An Improved Deep Learning Approach based on Variant Two-State Gated Recurrent Unit and Word Embeddings for Sentiment Classification

Muhammad Zulqarnain[1], Suhaimi Abd Ishak[2], Rozaida Ghazali[3], Nazri Mohd Nawi[4]
Muhammad Aamir[5],Yana Mazwin Mohmad Hassim[6]
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia[1, 2, 3, 5, 6]
Soft Computing and Data Mining Center (SMC), Universiti Tun Hussein Onn Malaysia[4]
86400, Parit Raja, Batu Pahat, Johor, Malaysia[1, 2, 3, 4, 5, 6]

***Abstract*—Sentiment classification is an important but challenging task in natural language processing (NLP) and has been widely used for determining the sentiment polarity from user opinions. And word embedding technique learned from a various contexts to produce same vector representations for words with same contexts and also has been extensively used for NLP tasks. Recurrent neural networks (RNNs) are common deep learning architecture that are extensively used mechanism to address the classification issue of variable-length sentences. In this paper, we analyze to investigate variant-Gated Recurrent Unit (GRU) that includes encoder method to preprocess data and improve the impact of word embedding for sentiment classification. The real contributions of this paper contain the proposal of a novel Two-State GRU, and encoder method to develop an efficient architecture namely (E-TGRU) for sentiment classification. The empirical results demonstrated that GRU model can efficiently acquire the words employment in contexts of user's opinions provided large training data. We evaluated the performance with traditional recurrent models, GRU, LSTM and Bi-LSTM two benchmark datasets, IMDB and Amazon Products Reviews respectively. Results present that: 1) proposed approach (E-TGRU) obtained higher accuracy than three state-of-the-art recurrent approaches; 2) Word2Vec is more effective in handling as word vector in sentiment classification; 3) implementing the network, an imitation strategy shows that our proposed approach is strong for text classification.**

Keywords*—RNN; GRU; LSTM; encoder; Two-state GRU; long-term dependencies; sentence classification

I. INTRODUCTION

Automated sentiment classification is the process of extracting the opinions of various user expressed in texts, that are contains emotions or opinions behind the sentences, and identifies the positive or negative aspects of comments, which is very useful in analyzing user generated contexts. The issue of sentiment classification is a popular area in natural language processing (NLP) tasks and recently has gained a plenty of concentration. There are lots of data publically are available on various online platforms, that allow us to performs a sentiment classification for the favors of academic attention. Sentiment analysis is done at both at phrase level and document or paragraph level. Both the levels offer unique challenges and hence require different techniques to tackle them. Word embedding is a fundamental task in NLP and commonly used method to encode words into a high dimensional space. In recent years, they have taken excellent attention to the base of extract both semantic and syntactic information. The main concept of word embeddings is a long history but it has become well-known since Bingio et al. efforts [1] in which each word is shown by word vector and the concatenations of many previously word vector are used to predicts the following based on a language model. Traditionally, to represented an each word as a vector with high-dimensional sparse vector similar to the number of unique terms in the vocabulary using distributional approaches [2] such as vector context-based techniques [3] [4] and the Hyperspace Analog to Language (HAL) model [5]. Recent, a latest distributed words representations training technique, identified as word embeddings [6] [7] [8] has been established to illustrate words as low-dimensional vector for text representation of real numbers, which can effectively capturing semantic and syntactic word similarities from big datasets. These word embedding approaches has been excellently applied for several tasks included entity recognition [9], dependency parsing [10], text classification [11], and speech recognition [12].

Word embedding, the logical meaning of words that trained from specific contexts tends to produce same vector representation for word with same contexts. This technique performs better for semantic-oriented applications but its problem for sentiment classification because words with same vectors representation due to same contexts may have an contrary sentiment separation, as in the examples of happy-sad referred in [13] and positive-negative in [14]. Furthermore, most of the authors have investigated the unique features of the vector representation of words through machine learning [15]. The two well-known word embeddings representation methods of neural network language models, is called Glove word vector and TF-IDF were introduced by Pennington and Aizawa [16], [17]. However, these traditional context-based word embedding such as TF-IDF and GloVe usually unsatisfied to capturing appropriate sentiment information, which may result in words with same vectors representation having a reverse sentiment polarity. To handle this issue, in this research have suggested using Word2Vec, word embedding method to representation words in text as vectors for sentiment classification.

In recent years, deep learning architectures have gradually presented better performance in several data mining applications, such as text classification, entity recognition and sentiment analysis. These architectures effectively addressing the features representation issue because they learnt features from contextual data automatically. Between deep feed-forward neural networks, convolutional neural networks (CNNs) [18] have been presented to capture from words or phrases, and recurrent neural networks (RNNs) [19] are capable to capture temporal dependencies in sequence information and have shown strong semantic composition approaches for sentiment classification [18]. Provided large texts in social media, there is absence of features. To achieve progressively important features, we further used appropriated concept of distributed illustration of words where each input is denoted by numerous features and each feature is included in several potentially sequential inputs. In particularly, we used the pre-trained Word2Vec word embedding method [6] for distributed representation of social media.

RNNs have been extremely used in recent years for the tasks of texts classification. The key benefit of RNNs is that they can be applied to extracts temporal sequential data with variable-length, which flexibilities generates in evaluating reviews of various lengths. Recent, simplified architecture through LSTM, known as Gated Recurrent Unit (GRU), was proposed by [20]. GRUs contains fewer parameters and explained by very simple set of equations, thus need significantly less computational power. Relation among GRU and LSTM effectiveness is an initiate problem and a domain of research.

In this research, we have investigated the effectiveness of GRU for sentiment classification with distributed represen-tation in social media. Applied variant-GRU model for the aim of preventing the issue of gradient exploding or vanishing in an existing RNNs and also overcome the deficiency of standard GRU. In this work, we conducted the experiment on two benchmark datasets, IMDB and Amazon Products Reviews. We compared the performance of proposed Encoder Two-State Gated Recurrent unit (E-TGRU) with three traditional RNNs models namely are, Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM) and Bidirectional Long Short Term Memory (Bi-LSTM).

