Design and Implementation for Research Paper Classification Based on CNN and RNN Models

Dipto Biswas[1], *Joon-Min Gil*[2*]

[1] *Department of Computer Software Engineering, Daegu Catholic University, South Korea*
[2] *Department of Computer Engineering, Jeju National University, South Korea*
dipto.biswas94@gmail.com, jmgil@jejunu.ac.kr

Abstract

Deep learning techniques are used as basic essential techniques in natural language processing. They rely on modeling nonlinear relationships within complex data. In this study, “Long Short-Term Memory” (LSTM) and “Gated Recurrent Units” (GRU) deep learning techniques are applied to the classification of research papers. We combine Bidirectional LSTM and GRU with “Convolutional Neural Networks” (CNN) to boost the classification performance for a recommendation system of research papers. In our method, word embedding is also used to classify and recommend research papers. Thus, in this study, we evaluate six types of models, LSTM, GRU, CNN with LSTM, CNN with GRU, CNN with BiLSTM, and CNN with BiGRU. These models used the Word2Vec (CBOW and Sg) pre-trained method to compare their performance on the FGCS dataset. The performance results show that the combined models with CNN architecture achieve better accuracy and F1-Score than the basic LSTM and GRU models. For a more in-depth analysis, the CNN with BiLSTM and CNN with BiGRU models exhibit superior performance compared to the CNN with LSTM and CNN with GRU models. Furthermore, the CBOW Word2Vec embedding method for combined CNN models consistently has better performance than the Sg Word2Vec embedding method.

Keywords: Word embedding, CNN, RNN, Bidirectional LSTM, Bidirectional GRU

1 Introduction

Deep learning is an important tool in text analysis. It has been increasingly used for monitoring of public opinion, service evaluations, satisfaction analysis in the network environment, plagiarism detection, and so on [1-3]. Most text analysis algorithms have been currently operated using statistical learning methods. Because this method has different performances depending on the quality of the feature extraction, it still requires a high level of expertise. This is a time-consuming and tedious task.

Sequence-to-sequence learning has been successfully applied to various areas, such as machine translation and speech recognition. To date, the dominant approach is to encode an input sequence with a series of Bidirectional recurrent neural networks (RNN) and interface them through soft attention mechanisms. It uses a different set of decoders RNN to generate variable-length output. Meanwhile, in machine translation, it was proven that this architecture outperforms existing syntax-based models by a significant margin. Different from RNN, CNN is not dependent on the computation of the previous time steps. This contrasts with RNN, which maintain the full past hidden state, preventing parallel computation within the sequence. CNN is also widely used for image classification as the main part of computer vision systems, such as Facebook's automatic photo tagging and self-driving cars. Recently, CNN has been applied to text classification as well as various classification problems [4-5]. However, CNN encounters a specific challenge when extracting high-dimensional features. Furthermore, it is limited by having only a few convolutional layers, which can pose difficulties in optimizing its performance [6-7]. Nowadays, the LSTM and GRU models [8-9] are used to extract feature from long-term dependencies in data. However, the LSTM and GRU are worked for feed-forward direction. For this reason, the existing methods such as Bidirectional LSTM and Bidirectional GRU [10-11] are used to extract data features in both backward and forward directions.

In this study, we use the Word2Vec pre-trained embedding methods to evaluate the performance of six types of models, LSTM, GRU, CNN with LSTM, CNN with GRU, CNN with BiLSTM, and CNN with BiGRU for the classification and recommendation of research papers.

Thus, in this study, we design and analyze the research paper classification and recommendation systems based on the combination of CNN and RNN models. Each year, a significant number of research papers have been published in various research disciplines. The proper storage and maintenance of these research papers poses considerable challenges. Nevertheless, some online journal sites categorize and maintain research papers according to their respective major and publication year. However, users are frequently not satisfied with search results for their desirous research papers because they are just provided with search results by keyword matching of paper titles for given words without considering the actual meaning of the words in research papers. For this reason, this paper basically utilizes the abstract data of research papers, obtained from Future Generation Computer

*Corresponding Author: Joon-Min Gil; E-mail: jmgil@jejunu.ac.kr
DOI: https://doi.org/10.70003/160792642024072504014

Systems (FGCS) journal in the actual website 'Science Direct'. In our analysis, an abstract of a research paper plays an important role to recommend research papers to users because it is the main summary of the research paper and provides meaningful words that can be accurately classified by comprehending the relationships between words.

