boldsymbol{h}_{t-1} + \boldsymbol{B}_z) \tag{3}$$

$$\boldsymbol{r}_t = \phi(\boldsymbol{V}_{xr}\boldsymbol{x}_t + \boldsymbol{U}_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{B}_r) \tag{4}$$

$$\boldsymbol{\hbar}_t = tanh(\boldsymbol{V}_{x\hbar}\boldsymbol{x}_t + \boldsymbol{U}_{h\hbar}(\boldsymbol{r}_t * \boldsymbol{h}_{t-1}) + \boldsymbol{B}_{\hbar}) \tag{5}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) * \boldsymbol{h}_{t-1} + \boldsymbol{z}_t * \boldsymbol{\hbar}_t) \tag{6}$$

Where $\boldsymbol{V}_{xz}$, $\boldsymbol{V}_{xr}$and $\boldsymbol{V}_{x\hbar}$ refer the weight matrix among the input layer and update gate, reset gate and candidate state while recurrent connection weight matrix are represented by $\boldsymbol{U}_{hz}$, $\boldsymbol{U}_{hr}$and

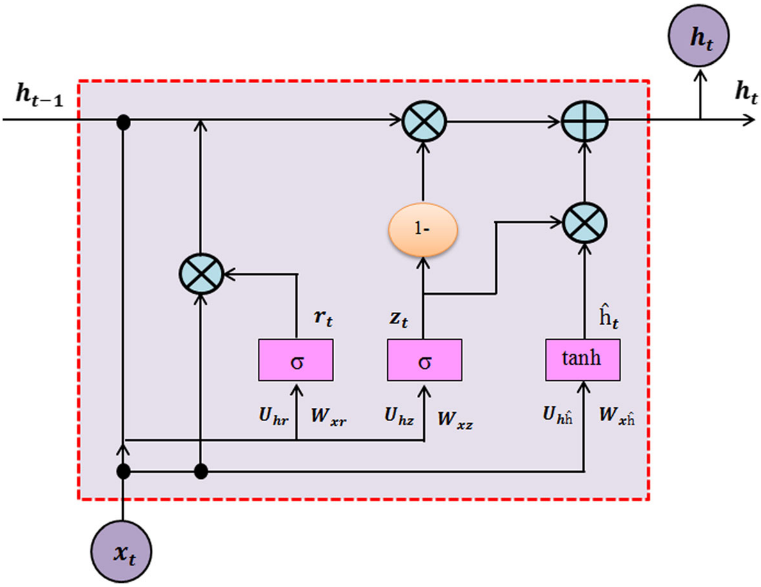

Fig. 3 Conventional architecture of GRU [31]

$\boldsymbol{U}_{h\hbar}$ respectively. $\boldsymbol{x}_t$ is the time series sample input, and hidden output is denoted by $\boldsymbol{h}_t$. ϕ is the sigmoid activation function of update and reset gates, * is perform element-wise multiplication operation and $\boldsymbol{B}_z$,$\boldsymbol{B}_r$ and $\boldsymbol{B}_{\hbar}$ are the corresponding biases.

3 The proposed mechanism

In this work, we illustrate the individual description of the proposed mechanism which includes two-state GRU with feature attention mechanism. The proposed architecture apply word embedding as inputs and learns to extract high-level context words features over time steps, and those features are predicted by embedding layer then provided to TS-GRU language mechanism, and finally, used a softmax classifier for final classification. The main contribution of proposed mechanism is extracting important features in two major phases included Pre-Feature Attention “TS-GRU and Post-Feature Attention TS-GRU.

3.1 The Embedding Layer

In the sentiment analysis, embedding layer plays an essential role in the model development. In sentence classification, the first step is preprocessing which involves cleaning the text data. We used the pre-trained 300-dimensional GloVe [27] method in the words embedding layer in sequence to transform each word in the sentences to a real value vector. Similarly, in the embedding process the pre-trained word vectors help to capture semantic and syntactic information, which are significantly for the tasks of sentiment classification and converts words context into real-valued features vectors. Let $L \in R^{V*d}$ be the embedding query table generated by Glove, where dimension of words is presented by d and V is the vocabulary size. Suppose that the sequential input includes of n words and the sentiment resource comprises m

words. The input sentence collects the word vectors from L and obtains a list of vectors [W_1, W_2 …, W_n] where $W_i \in R^d$ is the word vector of the i^{th} word. Consequently, the sentiment resource sequence can recover the word vectors from a list of vectors [W_1^s, W_2^s, …, W_m^s]. In this way, we can get the matrix $W^c = [W_1, W_2, \dots, W_n] \in R^{n*d}$ for context words and the matrix $W^s = [W_1^s, W_2^s, \dots, W_m^s] \in R^{m*d}$ for sentiment resource words. Consequently, we describe to structure the word-level connection among the sentiment words and the context words to the format of the correlations matrix presented in Fig. 4. This process is easily combination of all words embedding in V.

The training process took approximately 35 min using python compiler, in which each word is illustrated through a vector maximum "300-dimension. There are three core functions are defined:

- Build_vocab requires Harvard IV-4 dictionary and movie reviews dataset as the input and the output of the function is word_frequency array is includes special word_id and its frequency in the complete dataset. The range of sentiment value has spread from 0 to 1, where 0 is negative and 1 is positive 1, the word which has high sentiment value will have high word_frequency value.
- Build_cooccur required parameters to accept a word_frequency array and movie reviews dataset: the size of context window and minimum count were initialized by 10. The number of words applied to represent the context of each word through context window, while the mini-mum count was applied to drop rare word co-occurrence pairs. The range from context_word_id to m_word is calculated through count the number of words among m_word and context_word. This function builds a sparse matrix which yields co-occurrence tuples in the form *(word_id, context_word_id, xij*), where xij is the co-occurrence value. The flow diagrams of Build_vocab function and Build_co-occurrence function is presented in Figs. 5 and 6 respectively.