Our research work continues to further investigate standard GRU based on variant-GRU with encoder to automatic preprocessing data to provide an improved presentation of the inputs than original raw inputs. Based on previous work, our main objective is enhancing the standard GRU structure in order to increase accuracy of sentence classifications and minimize the information loss. In particularly, based on the above studied, the main contribution of this research is summarized as follows:

- Proposing a variant of GRU is included encoded gated recurrent unit (E-GRU) for sentiment analysis. This method performs automatically preprocess text data through encoder.
- Proposed a Two-State Gated Recurrent Unit (TGRU) for sentiment classification.
- To proposed Word2Vec pre-trained word embedding method is applied for sentiment classification to illustrate words as vectors in long-short term.
- Experimental results demonstrate that the (our network) namely Encoder Two-State GRU (E-TGRU) architecture performs well on both tasks to takes advantages of the encoded local features extracted and captured the long-term dependencies, and further enhance the sentiment classification performance but significantly less computational expensive than Bi-LSTM.

II. Models Descriptions

A. Recurrent Neural Network

A standard RNN is kind of artificial neural networks in which connections among the units form a bidirectional cycle, and they perform the similar task for each element in the sequence. The RNN technique is better for sequential issue especially capturing temporal information in the loop. The RNN uses recurrent hidden state whose activation that particularly dependent on the previous timestep while performing the sequential information and the current state rely on the current input. Thus, the current hidden state makes complete usage of past information. In this way, standard RNN can handle variable length and computes sequential data in dynamic processes. The architecture of an RNN is shown in Fig. 1. When provided a sequential inputs X = [x_1, x_2 ... x_t ... x_T] of length T, process output vector O_t, an RNN defines the hidden state U_o at the time t with the sequential of following equations:

$$\overrightarrow{O_t} = \varphi(\overrightarrow{W_x}x_t + \overrightarrow{U_o}\overrightarrow{h_{t-1}}) \quad (1)$$

$$\overleftarrow{O_t} = \varphi(\overleftarrow{W_x}x_t + \overleftarrow{U_o}\overleftarrow{h_{t+1}}) \quad (2)$$

$$O_t = [\overrightarrow{O_t} : \overleftarrow{O_t}] \quad (3)$$

where $W_x \in R^{O_h \times O_x}$ is the weights matrix connecting input layer to hidden layer, and $U_o \in R^{O_h \times O_h}$ is the hidden layers weights matrix. σ is the sigmoid activation function and W_x, U_o, are parameters of the traditional RNN.

In Fig. 1, the input unit is X_1 of time-step t, which shows the word vector of the t-*th* word in the text; h_t is the final activation state of step t; O_t shows the output at time t, the output is chosen according to the require of the network; U, W are the weights parameters of the networks are require to trained the model. However, RNNs are hard to train and suffer from vanishing and exploding gradients issue.

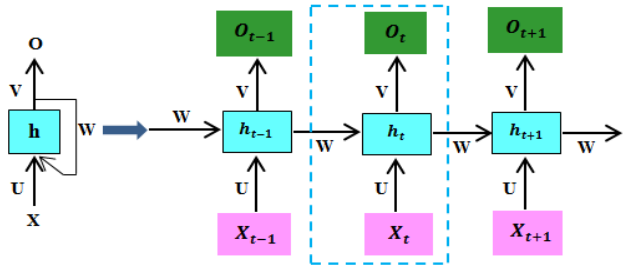

Fig. 1. Traditional Structure of RNN.

Although the RNN is very powerful when dealing with sequential problems, it is difficult to train with the gradient descent method and suffer from vanishing and exploding gradient (explosion) issue [20]. On the other hand, variants of RNN have been developed to solve the above issues, such as LSTM and GRU. Between them, GRU avoids overfitting, as well as saves training time. Therefore, GRU is adopted in our method.

B. Long Short Term Memory

Long Short-term Memory (LSTM) is one of the RNN structure that was initially proposed by [21] and has largely applied in natural language processing. LSTM consist the gated mechanism and internal cell memory that help to addresses the well-known issues relevant to vanishing gradients or exploding. The fundamental concept of "gates" used in LSTM for the aim of handling the sequence contextual data. Furthermore, a common LSTM architecture units is composed of an internal memory cell, and three gates in recurrent connection that help the model to determine how much information pass away and extracting more information with timestep. In addition, by using three gates are generally developed to handle to insert or ignore the information in the memory cell. These gates help the model to regulate how to update the current memory cell and the current hidden stateU_h.

Furthermore, this addition process is completed through three steps. First, Sigm have used as a sigmoid activation functions to process the data which require to store to the internal memory cell. Second, the tanh function is taken to achieve a vector over h_{t-1} and x_t. The final step, output gate is determined the task for choosing appropriate information from the memory cell to output.

The LSTM cells that are used in the transition functions are implemented as follows:

$$i_t = Sigm(W_{xi}x_t + U_{hi}h_{t-1} + b_i) \quad (4)$$

$$o_t = Sigm(W_{xo}x_t + U_{ho}h_{t-1} + b_o) \quad (5)$$

$$f_t = Sigm(W_{xf}x_t + U_{hf}h_{t-1} + b_f) \quad (6)$$

$$\hat{a}_t = tanh(W_{x\hat{a}}x_t + U_{h\hat{a}}h_{t-1} + b_{\hat{a}}) \quad (7)$$

$$c_t = f_t * x_{t-1} + (i_t * \hat{a}_t) \quad (8)$$

$$h_t = O_t * tanh\,(c_t) \quad (9)$$

where, Sigm is the logistic sigmoid function that produce the output between [0, 1]. The variables and bias to be calculated during the learning procedure are W_i, W_o, W_f, $W_c \in R^{m\times p}$, U_i, U_o, U_f, $U_c \in R^{m\times m}$ b_i, b_o, b_f, $b_c \in R^{m\times 1}$ and $*$ indicates the element-wise multiplication of the two vectors. Each gate is composed out of a sigmoid layer and has the capability to remove or add information from the memory cell. At the current time t, U_h denotes hidden states, i_t, O_t, f_t indicates as a input, output and forget gate. W_i, W_o, and W_f represent the weight parameters of LSTM respectively, while b_i, b_o, b_f refers to the biases of the gates.

C. Traditional Gated Recurrent unit

Gated Recurrent Unit (GRU) is another advance kind of RNN, and simplified variation of LSTM relatively development proposed by Cho et al. in [20]. GRU is simple variation of LSTM, and that contain only two gates update gate z_t and reset gate r_t that handle the flow of information inside the unit, while without having an individual memory cells are shown in Fig. 2. GRU also illustrates the powerful capability of modeling to capturing long-term dependencies between the elements of a sequence. GRU calculates two gates, which manages the flow of information through each hidden unit. Each hidden state U_h at time t, given input x_t calculated using the following equations:

$$z_t = \sigma_g\,(W_x^z x_t) + (U_h^z h_{t-1}) \quad (10)$$

$$r_t = \sigma_g\,(W_x^r x_t) + (U_h^r h_{t-1}) \quad (11)$$

$$\tilde{h}_t = \tanh(W_x^{\tilde{h}} x_t) + U_h^{\tilde{h}}(r_t * h_{t-1}) \quad (12)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (13)$$

where z_t refers to the update gate, and r_t refers reset gate, W, U and are weights matrix and vectors. σ_g is a sigmoid activation function and tanh is the hyperbolic tangent. After developing the design of the GRU model, the learning technique for the GRU model requires to be determined. At currently, for RNNs such as GRU, become a common training method include back propagation trough time (BPTT) and real time recurrent learning (RTRL).

