EAI Endorsed Transactions
on Internet of Things

Research Article EAI.EU

A Hybrid Deep Learning GRU based Approach for Text Classification using Word Embedding

Poluru Eswaraiah[1] and Hussain Syed[2,*]

[1, 2] VIT-AP University, School of Computer Science and Engineering, Andhra Pradesh, India

Abstract

Text categorization has become an increasingly important issue for businesses that handle massive volumes of data generated online, and it has found substantial use in the field of NLP. The capacity to group texts into separate categories is crucial for users to effectively retain and utilize important information. Our goal is to improve upon existing recurrent neural network (RNN) techniques for text classification by creating a deep learning strategy through our study. Raising the quality of the classifications made is the main difficulty in text classification, nevertheless, as the overall efficacy of text classification is often hampered by the data semantics' inadequate context sensitivity. Our study presents a unified approach to examine the effects of word embedding and the GRU on text classification to address this difficulty. In this study, we use the TREC standard dataset. RCNN has four convolution layers, four LSTM levels, and two GRU layers. RNN, on the other hand, has four GRU layers and four LSTM levels. One kind of recurrent neural network (RNN) that is well-known for its comprehension of sequential data is the gated recurrent unit (GRU). We found in our tests that words with comparable meanings are typically found near each other in embedding spaces. The trials' findings demonstrate that our hybrid GRU model is capable of efficiently picking up word usage patterns from the provided training set. Remember that the depth and breadth of the training data greatly influence the model's effectiveness. Our suggested method performs remarkably well when compared to other well-known recurrent algorithms such as RNN, MV-RNN, and LSTM on a single benchmark dataset. In comparison to the hybrid GRU's F-measure 0.952, the proposed model's F-measure is 0.982%. We compared the performance of the proposed method to that of the three most popular recurrent neural network designs at the moment RNNs, MV-RNNs, and LSTMs, and found that the new method achieved better results on two benchmark datasets, both in terms of accuracy and error rate.

Keywords: Text Classification, Natural Language Processing, RNN, GRU, LSTM

Received on 24 September 2023, accepted on 05 December 2023, published on 13 December 2023

doi: 10.4108/eetiot.4590

1. Introduction

Rapid progress in the social and technical information domain, coupled with the exponential growth of digital text formats, heralds the arrival of the age of massive text data [1]. There are promising future uses for text classification in information retrieval, digital libraries, and other areas [2]. Therefore, the ability to effectively organize and leverage these massive textual datasets is crucial in this setting. Automatic text sorting has become one of the most important issues for large companies that need to handle huge amounts of data. When it comes to managing massive amounts of data found on the internet, automatic TC is an essential tool. In recent years, DL algorithms have attracted a lot of interest [3] because of their ability to learn layered, hierarchical representations of high-dimensional data. Text categorization has not received as much attention as pattern recognition [4], sentiment analysis [5], and CV [6]. As a result of catastrophe dimensions, scarce data, and other difficulties, traditional text representation has become a limiting factor in the efficiency of many NLP operations.

Many novel sentiment analysis methods have emerged with the rise of deep learning technologies. Word and phrase meanings can be learned with the help of the abundant unlabelled textual data that is readily available. Word2vec [1]

[*] Corresponding author. Email: hussain.syed@vitap.ac.in

tries to do this by acquiring word embeddings from unlabelled text samples. It picks up new information by doing word predictions based on context (CBOW) and word predictions based on a specific word (SKIP-GRAM). These word embeddings are used in current methods like Tf-idf and others to create dictionaries and to minimise dimensionality. Additional methods, such as the recurrent neural tensor network (RNTN) [7], are discovered for capturing representations at the phrase level. In addition to its success in image classification, CNNs have been demonstrated to be useful in text categorization [8].

The fundamental issue is that sentences in natural English tend to be of varying lengths. Fixing the size of the context window helps, however this does not help with extracting semantics that go beyond the window's scope. Although recurrent neural networks can handle text sequences of varying lengths, they are notoriously difficult to train. This led to the use of RNN variants such as LSTM and GRU. In numerous NLP tasks, including sentiment analysis, translation, and sequence generation, LSTM is generating headlines since its proposal in 1997 by Hochreiter et al. [9]. The concept of GRUs was first presented in 2014 by K. Cho [10]. GRU are likely more practical because of their simpler form. In this paper, we make an effort to demonstrate its benefits over LSTM when used to sentiment analysis. Tf-idf, Word2Vec, k-means terms, and the Ensemble model. Are only some of the methods that have been applied by other researchers? It was discovered that GRU performed better than any of the other single models, and that this finding was further enhanced by employing an ensemble model. Figure 1 represents the Basic architecture of query-based text classification which is shown below.