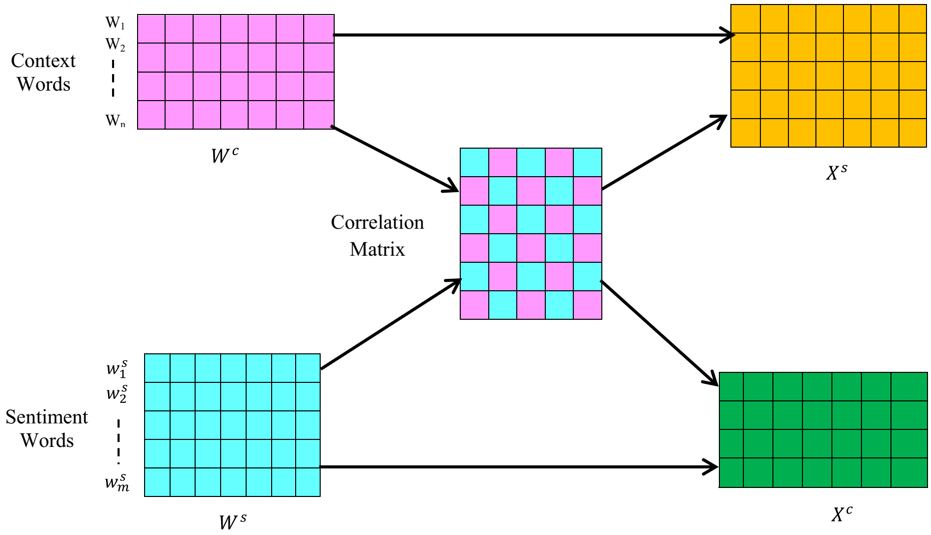

Fig. 4 Sentiment-context word correlation

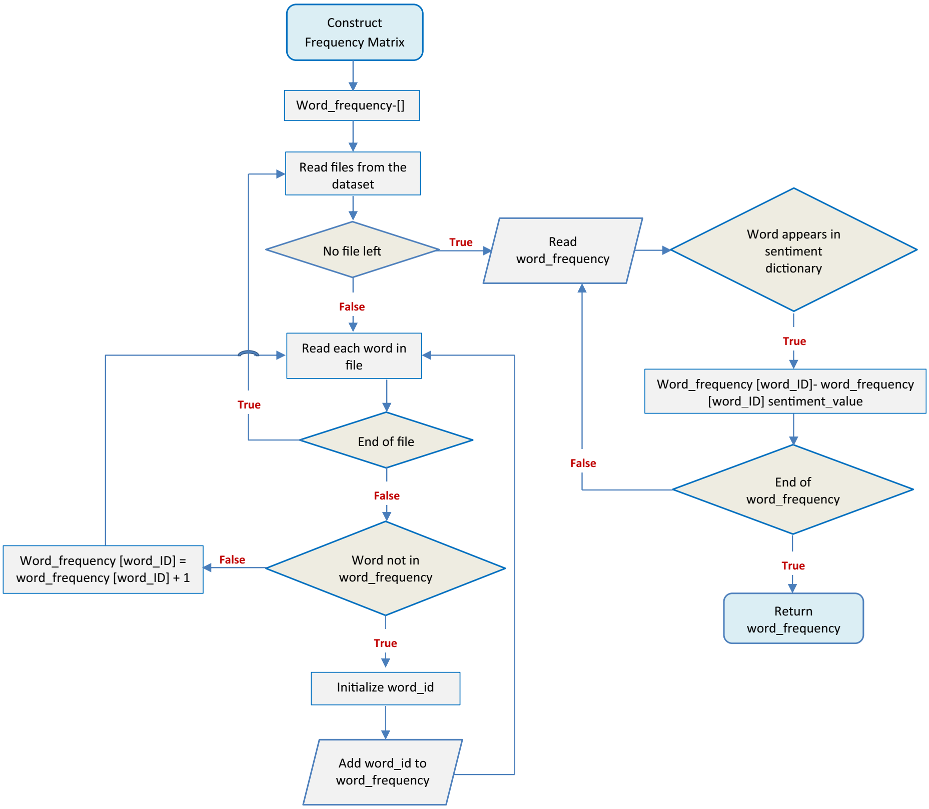
Construct Frequency Matrix
Word_frequency-[]
Read files from the dataset
No file left
True
Read word_frequency
Word appears in sentiment dictionary
False
True
Read each word in file
Word_frequency [word_ID]- word_frequency [word_ID] sentiment_value
True
End of file
False
End of word_frequency
False
True
Word_frequency [word_ID] = word_frequency [word_ID] + 1
False
Word not in word_frequency
Return word_frequency
True
Initialize word_id
Add word_id to word_frequency

Fig. 5 Flow diagram for build_vocab function

- To initialized the network parameters and manages the training process through Train_GloVe method. It utilizes co-occurrences data to re-calculate the biases and weights vector, in each iteration.

3.2 Two-state GRU mechanism

GRU is latest kind of traditional RNN which particularly has to be use for sequential modeling. However, a recurrent layer required the input vector $h_t \in \mathrm{R}^{\mathrm{n}}$ at each timestep t, and hidden state h_t by implementing the recurrent procedure describe in Eq. 7:

$$h_t = f(W_{xt} + Uh_{t-1} + b) \tag{7}$$

Where $\mathrm{W} \in R^{m \times n}$, $\mathrm{b} \in \mathrm{R}^{\mathrm{m} \times \mathrm{m}}$, $\mathrm{b} \in \mathrm{R}^{\mathrm{m}}$, weights matrix, and element-wise nonlinearity is represented by f. Training the long-term dependencies with RNN is very complicated due to the problem of vanishing gradient and exploding [22]. By applying the gating architecture, GRU can preserve memory significantly longer than traditional RNN. However, based on the recent studies," we find out that when GRU analyze a word it only includes the forward linguistic context, so it is impossible for GRU to learn the backward contexts. As the results,

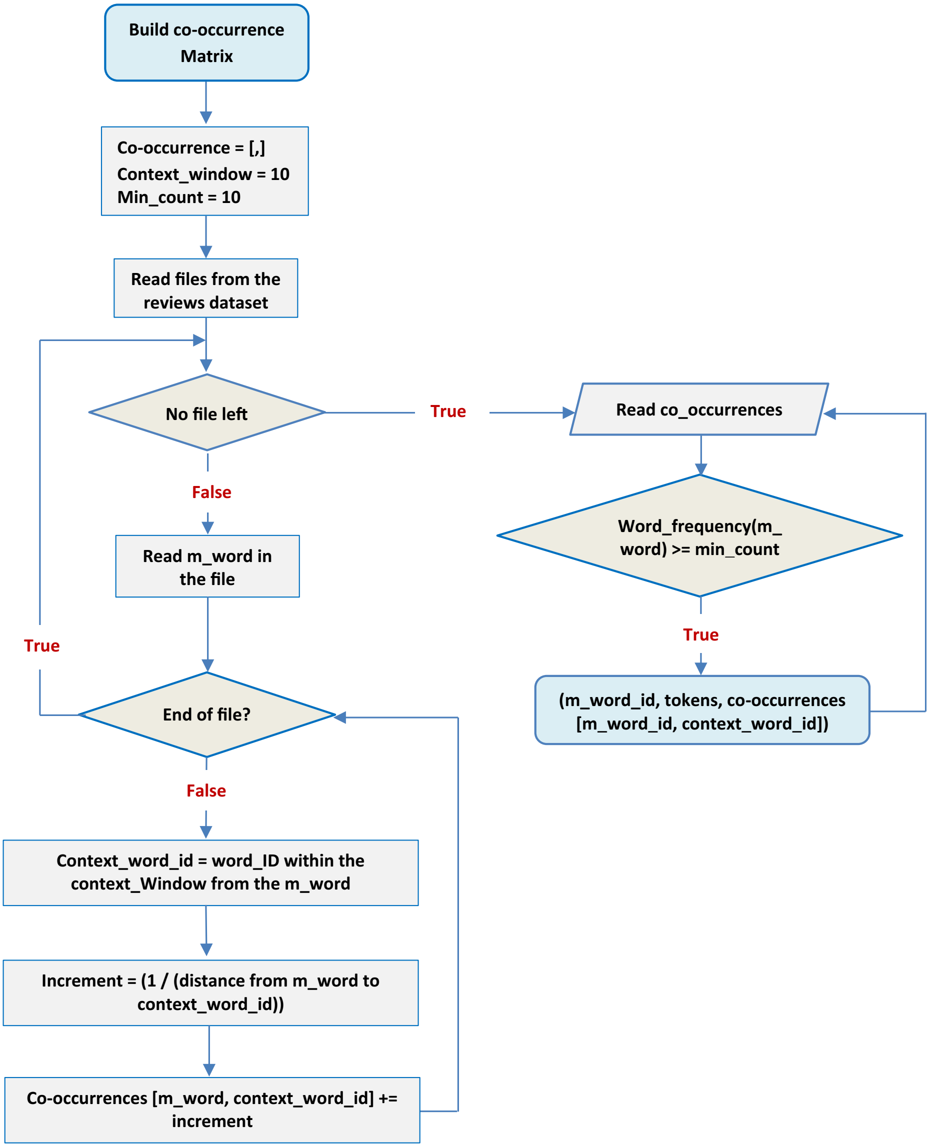
Build co-occurrence Matrix
Co-occurrence = [,]
Context_window = 10
Min_count = 10
Read files from the reviews dataset
No file left
True
False
Read m_word in the file
True
End of file?
False
Context_word_id = word_ID within the context_Window from the m_word
Increment = (1 / (distance from m_word to context_word_id))
Co-occurrences [m_word, context_word_id] += increment
Read co_occurrences
Word_frequency(m_word) >= min_count
True
(m_word_id, tokens, co-occurrences [m_word_id, context_word_id])

Fig. 6 Flow diagram for the build co-occurrence function

we also observed that in any language approach," which process of sentence is not affected only through forward context but also on the backward context.

Therefore, we proposed Two-State GRU (TS-GRU) to solve the above issue. The proposed TS-GRU model consists of two processes, one for positive pass known as "forward pass", and other for negative pass known as "backward pass" presented in Fig. 7. The two-state GRU can efficiently learn the words context through both directions. TS-GRU is inspired by the bidirectional recurrent neural networks (BRNNs) in [33]. It consists of two separate recurrent nets in the terms of forward passes (left to right) and backward passes (right to left) in the training process and finally both of them are combined to be the output layer. The Eqs. (8 to

11) presents the forward pass, while Eqs. (12 to 15) presents the backward pass of proposed TS-GRU model. The formulas for update gate z_t, reset gate r_t, candidate state $\hbar_t$, and final output activation state h_t of the forward and backward GRU are shown as a follows:

Forward pass.

$$\overrightarrow{z_t} = \sigma(\overrightarrow{W_{zx}}x_t) + \overrightarrow{U_{zh}}\overrightarrow{h_{t-1}} + \overrightarrow{b_z}) \tag{8}$$

$$\overrightarrow{r_t} = \sigma(\overrightarrow{W_{rx}}x_t) + \overrightarrow{U_{rh}}\overrightarrow{h_{t-1}} + \overrightarrow{b_r}) \tag{9}$$

$$\overrightarrow{\hat{h}_t} = \tanh(\overrightarrow{\mathrm{W}_{\hat{\mathrm{h}}}}\mathrm{x_t}) + \overrightarrow{\mathrm{r_t}}*\overrightarrow{\mathrm{U}_{\hat{\mathrm{h}}}}\overrightarrow{\mathrm{h_{t-1}}} + \overrightarrow{\mathrm{b}_{\hat{\mathrm{h}}}}) \tag{10}$$

$$\overrightarrow{h_t} = (1 - \overrightarrow{z_t})*\overrightarrow{h_{t-1}} + \overrightarrow{z_t*\hat{h}_t} \tag{11}$$

Additionally, we added backward pass in the proposed model to explore more valuable information.