However, based on the recent studied there are two issues in standard GRU network. First, in data pre-processing phase much manual experience is needed to preprocess data for accurately classification and second one is high consumption of memory. Therefore, we proposed GRU variant, such as encoder E-GRU. Fig. 3 illustrates the structure of an E-GRU. The weights W, U and the activation function in the E-GRU are the similar as in the standard GRU. Furthermore, the variant E-GRU applies the encoder for automatically pre-process large amount of data. The encoder usually provides a best illustration of the input compared to original raw input, and the encoder is consistently compresses the input data in which select significant features for training.

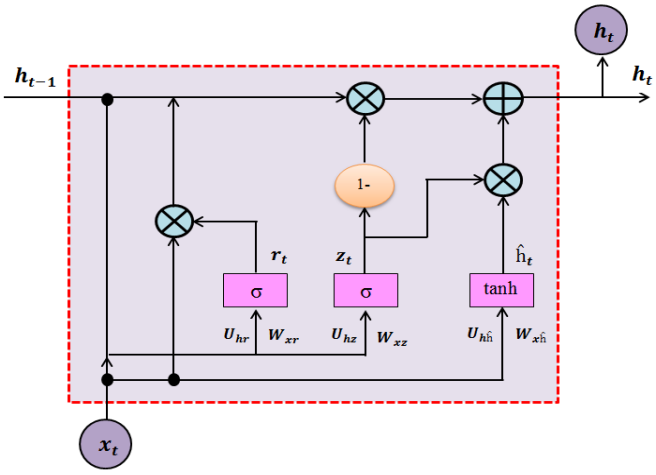

Fig. 2. GRU Architecture.

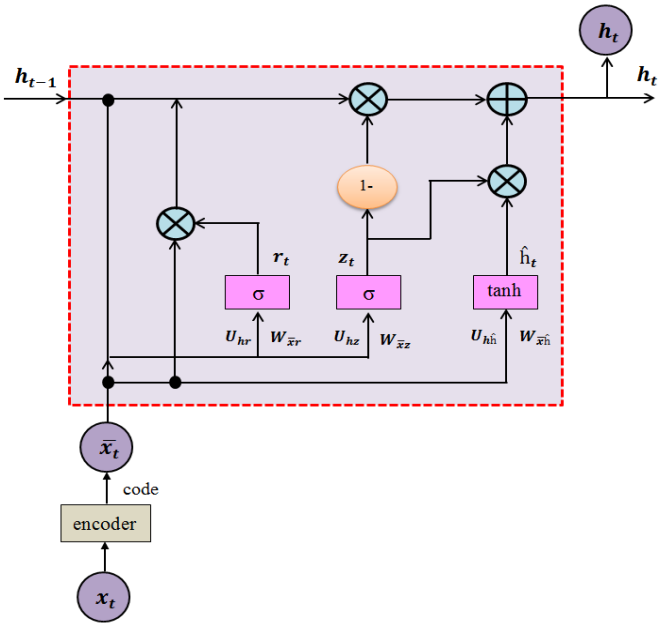

Fig. 3. Architecture of Variant E-GRU.

III. Proposed Methodology

In this section, we demonstrate the particular description of the proposed methodology architecture that contains encoder based gated recurrent unit with word embedding. The proposed methodology apply Two-State GRU which learns to extract forward and backward context features through time steps, whose outputs are then given by encoded GRU model, and finally, followed by a softmax classifier. The description of each methodology elements to solving overall sentiment classification problem are follows:

A. Gated Recurrent Unit (GRU)

It consist a gating structure and a advance type of standard RNNs. It also illustrates the powerful capability to process of sequential data and capturing long-term dependencies between the elements by preserve previous state in the internal state of model through time step t.

B. Variant Gated Recurrent Unit

The first variant consists of binary gated recurrent unit (Bin-GRU), and local feature-based GRU (LF-GRU), and so on. Zhao et al. [29] introduced LF-GRU. In LF-GRU, first it extracts local features from segment or windows of time-series data. After that, LF-GRU base model is applied to learn with average weighed features from sequential of local features. However, the above approaches are requiring more manual experience for preprocessing massive data. Therefore, we proposed encoder GRU (E-GRU). E-GRU uses the encoder for automatically preprocess data efficiently. In this way, the output of encoder (E-GRU) becomes the input of Two-State GRU appropriately. In this paper, we mostly discussed Two-State GRU and E-GRU.

A. Traditional Auto-Encoder

Auto-encoder is a kind of unsupervised artificial neural networks that learns to effectively compress and encode data. The main purpose of auto-encoder is identifying further helpful and valuable features from huge amount of dataset without application of any dimensionality reduction method. Basically, Auto encoder usually performs in two stages namely encoding and decoding. Along with encoding stage converts the input features to a new representation [22] while decoding stage tries to convert this new representation back as near as possibly to its original inputs.

B. Encoder GRU (E-GRU)

The It contain on encoder GRU that used to reduce dimensionality from input data by applying the encoder part of the auto-encoder, it gives the best representation of the inputs than original raw inputs, and then the outputs of the encoder as become the inputs of the Two-State GRU.

C. Features Extraction

In sentiment classification feature extraction technique obtain significant role for identifying relevant features from raw data. It also includes to eliminate unnecessary features [31] and maintaining important features that consider to improve the accuracy of the model.

D. Automatical Preprocessing

It presents to the automatically capturing of features from the original raw data and removes manual interference.

E. Word Embedding Layer

In sentence classification process the initial stage is pre-processing the inputs sentence and sentiment context words. To superior representation the limited contents in long and short text. In this paper, we applied the pre-trained Word2Vec [23] word embedding method in embedding layer to extract the contextual correlation between words in training data. Word2vec performs as a predictive model to train their co-occurrence vectors to extracts the correlation among the target word and the context words in a simple way: it tries to extract the relevant semantic regularities by learning with the activation of the target word and its context words. The word embedding layer of the model changes words context into real-valued features vectors that captured semantic and syntactic data. Let $L \in R^{V*d}$ be the embedding query table produced by Word2vec, where d is the dimension of words and V is the vocabulary size. Assume that the input sentence contains of n words and the sentiment resource contains of m words.