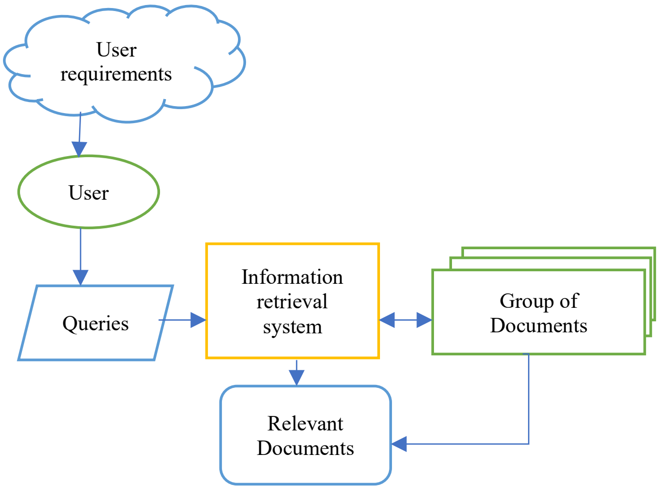

Figure 1. Basic architecture of query-based text classification

The proposed works are characterized by their significant contributions, which are outlined in detail below:

In this research, we will test a hybrid GRU network's ability to classify texts by using a pre-trained word embedding strategy (Glove). Using a GRU network allowed us to circumvent the gradient exploding or vanishing issue that plagues regular RNNs. Experimental configurations on two benchmark datasets (TREC and RNN) are used to evaluate the hybrid GRU model's performance to that of the RNN, the "Matrix-Vector Recurrent Neural Network," and the LSTM. For the purpose of text classification, GRUs proved beneficial due to their ability to remember long-term demands and capably grasp the meanings among words. When dealing with sequence data, GRU approaches really shine. The experimental findings for both datasets demonstrate that the hybrid GRU model (our method) provides superior accuracy and error rate.

- ➢ The hybrid model is superior to the individual components since it combines the advantages of GRU, RNN, and LSTM to classify text data. Many uses, from sentiment analysis to content suggestion, stand to profit from the accuracy boost.

- ➢ The method incorporates word embedding techniques, which improves the model's capacity to interpret and represent words in the context of written texts. The capture of semantic data is aided by this addition.

- ➢ The approach's strength comes from the fact that it may be adapted to noisy or unstructured text data. This is of paramount importance in contexts where user-generated material, social media, or other dirty data sources are being used.

For the purpose of text categorization, we employ the TREC dataset in conjunction with our hybrid model (GRU-LSTM-RNN). We compared the performance of the proposed method to that of the three most popular recurrent neural network designs now (RNNs, MV-RNNs, and LSTMs), and found that the new method achieved better results on single benchmark dataset, both in terms of accuracy and error rate.

2. Literature survey

Classification is one among the most well-known and widely used NLP applications. Texts mining involve more than just classifying documents; it also includes recognizing themes, filtering out spam, labeling semantic roles, and more. Natural language processing is one area where deep learning architectures have been widely recognized as producing state-of-the-art outcomes. The process of organizing unstructured data involves four steps: pre-processing, term weighting, feature selection, and ultimately, obtaining text vectors. Many deep learning models have been applied to a variety of natural language processing (NLP) applications, such as chunking, language modelling, semantically linked word recognition, web news classification, and chunking. The likelihood that w will follow t in a sequence is something a natural language model can predict [11]. For instance, "deep recursive neural networks model have been applied for parsing and sentiment analysis [12]," "question answering [13]," in addition to "logical inference [14]." Recurrent neural networks can be used to model language [15], recognize voice [16], and create phrases from images [17].

Bengio et al. [17] introduced a novel strategy based on a neural network-based language model (NNLM) to learn embeddings of words from each word's prior contexts. The C&W model generates word embeddings using a convolutional network that considers both local and global contexts. Using straightforward single-layer architecture, the CBOW and skip-gram algorithms developed by Mikolov et al. [18] enable rapid analysis of word embeddings from large datasets. While the methods presented in [20] for producing word embeddings relied on dependencies between words and presented global vectors (GloVe), the methods described in [19] relied solely on linear contexts and local contexts (typically just a few words in the preceding and subsequent contexts). To solve the issue of local contexts, the GloVe method considers global word-to-word co-occurrence statistics, whereas semantics-based word embeddings employ a dependency parser to produce syntactic con grammes.