Backward Pass:

$$\overleftarrow{z_t} = \sigma(\overleftarrow{W_{zx}}x_t) + \overleftarrow{U_{zh}}\overleftarrow{h_{t-1}} + \overleftarrow{b_z}) \tag{12}$$

$$\overleftarrow{r_t} = \sigma(\overleftarrow{W_{rx}}x_t) + \overleftarrow{U_{rh}}\overleftarrow{h_{t-1}} + \overleftarrow{b_r}) \tag{13}$$

$$\overleftarrow{\hat{h}_t} = \tanh(\overleftarrow{\mathrm{W}_{\hat{\mathrm{h}}}}\mathrm{x_t}) + \overleftarrow{\mathrm{r_t}}*\overleftarrow{\mathrm{U}_{\hat{\mathrm{h}}}}\overleftarrow{\mathrm{h_{t-1}}} + \overleftarrow{\mathrm{b}_{\hat{\mathrm{h}}}}) \tag{14}$$

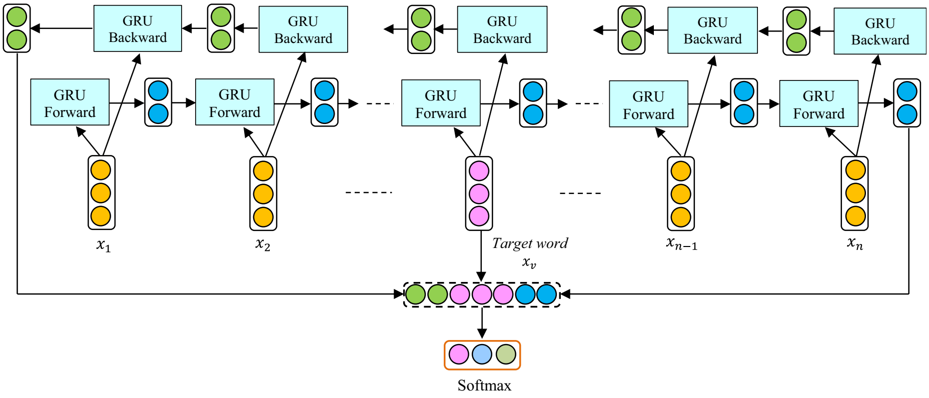

Fig. 7 The proposed "Two-State GRU architecture for sentiment analysis

Table 1 Numbers of parameters

Methods	Number of parameters
GRU	$3 \times (n^2 + nm + n)$
LSTM	$4 \times (n^2 + nm + n)$
Bi-LSTM	$8 \times (n^2 + nm + n)$
CNN-GRU	$11 \times (n^2 + nm + n)$
CNN-LSTM	$12 \times (n^2 + nm + n)$
FARNN-Att	$12 \times (n^2 + nm + n)$
BiGRU+CNN	$14 \times (n^2 + nm + n)$
Propose TS-GRU	$8 \times (\boldsymbol{n}^2 + \boldsymbol{nm} + \boldsymbol{n})$

$$\overleftarrow{h_t} = (1 - \overleftarrow{z_t})*\overleftarrow{h_{t-1}} + \overleftarrow{z_t * \hat{h}_t}) \tag{15}$$

The activation of a word at time t: $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ = for a random sequence $(x_1, x_2, \ldots, x_n)$ containing n words, at time t each word illustrated as a dimensional vector.

The forward GRU computes $\overrightarrow{h_t}$ which takes left-to-right contexts of the sentence while the backward GRU took right-to-left contexts $\overleftarrow{h_t}$ into consideration. Then forward and backward context representations are concatenated into a single context. In general, the complexity of an algorithm is computed by *O(W)*, while *W* is the number of estimated parameters of model. Two common variables used to compute *W* are the dimension of the input vector **m-dimension** and hidden layer dimension **n-dimension**. The Numbers of estimated parameters for deep learning approaches such as; GRU, LSTM, Bi-LSTM," CNN-GRU, FARNN-Att, CNN-LSTM, BiGRU + CNN and proposed TS-GRU are presented in Table 1.

The inner complexity of proposed TS-GRU is more than the complexity of conventional GRU due to the higher number of parameters. As a result, TS-GRU need more time and resources for execution than standard GRU and LSTM. While the proposed model is less complex as compared to others existing approaches such as Bi-LSTM, CNN-GRU, CNN-LSTM, FARNN-Att, BiGRU + CNN and thus require less execution time. However, our proposed mechanism is capable to extract useful information which significantly enhance the accuracy of the sentiment analysis.

3.3 Sequential mechanism by pre-feature attention TS-GRU

The Pre-feature attention TS-GRU mechanism which used to incorporate information from both word and proceeding for composing a first understanding in sentiment recognition. Generally, it is difficult for GRU and LSTM to extract important information for comprehensive sentiment classification, due to longer length of input sequences [31]. However, particular words also construct key contribution for more accurate classification. In feature-attention TS-GRU framework, the attention mechanism plays an effective role in order to extract the useful information in a long review of sentences [47], which assist to categorize emotions from the word-level appearance. Moreover, GRU control the stream of information inside the units through the gating mechanism, and two states GRU-framework concatenates information from the both words precedent and subsequent desirably contexts [26]. In addition, the pre-feature attention mechanism modeling includes a forward and a backward sub-state. The forward sub-state acquires the sequential words from embedding layer initializing to the end. The backward sub-state does the reverse computing as the forward sub-state do the forward computing.

Conventionally, at time step t of provided input word embedding x^k, based on previous and current forward candidate state $\overrightarrow{h}_{t-1}$ and $\overrightarrow{h}_t$, and its current and previous backward candidate state $\overleftarrow{h}_t$ and $\overleftarrow{h}_{t-1}$in the pre-feature attention TS-GRU are initialized in Eqs. (16, 17, 18) as:

$$\hbar_t = tanh\left(\overrightarrow{W_x^{(\hbar)}}x_t + \overrightarrow{r_t} * \overrightarrow{U^{(\hbar)}h_{t-1}} + \overrightarrow{b_\hbar}\right) \tag{16}$$

$$\hbar_t = tanh\left(\overleftarrow{W_x^{(\hbar)}x_t} + \overleftarrow{r_t} * \overleftarrow{U^{(\hbar)}h_{t-1}} + \overleftarrow{b_{(\hbar)}}\right) \tag{17}$$

$$\overline{h} = f\left(V\left[\overrightarrow{\widetilde{h}_t} : \overleftarrow{\widetilde{h}_t}\right] + k\right) \tag{18}$$

3.4 Word-Feature seizing by attention mechanism

In feature-attention process mechanism after receiving the final output by hidden state from first layer, we adopted the attention mechanism to support the architecture in determining sentiment polarity through concentrating on the helpful information from word-feature level. The comprehensive structure of attention mechanism has employed in our proposed mechanism is presented in Fig. 8, while Fig. 8 is also illustrating, the distribution of o_t^k at k^{th} time step generated by attention mechanism as follows:

$$o_t^k = \frac{exp\left(e_t^k\right)}{\sum_{i=1}^{m} exp\left(e_t^i\right)} \tag{19}$$

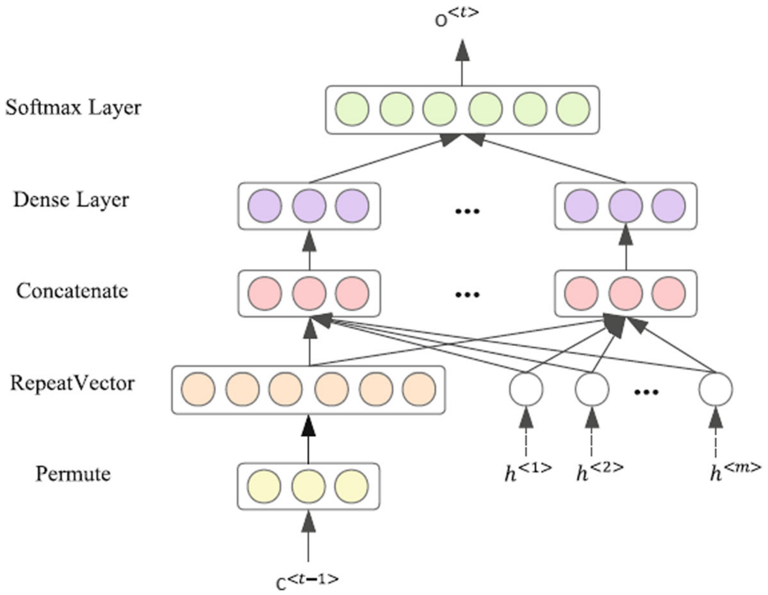

Fig. 8 The detailed structure of attention mechanism

Where score function is represented by e_t^k for the memory cell of k^{th} time step, and score function presented by e_t^k in Eq. 20:

$$e_t^k = \left[c_{t-1}^T h_1, c_{t-1}^T h_2, \ldots, c_{t-1}^T h_m\right] \tag{20}$$