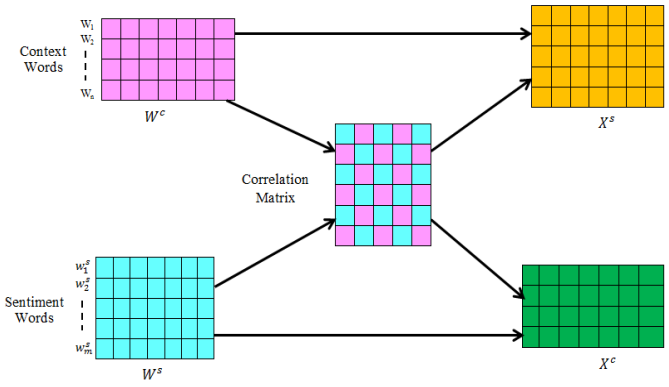

Fig. 4. Sentiment-Context Word Correlation.

The input sentence retrieves the word vectors from L and obtain 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. Similarly, the sentiment resource sequence can retrieve the word vectors from and form a list of vectors $[W_1^s, W_2^s, \dots, W_m^s]$. 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. Fig. 4 show the process is simply concatenation of all words embedding in V.

F. The TGRU Network Architecture

The GRU recurrent layer has the ability to represent sequences e.g. sentences and very useful to capturing long-term dependencies between elements of a sequence. GRU can be applied for sentiment classification in the same manner as it has been used in Cho K [24]. GRU is a recent simplified variant of LSTM, that combine the forget gate and input gate into a single new update gate, which enhance the convergence time and iteration times of model training. First, an embedding layer generated a suitable size. The embedding layer will perform to show each word by a real valued vector of similar range to the fixed dimension. These values are the weights among the embedding layer and the hidden layer on top of it. These units are not only connected to the layer below and above them but also connected to units within their own layer. At the end of the hidden layer we attain the representation of the entire sequence which can be used as input to linear model or classifier.

GRU structure are basically consists an update gate and reset gate. Reset gate (r_t) determines that how much previous memory can be ignore from previous hidden state h_{t-1}.The update gate (z_t) is determines how much previous state keep around and send among the existing state and new calculated state with parameter bias b_z. x_t as a p input vector dimension at time t , σ is the logistic sigmoid activation, and W_{zx} (q × p matrix), U_{zh} (q × q matrix), b_z (q×1 vector) are determined size parameters which are common through an whole model.

$$z_t = \sigma(W_{zx}x_t) + U_{zh}h_{t-1} + b_z) \quad (14)$$

where σ is the sigmoid activation applied for binary classification in the dense layer (output layer), and the value range of each element in the update gate z_t are [0, 1].

The reset gate r_t is calculated similarly to the update gate but with changed weights value: W_{rx}(q × p matrix), U_{rh} (q × q matrix), b_r (q × 1 vector).

$$r_t = \sigma(W_{rx}x_t) + U_{rh}h_{t-1} + b_r) \quad (15)$$

GRU reveals to the entire state of each iteration. In the same way, the candidate state $\hat{h}_t$ is similarly computed to the existing recurrent unit.

$$\hat{h}_t = tanh(W_{\hat{h}}x_t) + r_t * U_{\hat{h}}h_{t-1} + b_{\hat{h}}) \quad (16)$$

The candidate state $\hat{h}_t$ at the current timestep *t*, the reset gate r_t is handle the flow of the previous hidden activation h_{t-1} containing past information. If the reset gate is around zero, the previous hidden calculated state h_{t-1} will be removed.

Output state:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \quad (17)$$

The hidden state h_t uses the update gate z_t to update the previous hidden state h_{t-1} and the candidate hidden state $\hat{h}_t$. If the update gate is close to 1, the previous hidden state will be held and passed to the current instant. * is the Hadamard product between of the previous state $h_{t-1}$with $(1 - z_t)$ and element-wise multiplication * of the update gate z_t with candidate activation state $\hat{h}_t$.

GRU can preserve memory substantially longer than existing RNN due to gating mechanism. However, based on the recent studies and practically observation, we find out that when GRU examines a word it only considers the forward linguistic context, so it is very needed for GRU to learn the contexts by backward pass. We also observe that the meaning of word in any language model is affected not only the forward pass but also on the backward pass.

Therefore, we proposed Two-State GRU to handle the above problem; the proposed TGRU network contain two directions, one for positive time direction namely (forward state), and other for negative time direction namely (backward state) as presented in Fig. 5. TGRU learns the contexts of a word from both directions. TGRU is inspired by the bidirectional recurrent neural networks (BRNNs) in [25]. In the training process, it splits each training sequential process into both individual recurrent networks forward and backward directions, and finally these directions are jointly combine into the output layer. The equations for update gate z_t, reset gate r_t, candidate state $\hat{h}_t$, and final output activation state h_t of the forward and backward GRU are presented as a follows:

Forward Pass:

$$\overrightarrow{z_t} = \sigma(\overrightarrow{W_{zx}}x_t) + \overrightarrow{U_{zh}}\overrightarrow{h_{t-1}} + \overrightarrow{b_z}) \quad (18)$$

$$\overrightarrow{r_t} = \sigma(\overrightarrow{W_{rx}}x_t) + \overrightarrow{U_{rh}}\overrightarrow{h_{t-1}} + \overrightarrow{b_r}) \quad (19)$$

$$\overrightarrow{\hat{h}_t} = \tanh(\overrightarrow{W_{\hat{h}}}x_t) + \overrightarrow{r_t} * \overrightarrow{U_{\hat{h}}}\overrightarrow{h_{t-1}} + \overrightarrow{b_{\hat{h}}}) \quad (20)$$

$$\overrightarrow{h_t} = (1 - \overrightarrow{z_t}) * \overrightarrow{h_{t-1}} + \overrightarrow{z_t * \hat{h}_t}) \quad (21)$$

In addition, we added backward pass to our proposed approach to discover more useful information.