Recent advances in RNN language models have shown the value of distributed representations and the capacity to explain arbitrarily lengthy dependencies [20]. After training on a character-level dataset, the RNN variation described in [21] is able to produce phrases with a natural sounding tone. More recently, [22] shown how an RNN-LM may be utilized to generate image descriptions by conditioning the network model with a pre-trained convolutional image feature presentation. Having the ability to train deep networks also paves the way for a more sophisticated strategy for capitalizing on connections between labels and features, leading to a more accurate prediction.

This feature has opened the door for RNNs to be employed for sequential tasks such as text categorization and named entity recognition. More complicated network topologies are made possible by the tree-LSTM introduced in [23], a variant of RNN that allows each LSTM unit to incorporate data from many child units.

Ghosh et al. (2023) embarked on a comprehensive study to assess water quality through predictive machine learning. Their research underscored the potential of machine learning models in effectively assessing and classifying water quality. The dataset used for this purpose included parameters like pH, dissolved oxygen, BOD, and TDS. Among the various models they employed, the Random Forest model emerged as the most accurate, achieving a commendable accuracy rate of 78.96%. In contrast, the SVM model lagged behind, registering the lowest accuracy of 68.29%[31].

Alenezi et al. (2021) developed a novel Convolutional Neural Network (CNN) integrated with a block-greedy algorithm to enhance underwater image dehazing. The method addresses color channel attenuation and optimizes local and global pixel values. By employing a unique Markov random field, the approach refines image edges. Performance evaluations, using metrics like UCIQE and UIQM, demonstrated the superiority of this method over existing techniques, resulting in sharper, clearer, and more colorful underwater images [32]. Sharma et al. (2020) presented a comprehensive study on the impact of COVID-19 on global financial indicators, emphasizing its swift and significant disruption. The research highlighted the massive economic downturn, with global markets losing over US $6 trillion in a week in February 2020. Their multivariate analysis provided insights into the influence of containment policies on various financial metrics. The study underscores the profound effects of the pandemic on economic activities and the potential of using advanced algorithms for detection and analysis [33].

2.1 Steps for Classifying Text

The process of classifying text usually has four main steps: But first, there are steps that need to be taken to gather and prepare the information that will be used for the work.

2.1.1. Data Preparation

Preprocessing methods play a crucial part in enhancing the effectiveness of models. Improving the text dataset for different text mining applications begins with transforming unstructured text into structured text. We look at tokenization, stop word removal, and stemming, three essential parts of data preprocessing.

i. Tokenization

Tokenization is the process of separating individual words, phrases, symbols, or other recognized units from a continuous stream of text. Separating words inside a sentence is the primary objective of tokenization.

ii. Stop removing words

Common but unimportant words are eliminated at this stage. Both before and after processing natural language data (text), these terms are typically removed. Commonly used shortened forms of these words have been removed: "the," "an," "is," "at," "of," "but" and so on.

iii. Stemming

The practice of stemming unifies various word forms into a single representation called stems. In the text, for instance, the words "presentation," "presented," and "presenting" can all be shortened to the universal representation "present." The usage of stemming is prevalent in text processing, especially in the field of information retrieval (IR), based on the idea that a query containing the term "presenting" signals a demand for publications containing the words "presentation" and "presented."

2.1.2. Representation of Documents

The data must first be written in a way that is comprehensible to the categorization algorithm. One of the most common strategies is called the "Bag of Words" (BOW). It illustrates text by displaying the frequency with which individual words appear in each text.

2.1.3. Reduced Dimensions

It is impossible to analyse all the words in a text corpus as potential features for a classification system because of the large number of words involved. It can be computationally challenging to analyse such large datasets. This highlights the need of carefully picking out attributes that are truly representative to use as inputs in the subsequent classification stage.

2.1.4. Model Development

This is a crucial part of the text-sorting procedure. A subset of the text data in the dataset is chosen to serve as the training set. The classification model is then developed by training the chosen model on this dataset.