Where $h_k$is the hidden unit of pre-feature attention TS-GRU and memory cell represented by c_{t-1} at k^{th} time-step in post-feature attention GRU and finally, obtained attention output through Eq. (21) as follows:

$$o_t = \sum_{k=1}^{m} o_t^k h_k \tag{21}$$

3.5 Post-feature attention GRU

Based on the feature attention mechanism, a post-feature attention TS-GRU followed by the word-feature seizing mechanism to capture the sentence level information selected through repetition of learns the sentence intentionally similar to human. Moreover, in the second phase feature attention TS-GRU, we adapted a post-feature attention GRU imitating the decoded function is extracts the predicted features generated by pre-feature attention GRU and the attention mechanism layer. The all major equations of post-features attention TS-GRU are equivalent with the standard Bi-GRU, without candidate state (22) as a follow:

$$\hbar_t = tanh\left(W_x^{(\hbar)} x_t + r_t * U^{(\hbar)} h_{t-1} + b_{(\hbar)}\right) \tag{22}$$

In this way, the output features vector of post-feature attention TS-GRU are transform to the dense layer as a sentence representation and finally, we used a softmax classifier for final classification in order to predict class label ("positive, negative") of the sentiment analysis datasets.

3.6 Flowchart

This study also demonstrate the flowchart of the developed mechanism for solving sentiment analysis. Figure 9 illustrates a flowchart of sentiment analysis which consists four main phases. The first phase consists on data preprocessing for sentiment analysis objective.

The second phase refers to the parameter initialization of the proposed TS-GRU model. In this stage, we used embedding layer for changing the words context into real-valued vector which capture semantic, syntactic and morphological information about words. For this purpose we employed the pre-trained GloVe [27] technique at the words embedding layer in a sequence to convert each word into a real value vector, and then we implement the feature-attention two state GRU based mechanism for extracting more informative features. The third phase checks the training error according to threshold. Finally, the fourth phase focuses on testing verification process. To evaluate performance, we have used accuracy, execution time and error rate as metrics for the text classification tasks. It is observed that the features extraction and selection are very essential for improving the accuracy of the model, due to its directly affect the final performance of the model. Therefore, we proposed Feature Attention Two State GRU mechamism for sentiment analysis. The detailed architecture of feature attention TS-GRU mechanism is presented in Fig. 10.

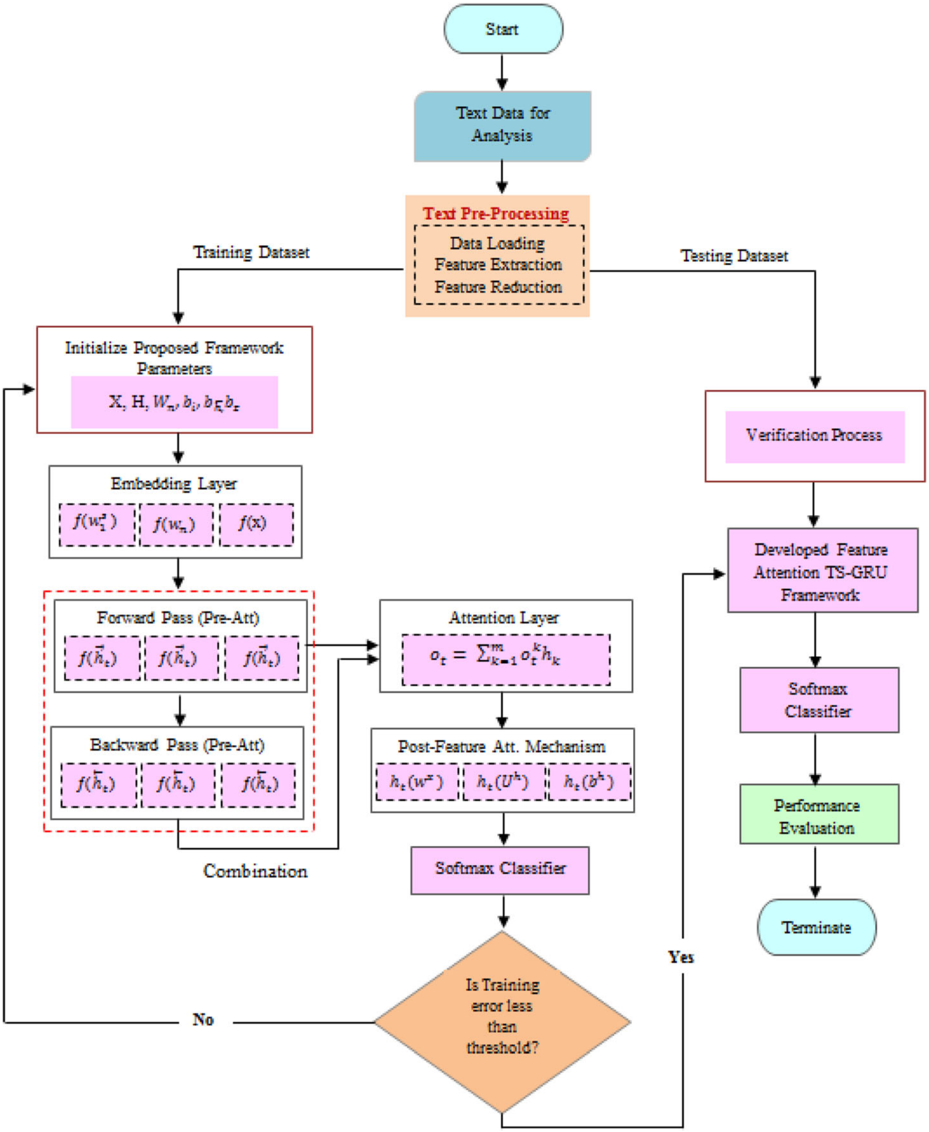

Fig. 9 Flowchart of network sentiment analysis

4 Experimental setup

All the simulations of this research are implemented on Intel core-i7-3770CPU @ 3.40 GHz, and 4GB of RAM with Window7 operating system. For data preprocessing and analysis, we used Python 3.7 compiler and anaconda as the development environment, with required libraries TensorFlow 1.14 and Keras 2.3. Furthermore, we provided the brief description of datasets and implementation hyper parameters setting in order to optimize our developed mechanism in the following subsections.

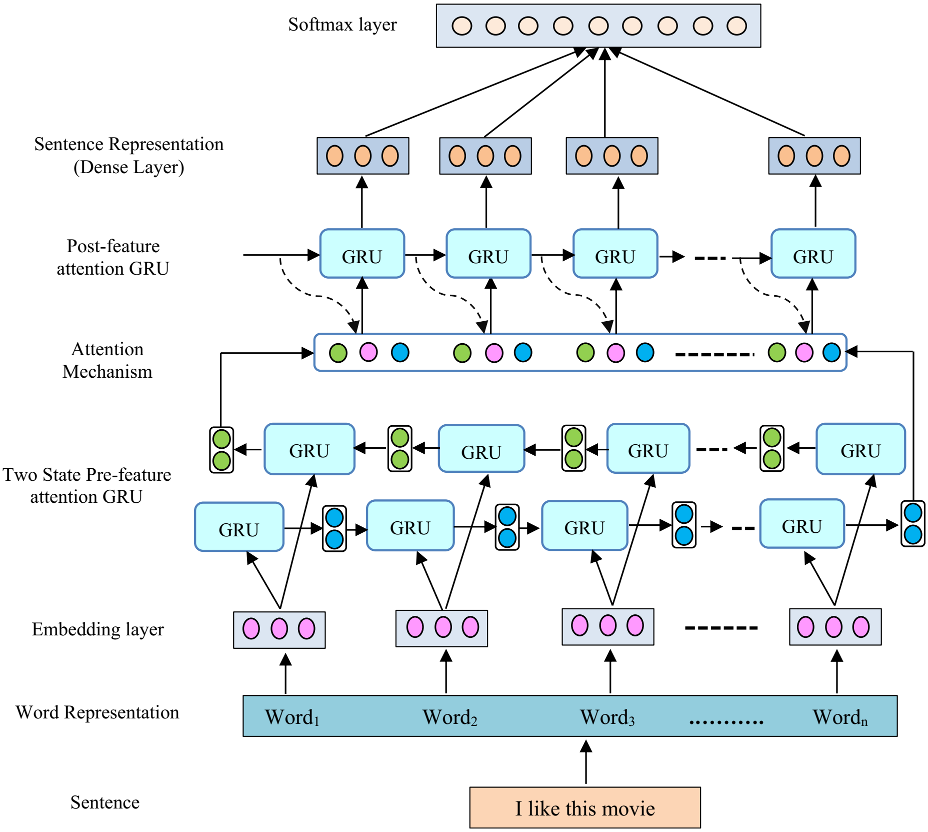

Fig. 10 The overall architecture of proposed feature-attention TS-GRU

4.1 Sentiment analysis datasets

In this sub-section, we managed to conduct the experiment on three selected freely accessible datasets which are compressed of public opinions in the term of sentiment analysis about movies included, Stanford Internet Movie "Review dataset (IMDB), Stanford Sentiment Treebank dataset (SST2) and Movie Review (MR). The fundamental task of these datasets is to predict whether a movie belongs to positive or negative class, and the statistical properties of three datasets are presented in Table 2.