Backward Pass:

$$\overleftarrow{z_t} = \sigma(\overleftarrow{W_{zx}}x_t) + \overleftarrow{U_{zh}}\overleftarrow{h_{t-1}} + \overleftarrow{b_z}) \quad (22)$$

$$\overleftarrow{r_t} = \sigma(\overleftarrow{W_{rx}}x_t) + \overleftarrow{U_{rh}}\overleftarrow{h_{t-1}} + \overleftarrow{b_r}) \quad (23)$$

$$\overleftarrow{\hat{h}_t} = \tanh(\overleftarrow{W_{\hat{h}}}x_t) + \overleftarrow{r_t} * \overleftarrow{U_{\hat{h}}}\overleftarrow{h_{t-1}} + \overleftarrow{b_{\hat{h}}}) \quad (24)$$

$$\overleftarrow{h_t} = (1 - \overleftarrow{z_t}) * \overleftarrow{h_{t-1}} + \overleftarrow{z_t * \hat{h}_t}) \quad (25)$$

To the initiation of a word at time *t*: $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ = for a random series $(x_1, x_2, \dots, x_n)$ consisting *n* words, each word shown as a dimensional vector at time *t*.

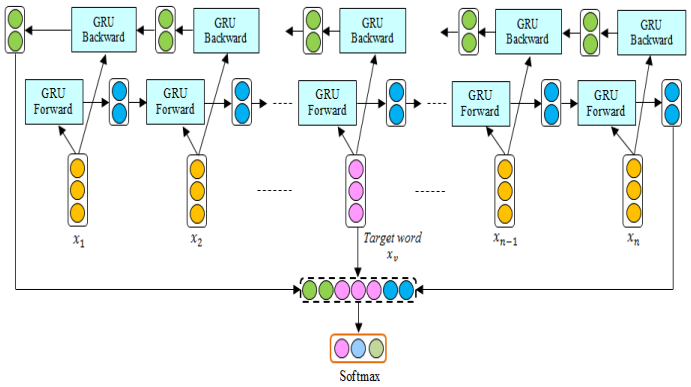

Fig. 5. The Proposed Two-State GRU Architecture for Sentiment Classification.

The forward pass of GRU calculates $\overrightarrow{h_t}$ that shows the left to the right contexts of the sentence while the backward pass of GRU phase take the right to left contexts $\overleftarrow{h_t}$ respectively. Then contexts representation of forward and backward directions were conbined into a single layer. Fig. 5 presents the detailarchitecture of TGRU.

Generally, the complexity of an model is calculated by *O(D)* and the model estimated parameters is computed by *D*. Two general pieces of information applied to calculate *D* is the dimension of the input vector p-dimension and hidden layer dimension q-dimension. Table I illustrates the detail of computed parameters of GRU, LSTM, Bi-LSTM and TGRU.

The complexity of TGRU double compare to standard GRU because the number of parameters are double. The performance of TGRU requires more time and resources for execution than GRU or LSTM while require less time and resoreces for execution than Bi-LSTM. However, our proposed E-TGRU approach is capable to explored useful information which greatly improve the accuracy of the sentiment classification.

G. Flowchart

In this sub-section we illustrate the flowchart of model for semtiment classification algorithm. Fig. 6 illustrates a flowchart of sentiment classification that contain three major phases. The 1st phase contain on data formatting for sentiment classification purpose. The 2nd phase consist of preprocessing data of model using variant GRU. In this phase, we apply the encoder to preprocess the text data. After that, the preprocessed data are utilized as input to the TGRU. The 3rd phase is cross verification.

TABLE. I. NUMBERS OF PARAMETERS

Model	**Number of parameters**
GRU	$3 \times (q^2 + qp + q)$
LSTM	$4 \times (q^2 + qp + q)$
Propose E-TGRU	$\mathbf{6 \times (q^2 + qp + q)}$
Bi-LSTM	$8 \times (q^2 + qp + q)$

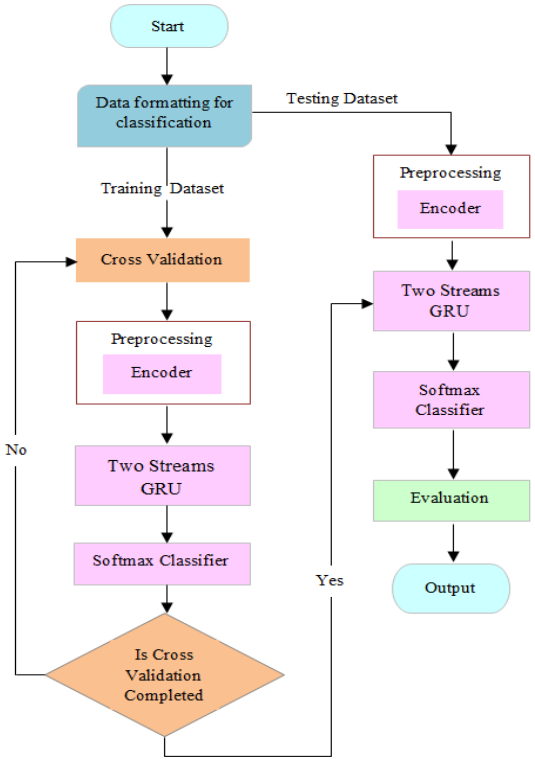

Fig. 6. Flowchart of Network Sentiment Analysis.

Algorithm 1: Encoder-two State Gated Recurrent Unit for for Sentiment Analysis

Input: vector V, that include U model features in the network, each network word as a vector $(v_{i1}, v_{i2}, \dots, v_{ir})$.
Output: Evaluates result in the dataset.

1. **Step 1: Create encoder layer**
2. Insert the 1st encoder layer of E_1 unit included *tanh* activation
3. Insert the 2nd encoder layer of E_2 unit included *tanh* activation.
4. **Step 2: Build Two-State GRU model**
5. Insert the 1st GRU layer of L_1 units with Sigmoid activation and dropout is $d_1$and recurrent dropout is rd_1.
6. Insert the 2nd GRU layer of L_2 units with Sigmoid activation and dropout is d_2 and recurrent dropout is rd_2.
7. **Forward Pass:**
8. Initialize from the input layer do a forward pass over the network and take left to right contexts $\overrightarrow{h_t}$ of the sentence.
9. **Backward pass:**
10. Initialize from output layer to do backward over the network and take the right to left contexts $\overleftarrow{h_t}$ of the sentence.
11. **Step 3: Train and validate model**
12. **while** initial stop condition is not met **do**
13. **while** training dataset is not empty **do**
14. Prepared a mini-batch dataset as network inputs.
15. Calculate categorical cross entropy loss function.
16. Update weights and bias using RMSprop optimizer algorithm.
17. **end while**
18. Validates network with validation set.
19. **end while**
20. **Step 4: Test model**
21. Test fine-tuned hyper-parameters with test dataset.
22. **return** Evaluates result in test dataset.