3. Proposed Method

Bidirectional loops between the units are a feature of RNNs, a form of artificial neural network. This structure is appropriate for pulling valuable linguistic data from lengthy word sequences in a collection because it is made to analyses consecutive events, such as word sequences. As shown in Figure 2, prior to passing those through the U, W, and V weight vectors in the RNN architecture, the present input and the previous hidden state are added at time-step t.

$$h_t = f_h(x_t.U + h_t.W + b_{(h)}) \quad (1)$$

$$O_t = f_o(h_t.V) \quad (2)$$

The output, O_t, is determined by Equation 2, while the value of the hidden state, ht, is determined by Equation 1. The weight matrix W is utilized to compute the persistent link between hidden states, while matrices U and V represent the connection from the hidden layer to the input and output, respectively. However, conventional RNNs have a poor reputation for ease of training because to frequent occurrences of bursting and disappearing gradients. This is because the slopes may become too steep. If the network is too small, it will not be able to learn, and if it is too big, the weights will grow too long, and the network will stop learning. With the addition of gated mechanisms, RNNs have spawned two variants—LSTM networks and GRUs—that are better able to deal with gradient vanishing.

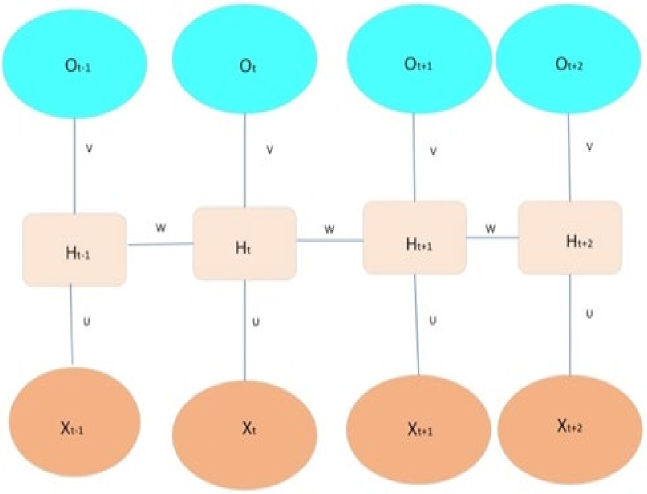

Figure 2. The traditional RNNs architecture.

Input, ignore, and output gates all play critical roles in the LSTM design, determining how information flows through the network. These gates together determine how much weight to give to the current step's data addition, preservation of the previous state, and effect over the surrounding network. The relationships between the inputs, forget, and output gates of an LSTM network are formulated mathematically.

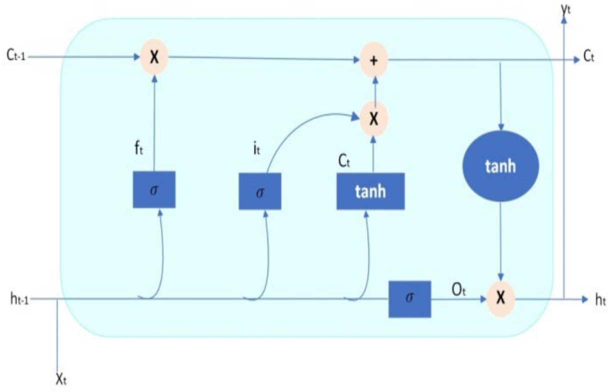

Figure. 3. LSTM Units

$$i^t = \sigma_g(w^{(i)}X_t + u^{(i)}h_{t-1} + b^{(i)}) \quad (3)$$

$$f^t = \sigma_g(w^f X_t + u^{(f)}h_{t-1} + b^{(f)}) \quad (4)$$

$$O^t = \sigma_g(w^{(o)}X_t + u^{(o)}h_{t-1} + b^{(o)}) \quad (5)$$

$$\hat{C}^t = tanh(w^{(\hat{c})}X_t + u^{(\hat{c})}h_{t-1} + b^{(\hat{c})}) \quad (6)$$

$$C^{(t)} = f^{(t)} * C^{t-1} + i^{(t)} * \hat{C}^{(t)}) \quad (7)$$

$$h^t = O^t * \tanh(C^{(t)}) \quad (8)$$

The input gate, ignore gate, and output gate control the flow of data through a recurrent neural network. Each cell has an input gate and an output gate that determines how much new information is added to the cell's state and how much of the current state is broadcast to the outside network. The amount of the previous state to be kept is likewise determined by the ignore gate. Both the gates and the suggested state acquire biases and preferences as they are trained. The proposed state is computed using Eq. (6), and Eq. (7) is used to modify the actual state ct. The ht hidden state is then computed using the (8) equation. Compounding elements individually is represented by the letter 'o'.