Table 2 Description of statistical properties of all datasets

Details	IMDB	MR	SST2
Train	37,500	8655	6920
Val	CV	CV	872
Test	12,500	1064	1821
Max.length	2385	56	52
Ave.length	220.8	19.5	18.4
V-size	88,642	18,983	17,674
D-size	50,000	10,662	9618
#Classes	2	2	2

The **IMDB** dataset was initially introduced by Andrew et al. [22] in 2011 for sentiment analysis purpose. It is freely available dataset that consist 50,000 binary reviews, and which divided by 25,000 are positive-labelled reviews and 25,000 are negative-labelled reviews. The core point of IMDB dataset, which mostly part of sentiment reviews samples, is compressed in several sentences. We observed average length of this dataset is 220.87 words, with the longest review of 2385 tokens.

The **SST2** dataset was proposed by Pang et al. [23] in 2005 and further extended by Socher et al. [36] in 2013 as benchmark for sentiment classification, and it consist of 9,618 review samples. It is not a balanced dataset and is a diverse form of the MR dataset. In this case, the average length is 18.4 words with a maximum review of 52 tokens.

The **MR** dataset was initially presented by Lee *et al* [25] in 2005as a benchmark for the sentiment analysis tasks. It is a stable dataset that is contain 10,662 short reviews and divided into 5,295 positive reviews while 5,295 portray negative reviews. The average length of complete sentence as observed is 19.5 tokens while the maximum reviews noted as 56 tokens. Table 2 show the description of statistical properties of all datasets where #Classes: number of target classes, Ave.length: average sentence length, Max.length: maximum sentence length, train/val/test: train/validation/test set size, V-size: vocabulary size, D-size: dataset size," CV: cross validation.

4.2 Hyperparameters setting

Deep learning based approaches have the ability of learning complex relationships between inputs and outputs [15]. In our experiment, we applied Adam optimizer to set their default optimal parameters setting with learning rate 0.001 and decay factor is 0.9. For a fair comparative analysis, a few preprocessing steps are performed in order to improve the quality of dataset, and secondly, all the word vectors are initialized through GloVe method selected for sentiment analysis. However, during the training process many connections are involved as a result of sampling noise; while it did not exist in the real test data. This problem may conduct to overfitting and minimize the prediction ability of the network [15]. For this issue, we applied the dropout method in order to reduce the overfitting in TS-GRU layer with 128 memory units for each forward and backward direction. After combining the forward GRU and backward GRU, another dropout layer was added to reduce 50% of the input to handle with the overfitting issue. Moreover, in this study, we used Cross Entropy with L_2 regularization as the loss function, which referred in Eq. (23) as follows:

$$J(w,b) = -\frac{1}{2}\sum_{i=1}^{m}[y_i log\hat{y}_i + (1-y_i)log(1-\hat{y}_i)] + \frac{\lambda}{2m}\sum_{l=1}^{m} \parallel w \parallel_F^2 \tag{23}$$

Where y_i is refer ground truth; and classification probability for each class represented by $\hat{y}_i$. We set $w = 0.001$, of Frobenius norm value by compressing L_2, which is the coefficient for L_2. During the training process, the result presents the L_2 regularization and dropout method can perform better to avoid overfitting. Table 3 provides the optimal values of hyperparameters, which have applied for the training of proposed framework.

The detail description of each layer in the proposed TS-GRU is presented in Fig. 11. In the input layer, we set the limit of length 300 words, according to the review length because review document contains different review length, clipping long movie reviews documents and padding shorter review documents are padding with zero values. The next layer is the TS-

Table 3 Optimal hyperparameters of the Proposed Framework

Hyperparameter	IMDB dataset	SST2 dataset	MR dataset
Optimizer	Adam	Adam	Adam
Loss Function	Cross-entropy	Cross-entropy	Cross-entropy
Learning Rate	0.001	0.001	0.001
Train Approach	CV	Train/Val/Test	CV
Max.length	400	50	45
Batch size	64	64	64
Dropout	0.2	0.2	0.2
Epochs	60	60	60
Learning rate decay	0.01	0.01	0.01
Momentum	0.9	0.9	0.9
Regularizer	L2	L2	L2

GRU layer with 128 memory units for each forward and backward pass. For the purpose of minimize the overfitting; we used dropout strategy for GRU layer in the each iteration.

Therefore, we set the 30% input Dropout-W while was set to 30% hidden state Dropout-U. After combing the forward and backward GRU, one more dropout was inserted in the layer fir further minimized overfitting issue to drop 48% of the input. Lastly, we used Dense layer for the sentence representation in the form of positive or negative prediction, and for this purpose applied sigmoid activation function to determine 0 or 1 for classified two classes.

5 Results and discussion

In this section, we carried out the many experiments on three movie review datasets to investigate the robustness of the proposed mechanism and offer fair comparison with existing deep learning approaches. We verified the effectiveness of our proposed model based on the following evaluation parameters.

5.1 Maximum length

For efficient training of the network, we need to maintain same length of all input samples as known maximum. Therefore, we fix the length of input samples according to the sentence of the text data. Thus, zeros will be padded, if the input length of the data is

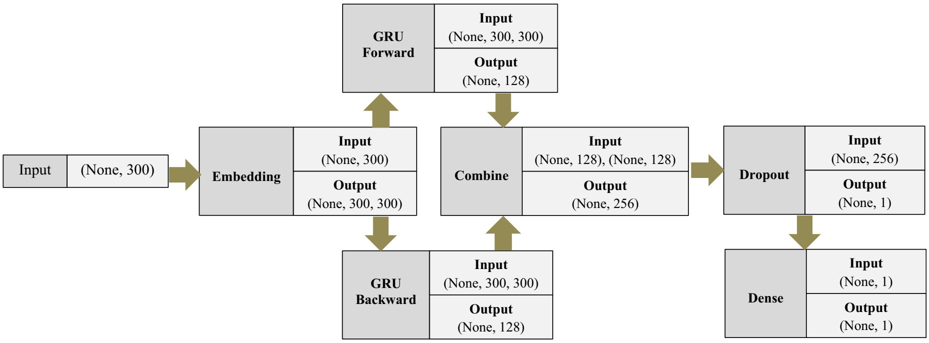

Fig. 11 The configuration of TS-GRU network

less than the maximum length, otherwise, input samples will be truncated. In preference, we discard maximum informative features based on low maximum length [41]. The optimal maximum lengths for MR, IMDB and SST2 datasets are 45,400, and 50 tokens, respectively. Figures 12 and 13 present the result of the model and noticed that increasing the input review length does not significant impact on the model performance after maximum length.

5.2 Epochs

In the learning algorithm epochs illustrate the number of times that performs across the complete training of dataset [15]. Number of epochs increases the generalization capability of the network; however, too many epochs are produced overfitting. Therefore, a proper structure of epochs is needed to take with general behavior and overfitting of the network. It can be notice in Fig. 14, in which the performance of proposed approach in the term of accuracy is not increasing after a particular epoch in every dataset. Specially, we found the optimal values of epochs for MR, IMDB, and SST2 datasets are 60, 50, and 60 respectively.