It is observing the data preprocessing is very important for increasing the accuracy of the network, because the valueable features are attained by data preprocessing directly affect the final performance of the model. Therefore, we introduced variant GRU include encoder method to preprocessing data.

IV. Experimental Results and Analysis

In this section, we conducted the different experiments to present how E-TGRU performs better as compared with three well-known state-of-the-art recurrent models on two benchmark sentiment classification datasets the Stanford Large Movie Review dataset (IMDB) and Amazon Product Reviews (APR) dataset.

A. IMDB

The results for this paper were achieved using the IMDB dataset originally collected by Andrew Maas [26]. It consists of the labeled dataset of 50,000 IMDB movie reviews, especially selected for sentiment classification. These reviews are divided 50:50 ratio into training and test data. We preprocessing the dataset following the implementation similar in [27]. Furthermore, it has 50,000 un-labelled movie reviews which we useful for unsupervised training.

B. Amazon Product Reviews(APR)

We used the dataset containing Health and Personal Care product reviews from Amazon website that was available in University of California [28]. We trained the data set that contains 10,000 reviews included 50% positive and negative reviews are including to preparing it a binary classification. Additionally, these reviews are divided 15% of the dataset is applied for testing and 15% for validation aim.

C. Implementation Detail

The Data preprocessing and manipulate have performed in 3.6 Python version and anaconda. The network was trained through 30 epochs. All the simulation works were performed on Intel Core i7-3770XPU on a Windows PC with @3.40 GHz, and 4GB RAM machine. We called the name of proposed approach E-TGRU to pre-trained word vector for sentiment classification.

In this sub section, we describe each layer of E-TGRU model in detail. In order to improve the performance of the proposed model, that first step is improve the quality of the dataset, we enhance the quality of text dataset by preprocessing technique, and then gain 300-dimensional word vector by using pre-trained word2vec method selected from Google for sentiment classification. In our experiment, we used RMSprop optimizer to set their default optimal parameter setting with learning rate is 0.001 and decay factor is 0.9. The model is trained by mini-batch to performed gradient descent with batch size of 64. The sigmoid activation has applied as dense layer for binary sentiment classification and softmax activation function for multiclass sentiment classification. To avoid the overfitting problem, we have applied dropout strategy for TGRU layer with 128 memory units for each forward and backward direction. We set dropout rate of 0.2 uses by embedding layer, while the recurrent structure has a dropout of 0.4. After combining the forward and backward GRU, one more dropout layer was added to reduce 50% overfitting issue. Moreover, 10-fold cross validation has applied to minimize the arbitrary impact of the model.

Fig. 7 is presenting the detail accuracy and loss function of the model. As illustrated in the figure, after 10 epochs the training and validation accuracy greatly improve over 86% and the training and validation loss reduce below 35%. And after 25 iterations the model finally achieved accuracy over 88% and loss decrease below 30%.

D. Sentiment Analysis Results

Fig. 8 summarized the classification results on IMDB, and APR datasets. We evaluated the efficiently of our proposed E-TGRU model and compared it with three state-of-the-art exiting RNNs approaches GRU, LSTM and Bi-LSTM. The results prove that our proposed model is suitable for sentence level sentiment classification with higher accuracy of 89.37%. In our research, to train the model by using Word2vec word vector method for computing a real value vector representation of a word and performs excellent on the both IMDB and APR datasets. We fixed both the word embedding dimensions and number of units to 64.

Additionally, the outperformance of E-TGRU through encoder method and two-state GRU mechanism to demonstrate that our model is much better than other traditional models for the task of sentiment classification. Although, it can obtains the higher performance in both binary and multiclass sentiment classification tasks. In this paper, we evaluated the classification performance of the proposed models based on three evaluation metrics, namely the accuracy, F1-scores and mean square error (MSE). First, we conduct the experiment on IMDB movie reviews dataset, the performance evaluates between three baseline models are presented in Fig. 8.

In Fig. 8 shown that the proposed model GRU-Embed have achieved better performance on IMDB dataset with accuracy of 89.37%. We can see the continuously excellent performance of E-TGRU than GRU, LSTM and Bi-GRU. Next, we evaluate the classification performance of Amazon Products Reviews (APR) dataset in Fig. 9. The proposed model E-TGRU also achieved best performance on APR dataset with accuracy of 87.58%, while GRU achieved 83.08% LSTM attained 83.63% and Bi-LSTM obtained much better accuracy of 84.96% than GRU and LSTM. On the other hand, we compared F1-score performance of two traditional recurrent models is presented in Fig. 10. It is clearly showing that better performance is achieved over the application of pre-trained word embeddings. The F1-scores results show that the pre-trained word vectors are better general features extractor and can be used entire datasets.

E. Comparing Tgru With Recent Studies

This research have evaluated the networks performance compared with existing recent researches [18], [29] and [30]. Kim et al. [18] proposed static and non-static convolutional neural network trained on top of pre-trained word vector using Word2Vec for sentence level classification. They applied multiple channels have obtained 81.58% accuracy using same dataset. Socher et al. [29] introduced recursive matrix-RNN network that learns compositional vector representations for

phrases and assigns the word vector and matrix method to every node in a parse tree to achieved 79.00% . Zulqarnain et al. [30] propose gated recurrent unit based on batch normalization for sentence classification. They applied batch normalization technique in forward layer and used Glove word vector in embedding layer.

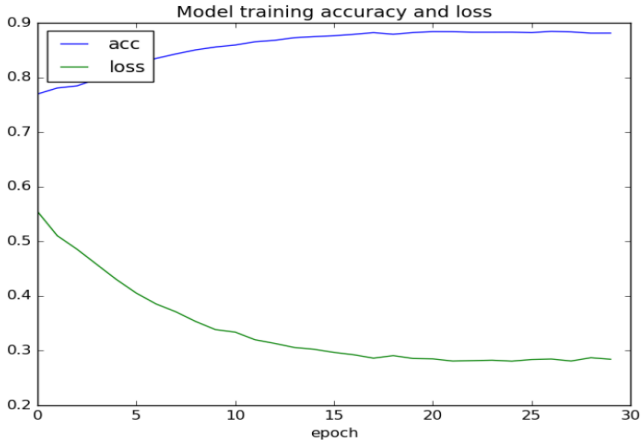

Fig. 7. Effect of Accuracy and Loss at Various Time Periods.