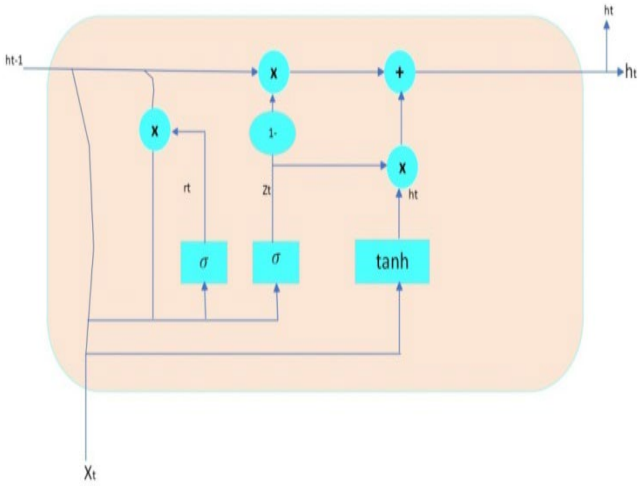

Figure. 4. GRU Units

Instead of a dedicated memory module, GRU relies on a gating mechanism to regulate data transfer. To control the flow of information from the previous activation, GRU computes two gates, the update gate and the reset gate, while calculating a new possible state using a reset gate.

To ensure that an update gate properly mixes the proportion of prior activation and new candidate activation, the following formulas are employed to determine each hidden state at time-step t:

Update gate:

$$z_t = \sigma_g(W^{(z)}X_t + U^{(z)}h_{t-1} + b^{(z)}) \tag{9}$$

Reset gate:

$$r_t = \sigma_g(W^{(r)}X_t + U^{(r)}h_{t-1} + b^{(r)}) \tag{10}$$

Candidate gate:

$$\widehat{h_t} = tanh(W^{(\widehat{h})}X_t + U^{(\widehat{h})}(r_t * h_{t-1})) \tag{11}$$

Final output:

$$h_t = z_t * h_{t-1} + \widehat{h_{t-1}} * (1 - z_t) \tag{12}$$

It is demonstrated by previous work that each neural network building has drawbacks that can be overcome by mixing several buildings. In this work, we investigate how adding layers from different designs to the current RNN model affects the model's performance. Comparing the LSTM, which is the memory layer (Figure. 3), to the GRU, we find that the former is better able to remember long-distance relations. (Figure. 4)

While the sigmoid activation function is used in both the LSTM and GRU units' storage layers, the hyperbolic activation function in the output layer facilitates data retrieval even after long-term storage. GRUs, on the other hand, perform better than LSTMs and is simpler to train because they need less information. That is why GRUs is always embedded against LSTM, even if retrieval is not necessary.

The proposed model, shown in Figure 5, employs an RNN composed of three LSTM layers and two GRU layers. Important long-term dependency management is enabled by the output gate layer. Considering the current state of the cell, the output vector from the previous cell, and the input vector, the forget gate layer assigns a value. The forgetting layer is responsible for deciding whether to forward the value to the input layer, which further proof of this is. In the LSTM's forget gate layer, sigmoid neural function is multiplied by a point-wise operator to generate values. The input gate layer has a dual purpose. The incoming vector input is used to update a value using a sigmoid activation function, and this value is then compared to another value obtained using a hyperbolic activation function. The outcomes of the comparison are then included into the cell's existing state. The Output Gate Layer compares the input vector generated by the sigmoid to the cell state update generated by the hyperbolic.

In comparison, an LSTM or a fully gated GRU needs only two gates (the forget gate layer and the input gate). GRU has been around for more than half a decade, but it is still favored in some situations because of the shorter training time and smaller dataset needs. Because of its unique update gate and forget gate, LSTM is obviously more complicated. As a result, the LSTM's complexity allows for the incorporation of GRU, with the result being improved model control thanks to the inclusion of GRU units. In light of these findings, we compare the capabilities of four models featuring MV-RNN, RNN, LSTM and GRU units, respectively. The TREC dataset is used to train the models because it can accurately capture syntactic and semantic representations of the words. A significant corpus is needed to train the model in GloVE, which increases the memory required in the long run because of the model's inability to capture out-of-vocabulary terms. When compared to Word2Vec, GloVE's operation is similar; however, the training process will not be focused on the weights associated with frequently used word pairs. The benefits of using the GloVE dataset to train the models under consideration in the study are supported by the points made above. Previous work has shown that the limitations of individual neural network topologies can be mitigated by employing a hybrid approach. In this study, we look into how incorporating new layer types from various designs into the existing RNN model can improve its performance. Contrasting the memory layer, LSTM when comparing the GRU [28]. In terms of their ability to remember long-distance relationships, we find that the former is superior.