5.3 Evaluation Matrix

The complete performance of the network is evaluated by computing the confusion matrix, such as error matrix or contingency table. To observe the experimental results, the Table 4 presents the confusion matrix components which are included four terms, are called True Positive (*TP*), True Negative (*TN*), indicate the correctly classification for the relevant class while False Positive (*FP*), and False Negative (*FN*) are determine the false classification for the relevant class. In this study, we adopt the four evaluation parameters from confusion matrix in order to investigate the effectiveness of our developed mechanism for sentiment analysis. These four evaluation parameters are precision, recall, *F*1 score, and accuracy, which are briefly explained as follows:

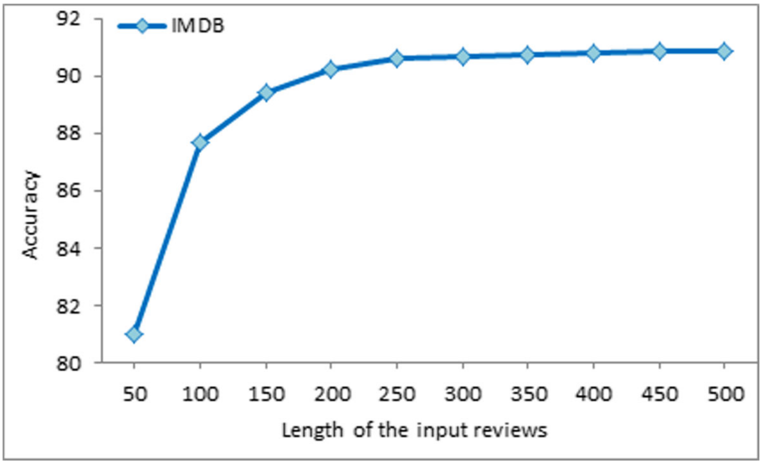

Fig. 12 The accuracy of proposed mechanism against the selected length of the input reviews on IMDB dataset

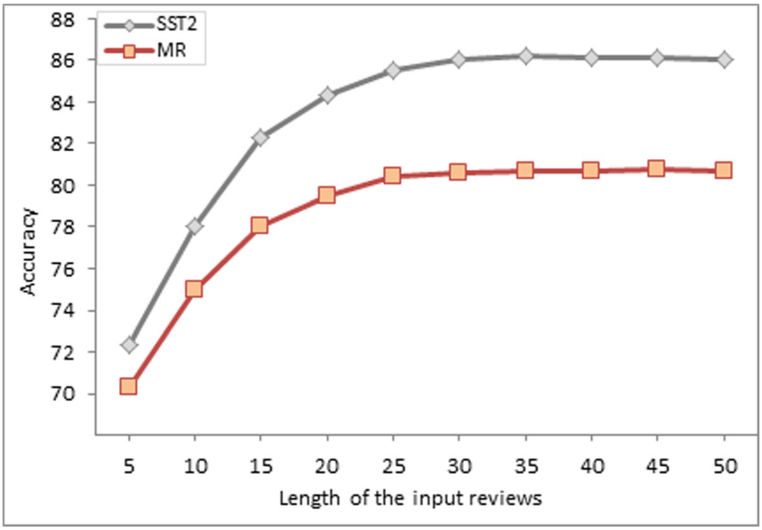

Fig. 13 The accuracy of proposed mechanism against the selected length of the input reviews on SST2 and MR datasets

Accuracy: its complete correctness of the model and is calculated as ratio of correctly classified instances (*TP*) divided by the total numbers of instances (*TP* + *FP*) on the entire dataset. However, the accuracy cannot provide us a complete assessment of the model. It is defined by Eq. (24) as:

$$\text{Accuracy} = \frac{\text{TP}_{(n)}^{(m)} + \text{TN}_{(n)}^{(m)}}{\text{TP}_{(n)}^{(m)} + \text{FP}_{(n)}^{(m)} + \text{FN}_{(n)}^{(m)} + \text{TN}_{(n)}^{(m)}} \tag{24}$$

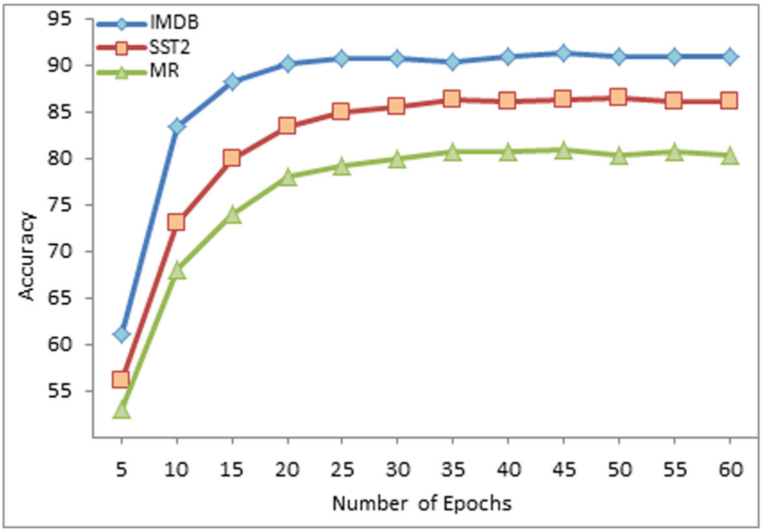

Fig. 14 The accuracy of proposed mechanism based on the number of epochs

Precision and recall are the two most extensively employed evaluation parameters beside accuracy and it evaluate the exactness of a model. Higher precision means that most of the instances which were classified into correctly positive instances, while recall or sensitivity defined as the proportion of correctly classified the instances (*TP*) to the total number of actual instances (*TP* + *FN*) and it defined in Eqs. (25, 26) as follow:

$$\text{Precision} = \frac{\text{TP}_{(\text{n})}^{(\text{m})}}{\text{TP}_{(\text{n})}^{(\text{m})} + \text{FP}_{(\text{n})}^{(\text{m})}} \tag{25}$$

$$\text{Recall} = \frac{\text{TP}_{(\text{n})}^{(\text{m})}}{\text{TP}_{(\text{n})}^{(\text{m})} + \text{FN}_{(\text{n})}^{(\text{m})}} \tag{26}$$

F1 score: it calculates the weighted average of precision and recall or sensitivity, and it commonly applied for network optimization toward either precision or recall. F1 is defined in Eq. (27) as:

$$\text{F1score} = \frac{2 \times \text{Precision}_{(\text{n})}^{(\text{m})} \times \text{Recall}_{(\text{n})}^{(\text{m})}}{\text{Precision}_{(\text{n})}^{(\text{m})} + \text{Recall}_{(\text{n})}^{(\text{m})}} \tag{27}$$

All the evaluation matrix of our proposed framework is given in Table 4.

5.4 IMDB results

For IMDB dataset, we evaluated the competitiveness of our proposed mechanism are demonstrated in Table 5. The proposed mechanism achieved precision scores for positive and negative reviews are 90.74% and 90.70%, accordingly, while the recall reviews for negative and positive noted as 90.60% and 90.77%, respectively. Similarly, based on the precision and recall finally our proposed mechanism got the F1 score of 90.695% along with an accuracy of 90.85%. Therefore, in this study we compared the results of IMDB dataset in the term of accuracy with state-of-the-art existing approaches which are contain complex architectures. On the other hand, Long et al. [20] proposed the cognition based attention (CBA) approach and integrated with local text context-based attention (LA) networks, based on the LSTM model and obtained the superior accuracy of 90.10% on IMDB dataset. However, in this case the researcher investigated two distinct attention mechanisms and combined with LSTM approach, but it could not achieve satisfy results. Although, our proposed mechanism provided better accuracy as compared to [20] even it is not need as such ideas. In the same way, C. C. Jose and P. Mohammad Taher [4] obtained the accuracy of 88.9% on IMDB dataset by examined the hybrid impact of CNN and LSTM approaches. Even this research, applied various cleaning procedures through the combination of CNN and LSTM models, it also could not achieve superior results. On the other side, in our research we used two-state GRU with

Table 4 Performance of the developed mechanism for all datasets

Datasets	Normalized Confusion Matrices			Performance Evaluation Parameters			Average Values		
	Predict class → *Actual class ↓*	0	1	Precision (%)	Recall (%)	F1 Score (%)	Test size	Accuracy (%)	F1 score (%)
IMDB	0	0.9068	0.0950	90.74	90.60	90.66	CV	**90.85**	**90.695**
	1	0.0943	0.0975	90.70	90.77	90.73	CV		
MR	0	0.8147	0.1905	80.06	81.46	80.75	CV	**80.72**	**80.640**
	1	0.2045	0.7967	81.42	79.67	80.53	CV		
SST2	0	0.8371	0.1657	87.91	83.99	85.90	912	**86.51**	**86.345**
	1	0.1221	0.8837	85.18	88.48	86.79	909		

The bold entries are indicating the highest results of proposed and comparative technique

feature attention mechanism as a key layer and obtained comparatively 1:95% points higher accuracy. Furthermore, Fu et al. [9] used to investigate ALE-LSTM and WALE-LSTM approaches and provided 89:30% and 89:50% of accuracy on the IMDB dataset. Consequently, this research employed various attention mechanism but attained only reasonable results. Especially, our proposed mechanism comparatively got 1:35 and 1:15% points better accuracy as compared to existing ALE-LSTM and WALE-LSTM approaches respectively. Moreover, for sentiment analysis on the IMDB dataset, the combination of CBOW language model with deep CNN model proposed by *Lio Bing* [19] to construct a continuous distributed word level representation and reported an accuracy of 87.20% respectively. Finally, Yaohong et al. [21] proposed a combine feature-based adversarial RNN model with attention framework (FARNN-Att) and presented an accuracy of 89:22% on the IMDB dataset. Here, the researchers employed the combine adversarial training process as a regularization approach to trained the BiLSTM network with attention mechanism but could not achieve remarkable results.

However, we trained Two-State GRU (TS-GRU) model based on feature attention mechanism and achieved an excellent result than existing deep learning approaches. In summary, our proposed feature attention based Two-State GRU (TS-GRU) mechanism offered remarkable results on the IMDb dataset as compared to state-of-the-art research, [4, 9, 19–21] with comparatively complex architectures, as presented in Table 5.