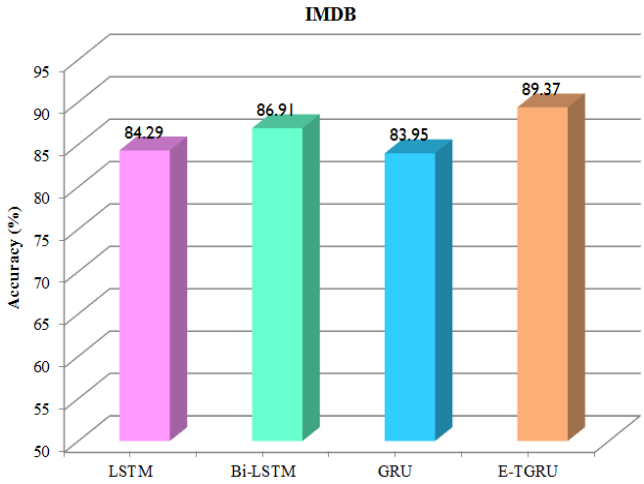

Fig. 8. Accuracy Results Comparison of Proposed E-TGRU Model with Three Baseline RNNs Models.

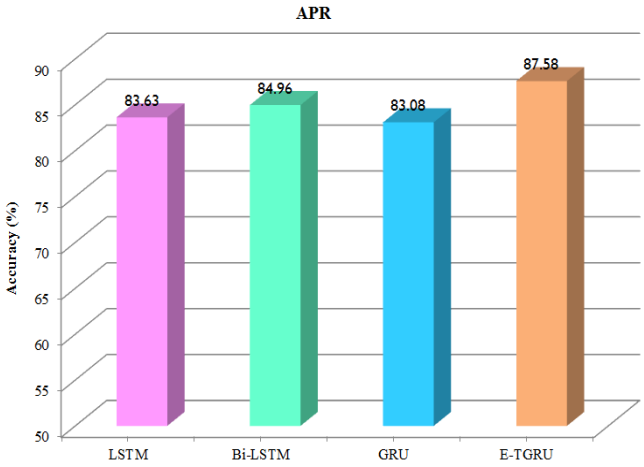

Fig. 9. Classification Performance Comparison for APR Dataset.

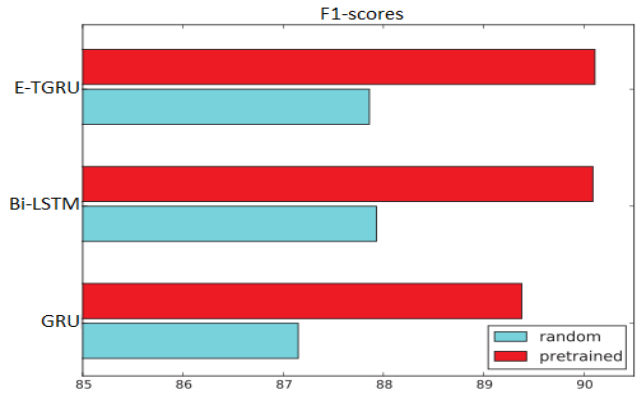

Fig. 10. E-TGRU Model Compared with Bi-LSTM and GRU in the Term of F1-Score (%) Performance by Impact of Initialize Word Embedding.

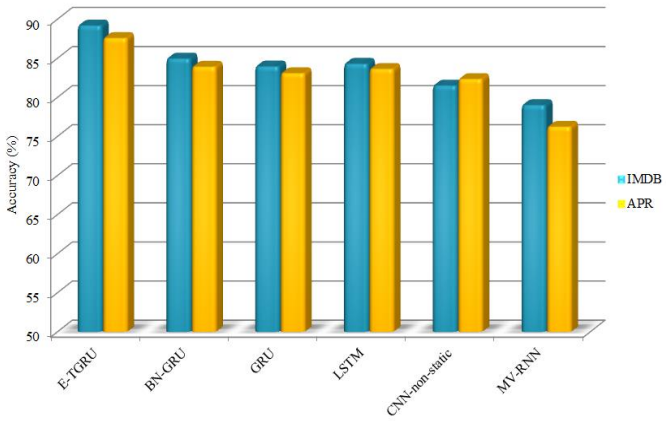

Fig. 11. Our Proposed Model Compared with Three Existing Studies in the Term of Accuracy.

LSTM, GRU and E-TGRU were executed on the same datasets and same period with three approaches. Bi-LSTM (bidirectional LSTM), which is extended variation of LSTM that has been extensively applied in recent studied [31], is also included for further comparisons. We deployed LSTM and GRU on same dataset with similar parameters to those in [32]. Fig. 11 showed the result that our proposed E-TGRU model outperforms than other existing models. This improve development is justifiable because not only we build the encoder and Word2Vec embedding method in GRU, but we also combine the forward and backward contexts to learn more useful information.

F. Comparison Error Rate with Traditional RNNs

In this section, we perform to analysis an error rate of our proposed E-TGRU model with three state-to-the-art RNNs approaches such as standard GRU, LSTM and recent Bi-LSTM. Execution setup showed that with the continuous increase of epochs, the mean square error is continuously decreasing and the final MSE is 0.0162 on IMDB dataset and 0.2713 on APR. We fixed both the word embedding dimensions and number of units to 64 and execute the model for 30 epochs. We found that proposed model converged faster than GRU, LSTM and Bi-LSTM to achieved very lower error rate even after many epochs. To make these models comparable, we implement these models with the identical

structural design. Finally, we evaluate our E-TGRU model with state-of-the-art existing RNNs models on IMDB and APR datasets. Table II demonstrates the results that proposed E-TGRU model achieves much better performance in the term of the error rate than GRU, LSTM and Bi-LSTM.

TABLE. II. COMPARISON ERROR RETE (%) WITH EXISTING RNNS MODELS

Models	**IMDB**	**APR**
LSTM	*0.1278*	*0.3569*
GRU	*0.1025*	*0.3490*
Bi-LSTM	*0.0842*	*0.3125*
E-TGRU	***0.0162***	***0.2713***

V. CONCLUSION

Sentiment classification remains popular and significant area of natural language processing. In this paper, we investigated variant gated recurrent unit included encoder GRU (E-GRU) to preprocess the texts data for sentiment classification. E-GRU frequently provides an excellent representation of the input than the original raw input. Furthermore, we also developed Two-State Gated Recurrent Unit (TGRU) which is included forward and backward states, that is capable to learn more valuable information, especially for text processing issue. Then, we used Word2Vec pre-trained word embeddings method that is possible to learn the contextual semantics of words from the text can be effectively classified. Based on experimental observation, we found that RNNs models, being a recurrent network it can effectively capturing the useful information from a massive array of sequential data and the best choice in terms of accuracy. We conduct the experiment on two benchmark sentiment analysis datasets, included IMDB and APR respectively. The proposed E-TGRU model achieved highest 89.37% accuracy on IMDB dataset and 87.58% accuracy on APR dataset. Our proposed model achieves much better performance in the term of an error rate than GRU, LSTM and Bi-LSTM, when increases the number of epochs.