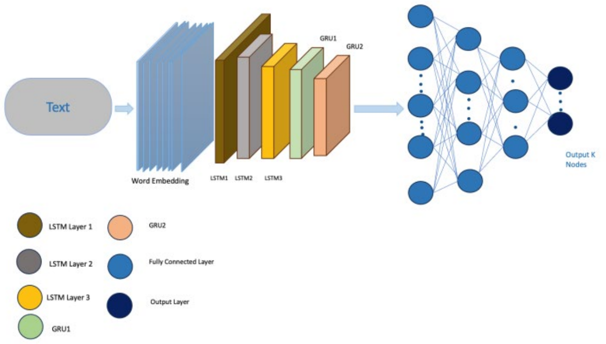

Figure. 5. The Proposed Hybrid GRU Layers

4. Result and Discussion

4.1. TREC Dataset Description

The efficacy of the word embedding based text classification for hybrid GRU model was evaluated experimentally in this research. Through a rigorous training and testing procedure, we assessed the model's efficacy on commonly used benchmark dataset TREC dataset. In order to establish the suggested model's supremacy, we also contrasted it with other traditional RNN-based methods like MV-RNN, GRU LSTM, and Recurrent Neural Network.

We performed tests on the reference dataset TREC [24] to evaluate the performance of the suggested model. TREC includes six distinct query categories, including "LOC," "NUM," and "ENTY," as shown in Table 1. 500 questions make up the test dataset, while 5,452 labelled questions make up the training dataset.

Table 1. TREC Dataset Statistics Summary

Label	Training	Validation
ENTY	1256	97
DESC	1156	135
ABBR	88	12
NUM	889	111
HUM	1223	60
LOC	840	85
Total	5452	500

To improve the performance of the suggested model, we first employed flowing to combine all the comparable example words into a single word, and we also deleted stop words from the input order (such as "and" "are," "of," "the," and "to"). Both the word embedding size, d, and the GRU size, u, was held constant throughout the experiment. Next, GloVe word vectors that had been pre-trained by Penington et al. [24] were used to initialize all word embeddings from the text data. Several researchers have modified word vector training approaches [25] to improve performance when categorizing phrases based on their emotional tone.

To depict the model's generalization abilities better accurately, we try to use the same embedded data across datasets whenever possible. When there was no clear classification source word in the texts, we devised the self-attention mechanism by instead considering all of the surrounding terms to be resource words for categorization. Each GRU unit in the deep learning networks was tweaked to have 200 hidden states, which required modifying all of the network layers. We applied AdaDelta optimization [26] to the proposed model, setting the learning rate to 0.001 and the minibatch size to 64. To prevent GRU layer overfitting, we used L2 regularization (with a coefficient r of 105) and the dropout technique [27]. We have measured the state-of-the-art in the text classification task in terms of error rate and accuracy.

4.2. Performance Evaluation

We used the TREC benchmark dataset and other custom parameters to evaluate our suggested hybrid GRU model. As a benchmark, we compared the suggested hybrid GRU model's performance on both datasets to that of four well-known classical recurrent neural networks: the RNN, the Recursive M-V RNN, the Gate Recurrent Units, and the LSTM [29, 30]. To train the models, we used a stochastic gradient descent approach on mini batches created at random. We also discovered that networks with fewer gated units and fewer embedding dimensions perform better than those with more gated units and more embedding dimensions. For this, we employed a word-level embedding layer trained with the pre-trained GloVe method to assign actual values to each text word in order.

In this experiment, we examined the state-of-the-art RNN, MV-RNN, and LSTM models based on their classification results using the conventional assessment metrics of accuracy and means square errors. Figure 6 displays a comparison of four models' results on the first TREC dataset.