Table 5 Comparison results of proposed mechanism with existing approaches on IMDB dataset

Models	Model complexity	Accuracy (%)
FARNN-Att [21]	Attention framework and adversarial training with BiLSTM	89.22
WALE-LSTM [9]	Hybrid of lexicon and attention mechanism along with LSTM	89.50
CNN+LSTM [4]	Fusion of CNN and LSTM with various cleaning process	88.90
CBOW+D-CNN [19]	Combine CBOW language model with deep CNN model to constructed a continuous distributed word level representation	87.20
LSTM+CBA+LA [20]	Combination of two various attention framework with LSTM	90.10
Proposed	Feature Attention Mechanism with Two-State GRU	**90.85**

The bold entries are indicating the highest results of proposed and comparative technique

5.5 MR results

We evaluate the results of our proposed feature attention Two-State GRU (TS-GRU) on MR dataset is presented in Table 6. The proposed mechanism has attained precision assessments for positive and negative observations are reported as 80.06% and 81.42%, respectively. Similarly, the recall measures for positive and negative sentiments are represented as 81.46% and 79.67%, respectively. Finally, based on the precision and recall, our proposed mechanism got the F1 score of 80.640% along with an accuracy of 80.72%. Similar to IMDB results, in this study we compared the effectiveness of MR results in the term of accuracy with traditionally published research having fusion architectures. On the other hand, for the sentiment analysis of movie reviews in the term of binary prediction dataset, Socher et al. [37] proposed matrix-vector recursive neural network (MV-RNN) approach and achieved an accuracy of 79:00% on MR dataset. The MV-RNN architecture depends on the parse trees structure, where architecture shows to increase complexity with longer analysis. In contrast, our proposed mechanism performs independently of the input length and reported excellent results 1:72 ratio points than the MV-RNN model. Also, on MR dataset, Fu et al. [9] suggested attention framework (ALE-LSTM) based on a lexicon-improved LSTM approach and obtain 80% accuracy. In addition, the authors introduced word embedding based ALE-LSTM (WALE-LSTM) architecture and achieved 79:9% on the movie reviews dataset. In this case, the authors merged both approaches with different concepts, but could not receive satisfied result, while our proposed model provided better results than ALE-LSTM and WALE-LSTM, on MR dataset. Similarly, Zhang et al. [44] proposed a new leveraging series based architecture along with the concatenation of bidirectional GRU (BiGRU) and CNN (BiGRU + CNN) and referred 78.30% accuracy on MR dataset, while our model given 2.42% better accuracy than [44]. In the same way, the encountering multi-level and multi-type features approaches are proposed by Usama et al.[39] through the combination of GRU, LSTM and CNN approaches and shown the accuracies of 79:80% and 80:20%, respectively. Finally, Qian et al. [29] introduced the sentence level annotation based on Linguistic Regularizer (LR) combine with bidirectional LSTM to addresses the sentient shifting effect of sentiment and reported an accuracy of 82.10% on MR dataset. We observed that LR-Bi-LSTM approach outperform our proposed mechanism by an accuracy of 1:38% on SST2 dataset. It is essential to describe that LR-Bi-LSTM model achieved excellent result as compared to our proposed model but its basic structure is comparatively more complex than our model is mention in [29].

In summary, our proposed feature attention Two-State GRU (TS-GRU) based mechanism offered significant results on the IMDb dataset as compared to recent existing studies [9, 29, 37, 39, 44] with comparatively complicated architectures, as shown in Table 6.

5.6 SST2 results

To investigate the competitiveness of our proposed framework for SST2 dataset are described in Table 7. In this case, the proposed mechanism has obtained precision measures for positive and negative sentiments are referred as 87.91% and 85.18%, respectively. Similarly, the recall measures for positive and negative observations are represented as 83.99% and 88.48%, accordingly. Subsequently, the proposed mechanism provided an accuracy of 86.51% along with the F1 measures of 86.345% on SST2 dataset. Similar to IMDB and MR results, this study fairly offer comparison competency of proposed mechanism on SST2 dataset in the term of accuracy with recent state-of-the-art studies based on complex architectures. In particular,

Table 6 Comparison results of proposed mechanism with existing approaches on MR dataset

Models	Model Complexity	Accuracy (%)
BiGRU+CNN [44]	Serial fusion of BiGRU and CNN methodologies	78.30
MV-RNN [37]	Combination of parse trees and RNN network	79.00
WALE+LSTM [9]	Fusion of lexicon and attention framework conjointly with LSTM	79.90
CNN-GRU-multilevel and multitype fusion [39]	Combination of multilevel and multi-type features based on CNN and GRU models	80.20
LR-Bi-LSTM [29]	Presented the sentence level annotation based on Linguistic Regularizer combine with bidirectional LSTM	**82.10**
Proposed	Feature Attention Mechanism with Two-State GRU	80.72

The bold entries are indicating the highest results of proposed and comparative technique

the Recursive Neural Tensor Networks (RNTN) approaches presented by Socher et al. [36] which obtained 85:40% accuracy on the SST2 dataset. However, the RNTN approaches got remarkable results, due to relies on the architecture of parse trees [37], where the architecture increases complexity with long-term analysis. Furthermore, these mentioned sparse trees architecture are not consequently accessible for all dataset. In contrast, our proposed mechanism performs independently of the input length and reported 1:11% higher accuracy than the RNTN model. In the same way, the system based network on the combination of bidirectional GRU (BiGRU) along with CNN (BiGRU + CNN) has proposed by Zhang et al. [44] and provided 85:40% accuracy on SST2 dataset. Contrary to BiGRU + CNN, our proposed mechanism relies on Two-State GRU (TS-GRU) only and keeps recorded 1:11% better accuracy than BiGRU + CNN. Similarly, on SSTb dataset, Usama et al. [39] proposed the multitype features selection based framework, which are parallel combination of CNN and LSTM approaches and acknowledged 85:70% accuracy. Furthermore, in the same study, authors investigated two another multitype fusions of deep learning approaches with combination of CNN and GRU and other one is the combination of CNN and GRU along with multilevel and multitype features selection of fusion-rand [39] and reported an accuracy of 86.20% and 85.60% on SSTb dataset respectively. Contrary to the studies of [39], our proposed mechanism obtained better accuracy than [39] with significantly less nodes.

Finally, Yang et al. [43] investigated the together impact of capsule models in combination with LSTM (Capsule-LSTM) and achieved 86.40% accuracy on SST2 dataset, while our

Table 7 Result comparisons of SST2 dataset with other methods

Models	Model Complexity	Accuracy (%)
RNTN [36]	Fusion of parse and RNTN network	85.40
CNN-GRU-multilevel & multitype fusion-rand [39]	Parallel combination-based architecture of CNN and GRU with multilevel and multitype fusion-rand	85.60
CNN-LSTM-multitype fusion [39]	Parallel combination of CNN and LSTM with multi-type features selection	85.70
CNN-GRU-multitype fusion [39]	Parallel fusion of CNN and GRU with multi-type features selection	86.20
BiGRU+CNN [44]	Pipelining of BiGRU and CNN models	85.40
Capsule-LSTM [43]	Merging of capsule approaches with LSTM network	86.40
Proposed	Feature Attention Mechanism with Two-State GRU	**86.51**

The bold entries are indicating the highest results of proposed and comparative technique proposed model provided better accuracy as compared to Capsule-LSTM. In summary, it is inferred that our proposed feature attention Two-State GRU (TS-GRU) based mechanism offered remarkable results on the IMDb dataset are compared recently published studies [36, 39, 43, 44] with complex architectures, as shown in Table 7.

6 Conclusion and future scope

This paper has proposed an effective classification deep learning approach for sentiment analysis tasks by integrating Two-State Gated Recurrent Unit ("TS-GRU) model through Feature-Attention Mechanism. The proposed mechanism fully investigates the potential of the word embedding layer and examined the sentiment polarity from the aspects of sentential-sequence pattern and word feature seizing to predict the sentiment reviews. We designed a novel feature-attention mechanism to capture more informative feature representation via Pre & Post-feature attention layer. Furthermore, a post-feature attention GRU was inserted for imitation the function of the decoder to extract the predicted features learned on the pre-feature attention TS-GRU and the attention layer. This study also demonstrated the importance of concentrating on the core information of a sequential input from the word-feature level. We conducted the experiment on three benchmark datasets about movie reviews, included IMDB, MR and SSTb respectively and efficiently predict the sentiments polarity by using less parameters computational mechanism. The proposed feature-attention TS-GRU mechanism achieved excellent results and outperformed various existing approaches in the terms of accuracy, especially on IMDB and SSTb datasets. Therefore, in this study the results illustrated that it can possible to employ less complicated architecture to attain the similar level of classification performance. However, our developed mechanism could not perform well a few recently published frameworks on SSTb dataset.