In future work, there are many ways to extend this work. Future research can be dedicated the proposed approach using multiple sentiment lexicons and much powerful ranking approaches to enhance the sentiment classification performance and also reduce the computational complexity of the proposed model.

ACKNOWLEDGMENT

The authors would like to thanks Ministry of Education Malaysia and Universiti Tun Hussein Onn Malaysia (UTHM) for funding this research activity under the Fundamental Research Grant Scheme (FRGS), Vot Number: FRGS/1/2019/ICT05/UTHM/03/1.

REFERENCES

[1] P. Vincent, “A Neural Probabilistic Language Model,” A neural probabilistic Lang. Model. J. Mach. Learn. Res., vol. 3, pp. 1137–1155, 2003.

[2] M. Baroni, “Composition in distributional semantics,” Lang. Linguist. Compass, vol. 07, no. June, pp. 511–522, 2014.

[3] P. D. Turney and P. Pantel, “From Frequency to Meaning : Vector Space Models of Semantics,” J. Artif. Intell. Res., vol. 37, pp. 141–188, 2010.

[4] M. Aamir, F. Wahid, H. Mahdin, and N. M. Nawi, “An efficient normalized restricted Boltzmann machine for solving multiclass classification problems,” Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 8, pp. 416–426, 2019.

[5] C. Burgess, C. Burgess, K. Livesay, and K. Lund, “Explorations in Context Space : Words , Sentences , Discourse,” Discourse Process., vol. 25, no. June, pp. 211–257, 2015.

[6] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient Estimation ofWord Representations in Vector Space,” arXiv Prepr. arXiv1301.3781, pp. 1–12, 2013.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “5021-Distributed-Representations-of-Words-and-Phrases-and-Their-Compositionality,” Adv. Neural Inf. Process. Syst., vol. April, pp. 3111–3119, 2013.

[8] M. Zulqarnain, R. Ghazali, M. G. Ghouse, and M. F. Mushtaq, “Efficient Processing of GRU Based on Word Embedding for Text Classification,” Int. J. Informatics Vis., vol. 3, no. 4, pp. 377–383, 2019.

[9] L. Ratinov and J. Turian, “Word representations : A simple and general method for semi-supervised learning,” Proc. 48th Annu. Meet. Assoc. Comput. Linguist., no. July, pp. 384–394, 2010.

[10] K. Gimpel, “Tailoring Continuous Word Representations for Dependency Parsing,” Proc. 52nd Annu. Meet. Assoc. Comput. Linguist., vol. 2, pp. 809–815, 2014.

[11] W. Sharif, N. A. Samsudin, M. M. Deris, and M. Aamir, “Improved relative discriminative criterion feature ranking technique for text classification,” Int. J. Artif. Intell., vol. 15, no. 2, pp. 61–78, 2017.

[12] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light Gated Recurrent Units for Speech Recognition,” IEEE Trans. Emerg. Top. Comput. Intell., vol. 2, no. 2, pp. 92–102, 2018.

[13] S. M. Mohammad, B. J. Dorr, G. Hirst, and P. D. Turney, “Computing Lexical Contrast,” Comput. Linguist., vol. 39, no. January 2010, pp. 555–590, 2013.

[14] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, “Sentiment Embeddings with Applications to Sentiment Analysis,” IEEE Trans. Knowl. Data Eng., vol. 28, no. October, pp. 496–509, 2016.

[15] S. Clark, “Vector Space Models of Lexical Meaning,” Handb. Contemp. Semant. Ed., no. September, pp. 1–42, 2012.

[16] J. Pennington, R. Socher, and C. D. Manning, “GloVe : Global Vectors for Word Representation,” Proc. Conf. Empir. Methods Nat. Lang. Process., no. October, pp. 1532–1543, 2014.

[17] A. Aizawa, “An information-theoretic perspective of tf – idf measures q,” Inf. Process. Manag., vol. 39, pp. 45–65, 2003.

[18] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Artif. Intell. Rev., no. 4, pp. 655–665, 2014.

[19] P. Liu, X. Qiu, and X. Huang, “Recurrent Neural Network for Text Classification with Multi-Task Learning,” Proc. 25th Int. Jt. Conf. Artif. Intell. IJCAI-16, p. to appear, 2016.

[20] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” arXiv, no. September, pp. 1–15, 2014.

[21] S. Hochreiter, “Long Short Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1–32, 1997.

[22] Y. Wang, H. Yao, and S. Zhao, “Auto-encoder based dimensionality reduction,” Neurocomputing, vol. 184, no. November, pp. 232–242, 2016.

[23] A. H. Ombabi, O. Lazzez, W. Ouarda, and A. M. Alimi, “Deep Learning Framework based on Word2Vec and CNN for Users Interests Classification,” 2017.

[24] K. Cho, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,” arXiv, vol. 5, pp. 1–9, 2014.

[25] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.

[26] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis,” Proc. 49th Annu.

Meet. Assoc. Comput. Linguist. Hum. Lang. Technol., vol. 1, pp. 142–150, 2011.

[27] P. Lamblin et al., "Theano : new features and speed improvements," arXiv Prepr. arXiv1211.5590, pp. 1–10, 2012.

[28] J. Mcauley, R. Pandey, and J. Leskovec, "Inferring Networks of Substitutable and Complementary Products," Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. data Min., pp. 785–794, 2015.

[29] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic Compositionality through Recursive Matrix-Vector Spaces," Proc. 2012 Jt. Conf. Empir. methods Nat. Lang. Process. Comput. Nat. Lang. Learn., no. July, pp. 1201–1211, 2012.

[30] M. Zulqarnain, R. Ghazali, S. H. Khaleefah, and A. Rehan, "An Improved the Performance of GRU Model based on Batch Normalization for Sentence Classification," Int. J. Comput. Sci. Netw. Secur., vol. 19, no. 9, pp. 176–186, 2019.

[31] P. Zhou et al., "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification," Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap., pp. 207–212, 2016.

[32] H. Peng, E. Cambria, and A. Hussain, "A Review of Sentiment Analysis Research in Chinese Language," Cognit. Comput., vol. 9, no. 4, pp. 423–435, 2017.