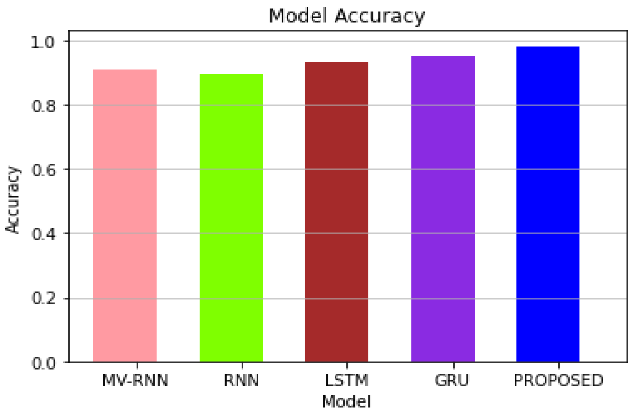

Figure. 6. Comparative analysis of Classification on TREC dataset

The efficacy of all models is similar with minor variations, as shown in Figure 6. The proposed model has the greatest result in terms of accuracy 0.982, while hybrid GRU has a similar F-measure 0.952.

In this section, as shown in figure 7, we will examine the hybrid GRU model that we proposed and compare it to the error rates of the LSTM, MV-RNN, GRU, and RNN models. To conduct this study, we created an execution environment in which to train these models using TREC's benchmark dataset. The experiment showed that the mean square error always went down as the number of epochs went up, and the final Mean Square Error value for TREC was 0.529.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(P_i - P_i^*)^2 \quad (13)$$

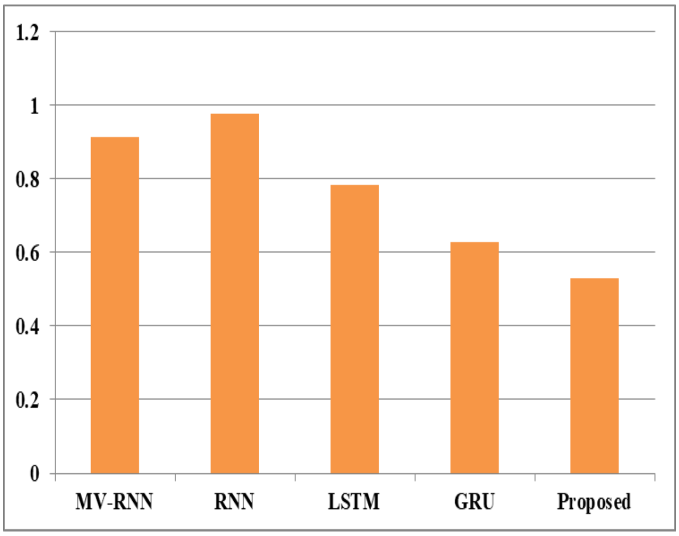

Figure. 7. Error Rate on TREC dataset

In this analysis, we calculated the error rate by dividing the total squared difference between observed and predicted values by the total number of data points (n). Word embedding size and unit amount were both adjusted to 64, as described in the model's implementation settings. Compared to other RNN models, the suggested one converged quickly and had a lower error rate even after many training iterations. All the models had the same basic framework so that the evaluation would be consistent. We evaluated our suggested model against the state-of-the-art RNN models using the TREC datasets. Table 2 shows that the suggested model has a lower error rate than the MV-RNN, LSTM, RNN, and GRU.

Table 2. Comparisons Error Rate (%) with Existing Models

Models	TREC Dataset
MV-RNN	0.914
RNN	0.977
LSTM	0.783
GRU	0.629
Proposed	0.529

4. Conclusion

This study uses the hybrid GRU method to examine the problem of document-level text categorization efficiently. On standard text classification datasets, traditional recurrent neural networks (RNNs) were investigated, and it was discovered that Gated Recurrent Units beat other models in terms of accuracy and error rates. GloVe word vectors and other pre-trained word embedding were used to derive semantic characteristics between words in documents. According to the research, proposed particularly with large quantities of learning data, attention is a suitable model for text classification of sequential data because it can efficiently record long sequence data for natural language comprehension. When compared to conventional RNN, LSTM, GRU, MVRNNs models, the suggested model outperformed them according to empirical findings from the TREC standard dataset to implement text classification and accuracy, error rates. In comparison to the GRU's F-measure 0.952%, the proposed model's F-measure is 0.982%. The paper recommends further investigation into the suggested model's applicability to added NLP techniques.

References

[1] J. Protasiewicz, “A recent overview of the state-of-the-art elements of text classification,” Expert Syst. Appl., vol. 106, pp. 36–54, 2018.

[2] 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.

[3] Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A Review on Deep Learning Techniques Applied to Semantic Segmentation,” pp. 1–23, 2017.

[4] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks,” IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1717–1724, 2014.

[5] 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.