In the future work, we plan to apply multi class for sentiment analysis along with multitype technique. Also, we can use the two-state GRU model integrated with Bidirectional Long Short Term Memory (Bi-LSTM) for the sentential-modeling. In addition," we develop to design more efficient and versatile attention architecture in the word-feature seizing level and also minimize the computational cost of the developed mechanism.

Acknowledgements The authors would like to thanks Ministry of Education Malaysia, Universiti Tun Hussein Onn Malaysia and Research Management Center (RMC).

Declarations

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

References

1. Acharjya DP, Kauser AP (2016) Acharjya DP, Kauser AP (2016) A survey on big data analytics: challenges, open research issues and tools. Int J Adv Comput Sci Appl 7(2):511–518
2. Balyan R, McCarthy KS, McNamara DS (2020) Applying natural language processing and hierarchical machine learning approaches to text difficulty classification. Int J Artif Intell Educ 30(3):337–370
3. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166

4. Camacho-Collados J, Pilehvar MT (2018) On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. arXiv Prepr arXiv170701780:40–46
5. Cho K et al (2014) On the properties of neural machine translation: Encoder–decoder approaches. arXiv 5:1–9
6. Cho K, Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv [Internet]: (September):1–15. Available from: http://arxiv.org/abs/1406.1078
7. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. 1–9. Available from: http://arxiv.org/abs/1412.3555
8. Do HH, Prasad PWC, Maag A, Alsadoon A (2019) Deep learning for aspect-based sentiment analysis: A comparative review. Expert Syst Appl [Internet] 118:272–99. Available from: https://doi.org/10.1016/j.eswa.2018.10.003
9. Fu X, Yang J, Li J, Fang M, Wang H (2018) Lexicon-enhanced LSTM with attention for general sentiment analysis. IEEE Access 6(c):71884–71891
10. Ghazali R, Husaini NA, Ismail LH, Herawan T, Hassim YMM (2014) The performance of a Recurrent HONN for temperature time series prediction. In: 2014 International Joint Conference on Neural Networks (IJCNN) (July). IEEE, Beijing, pp 518–524
11. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
12. Hourri S, Nikolov NS, Kharroubi J (2021) Convolutional neural network vectors for speaker recognition. Int J Speech Technol 24(2):389–400
13. Hunsinger S (2018) Text Messaging Today: A Longitudinal Study of Variables Influencing Text Messaging from 2009 to 2016. J Inform Syst Appl Res 11(3):25
14. Kalyanathaya KP, Akila D, Rajesh P (2019) Advances in natural language processing–a survey of current research trends, development tools and industry applications. Int J Recent Technol Eng 7:199–202
15. Ketkar N (2017) Stochastic gradient descent. In: Deep learning with Python Apress, Berkeley, vol. 1, pp 113–132
16. Kumar RS, Devaraj AFS, Rajeswari M, Julie EG, Robinson YH, Shanmuganathan V (2021) Exploration of sentiment analysis and legitimate artistry for opinion mining. Multimed Tools Appl 81:11989–12004. https://doi.org/10.1007/s11042-020-10480-w
17. Lee OJ, Jung JJ (2020) Story embedding: Learning distributed representations of stories based on character networks. Artif Intell 281:103235
18. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. arXiv: 1506. 00019v4 [cs. LG] 17 Oct 2015.1–38
19. Liu B (2020) Text sentiment analysis based on CBOW model and deep learning in big data environment. J Ambient Intell Humaniz Comput [Internet] 11(2):451–8. Available from: https://doi.org/10.1007/s12652-018-1095-6
20. Long Y, Lu Q, Xiang R, Li M, Huang CR (2017) A cognition based attention model for sentiment analysis. EMNLP 2017 - Conf Empir Methods Nat Lang Process Proc, 462–71
21. Ma Y, Fan H, Zhao C (2019) Feature-based fusion adversarial recurrent neural networks for text sentiment classification. IEEE Access 7:132542–132551
22. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. Proc 49th Annu Meet Assoc Computing Linguist Hum Lang Technol 1:142–150
23. Pang B, Lee L (2005) Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. arXiv preprint cs/0506075.
24. Parimala M, Swarna PRM, Praveen KRM, Lal CC, Kumar PR, Khan S (2021) Spatiotemporal-based sentiment analysis on tweets for risk assessment of event using deep learning approach. Software: Pract Experience 51(3):550–570
25. Parkhe V, Biswas B (2016) Sentiment analysis of movie reviews: finding most important movie aspects using driving factors. Soft Comput 20(9):3373–3379
26. Peng P, Zhang W, Zhang Y, Xu Y, Wang H, Zhang H (2020) Cost sensitive active learning using bidirectional gated recurrent neural networks for imbalanced fault diagnosis. Neurocomputing 407:232–245
27. 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) (October), Doha, Qatar, pp 1532–1543
28. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP et al (2018) A survey on deep learning: Algorithms, techniques, and applications. ACM Comput Surv 51(5):23–51
29. Qian Q, Huang M, Lei J, Zhu X (2016) Linguistically regularized lstms for sentiment classification. arXiv preprint arXiv:1611.03949
30. Rahman S, Chakraborty P (2021) Bangla document classification using deep recurrent neural network with BiLSTM. In: Proceedings of International Conference on Machine Intelligence and Data Science Applications. Springer, Singapore, pp 507–519

31. Sachin S, Tripathi A, Mahajan N, Aggarwal S, Nagrath P (2020) Sentiment analysis using gated recurrent neural networks. SN Comput Sci [Internet] 1(2):1–13. Available from: https://doi.org/10.1007/s42979-020-0076-y
32. Say B (2021) A unified framework for planning with learned neural network transition models. In: Proceedings of the AAAI Conference on Artificial Intelligence 35(6): 5016–5024
33. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11): 2673–2681
34. Serrano E, Bajo J (2019) Deep neural network architectures for social services diagnosis in smart cities. Futur Gener Comput Syst [Internet] 100:122–31. Available from: https://doi.org/10.1016/j.future.2019.05.034
35. Shiau WL, Dwivedi YK, Lai HH (2018) Examining the core knowledge on facebook. Int J Inf Manag [Internet]. 43(May):52–63. Available from: https://doi.org/10.1016/j.ijinfomgt.2018.06.006
36. Socher R, Perelygin A, Wu J (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing [Internet]. (October):1631-42. Available from: http://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf%5Cn, http://www.aclweb.org/anthology/D13-1170%5Cn, http://aclweb.org/supplementals/D/D13/D13-1170
37. Socher R, Huval B, Manning CD, Ng AY (2012) Semantic Compositionality through Recursive Matrix-Vector Spaces. Proc 2012 Jt Conf Empir methods Nat Lang Process Comput Nat Lang Learn (July):1201–11
38. Song H, Kwon B, Yoo H, Lee S (2020) Partial gated feedback recurrent neural network for data compression type classification. IEEE Access 8:151426–151436
39. Usama M, Xiao W, Ahmad B, Wan J, Hassan MM, Alelaiwi A (2019) Deep learning based weighted feature fusion approach for sentiment analysis. IEEE Access 7:140252–140260
40. Xing Y, Xiao CA (2019) GRU model for aspect level sentiment analysis. J Phys Conf Ser 1302:032042
41. Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access 7(c):51522–51532
42. Yang CHH, Qi J, Chen SYC, Chen PY, Siniscalchi SM, Ma X, Lee CH (2021) Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 6523–6527. IEEE
43. Yang M, Zhao W, Chen L, Qu Q, Zhao Z, Shen Y (2019) Investigating the transferring capability of capsule networks for text classification. Neural Netw [Internet] 2019;118:247–61. Available from: https://doi.org/10.1016/j.neunet.2019.06.014
44. Zhang D, Tian L, Hong M, Han F, Ren Y, Chen Y (2018) Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification. IEEE Access 6:73750–73759
45. Zulqarnain M, Ghazali R, Ghouse MG, Mushtaq MF (2019) Efficient processing of GRU based on word embedding for text classification. Int J Inf Vis 3(4):377–383
46. Zulqarnain M, Ghazali R, Ghouse MG, Hassim YMM, Javid I (2020) Predicting financial prices of stock market using recurrent convolutional neural networks. Int J Intell Syst Appl 12(6):21–32
47. Zulqarnain M, Ishak SA, Ghazali R, Nawi NM (2020) An improved deep learning approach based on variant two-state gated recurrent unit and word embeddings for sentiment classification. Int J Adv Comput Sci Appl 11(1):594–603
48. Zulqarnain M, Ghazali R, Hassim YMM, Aamir M (2021) An enhanced gated recurrent unit with auto-encoder for solving text classification problems. Arab J Sci Eng 46:8953–8967
49. Zulqarnain M, Alsaedi AKZ, Ghazali R, Ghouse MG, Sharif W, Husaini NA (2021) A comparative analysis on question classification task based on deep learning approaches. PeerJ Comput Sci 7:e570

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