[6] R. Zhao, W. Ouyang, H. Li, and X. Wang, “Saliency detection bymulti-context deep learning,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07–12–June, pp. 1265–1274, 2015.

[7] O. I. and C. Cardie, “Deep Recursive Neural Networks for Compositionality in Language,” Adv. neural Inf. Process. Syst., pp. 2096–2104, 2014.

[8] Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, “Arabic Sentiment Classification Using Convolutional Neural Network and Differential Evolution Algorithm,” Comput. Intell. Neurosci., vol. 2019, pp. 1–16, 2019.

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

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

[11] R. Collobert and J. Weston, “A unified architecture for natural language processing,” Proc. 25th Int. Conf. Mach. Learn. - ICML ’08, pp. 160–167, 2008.

[12] R. Socher, A. Perelygin, and J. Wu, “Recursive deep models for semantic compositionality over a sentiment treebank,” Proc. …, no. October, pp. 1631–1642, 2013.

[13] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, “A Neural Network for Factoid Question Answering over Paragraphs,” Proc. 2014 Conf. Empir. Methods Nat. Lang. Process., pp. 633–644, 2014.

[14] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” 2015.

[15] Kumar et al., “Ask Me Anything: Dynamic Memory Networks for Natural Language Processing,” vol. 48, 2015.

[16] 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.

[17] T. Mikolov, J. Kopecky, L. Burget, O. Glembek, and J. Cernocky, “Neural network based language models for highly inflective languages,” Icassp-2009, pp. 4725–4728, 2009.

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

[19] M. J. Berger, “Large Scale Multi-label Text Classification with Semantic Word Vectors,” Tech. Rep., pp. 1–8, 2014.

[20] 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.

[21] Y. Xiao and K. Cho, “Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers,” arXiv, vol. 1602, no. 00367, 2016.

[22] Karpathy, “Deep Visual-Semantic Alignments for Generating Image Descriptions.”

[23] X. Phan, “Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections,” Proc. 17th Int. Conf. World Wide Web, pp. 91–100, 2008.

[24] R. Johnson and T. Zhang, “Effective Use of Word Order for Text Categorization with Convolutional Neural Networks,” no. 2011, 2014.

[25] D. Roth, “Learning Question Classifiers £,” Proc. 19th Int. Conf. Comput. Linguist., vol. 1, no. August, pp. 1–7, 2002.

[26] H. Lee, “for Modeling Sentences and Documents,” Proc. 15th Annu. Conf. North Am. Chapter Assoc. Comput., no. June, pp. 1512–521, 2015.

[27] D. P. Kingma and J. L. Ba, “A method for stochastic optimization,” arXiv, no. March, pp. 1–15, 2015. Hinton, “Dropout : A Simple Way to Prevent Neural Networks from Overfitting,” J. Mach. Learn. Res. 2014, 15, vol. 15, pp. 1929–1958, 2014.

[28] Sunagar, P., & Kanavalli, A. (2022). A Hybrid RNN based Deep Learning Approach for Text Classification. International Journal of Advanced Computer Science and Applications, 13(6).

[29] Zulqarnain, M., Ghazali, R., Ghouse, M. G., & Mushtaq, M. F. (2019). Efficient processing of GRU based on word embedding for text classification. JOIV: International Journal on Informatics Visualization, 3(4), 377-383.

[30] Arava, K., Chaitanya, R. S. K., Sikindar, S., Praveen, S. P., & Swapna, D. (2022, August). Sentiment Analysis using deep learning for use in recommendation systems of various public media applications. In 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 739-744). IEEE.

[31] Ghosh, H., Tusher, M.A., Rahat, I.S., Khasim, S., Mohanty, S.N. (2023). Water Quality Assessment Through Predictive Machine Learning. In: Intelligent Computing and Networking. IC-ICN 2023. Lecture Notes in Networks and Systems, vol 699. Springer, Singapore. https://doi.org/10.1007/978-981-99-3177-4_6

[32] Alenezi, F.; Armghan, A.; Mohanty, S.N.; Jhaveri, R.H.; Tiwari, P. Block-Greedy and CNN Based Underwater Image Dehazing for Novel Depth Estimation and Optimal Ambient Light. Water 2021, 13, 3470. https://doi.org/10.3390/w13233470

[33] G. P. Rout and S. N. Mohanty, "A Hybrid Approach for Network Intrusion Detection," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 2015, pp. 614-617, doi: 10.1109/CSNT.2015.76.