Our contributions can be summarized as follows.

1) The utilization of Word2Vec methods contributes to the creation of meaningful word embeddings. Word2Vec techniques capture semantic relationships between words, allowing the model to understand the contextual meaning of terms in research papers.
2) The integration of CNN with Bidirectional LSTM and Bidirectional GRU models enables the model to effectively capture both local patterns and long-term dependencies within the textual data of research papers. This synergistic combination contributes to a more comprehensive understanding of the document structure.
3) The proposed model addresses and surpasses limitations observed in previous research paper classification models. By incorporating CNN, Bidirectional LSTM, and Bidirectional GRU models along with Word2Vec embeddings, the model offers a more sophisticated and accurate solution for understanding and classifying complex academic texts.
4) Experiments are conducted with the FGCS dataset, and the results unequivocally demonstrate the effectiveness of the proposed methods. We compare six types of variant models using two different pretrained models of Word2Vec (CBOW and Sg). In particular, the combination of CNN with BiLSTM and CNN with BiGRU models shows the maximum and best performance with both Word2Vec pre-trained models in this study.

The remainder of this paper is organized as follows. Section 2 explains the related works. Section 3 describes our system model and explains the dataset and data processing methods. This section additionally elucidates the word embedding pre-training techniques, employing both Continuous Bag-of-Words (CBOW) and Skip-gram (Sg) methodologies, for integration with neural network models. Section 4 presents LSTM, GRU, CNN, Bidirectional RNN model. The experimental environments and results of this study are presented in Section 5. Finally, Section 6 concludes the paper with a summary and provides an outlook for future research plans.

2 Related Works

Various types of works can be combined to contribute to this study. We describe the related works as follows:

A BiLSTM model based on Word2Vec techniques has a significant objective [12]. This model aims to efficiently learn a general context embedding function, simplifying the context representation for variable-length sentence contexts around the target word. As a result, it achieved remarkable performance in sentence completion, lexical substitution, and word semantic disambiguation, outperforming other techniques such as the general contextual averaging of word embedding representations proposed by Melamud et al. [12].

Xiao et al. [13] proposed a patent text classification method in the security field. They used a pre-trained Word2Vec model to overcome the high dimensionality suffered by traditional methods. Finally, by training the LSTM and GRU classification models, text functions were extracted from the security field and patent text classification was performed. To improve classification accuracy, some studies [13] have combined CNN, LSTM, and GRU. CNN, LSTM, and GRU models have been used in various natural language processing (NLP) tasks, with surprising and effective results. The study proposed in [14] introduced a text classification model known as CNN-COIF-LSTM. Through experiments involving eight variants, it was demonstrated that the combination of CNN and LSTM without an activation function, or a variant thereof, yields higher accuracy. The hybrid model [15] utilized deep CNN and LSTM to effectively address the challenges associated with sentiment analysis. Additionally, the hybrid model used a dropout technique, regularization technique, and modified linear unit as proposed by Rehman et al. [15], to enhance prediction accuracy. In [16], the authors propose a tree-structured regional CNN-LSTM model to predict VA (valence–arousal) ratings in texts, while in [17] the authors present a multidimensional relation model to predict the dimension scores in deep neural networks. The CNN model with region-based classification uses parts of the text as regions. Combining CNN and LSTM further increases the classification accuracy because it considers both the local information of the sentences and the long-range dependencies between sentences. A new hybrid CNN-LSTM model [16] was proposed, with better results than the previous model proposed by Wang et al. [16]. Salur et al. [17] also proposed a new hybrid model that combines different word embeddings (Word2Vec, FastText, character-level embedding, etc.) [18] with different learning approaches (LSTM, Gated Recurrent Unit (GRU), BiLSTM, CNN). This model uses CNN and LSTM for feature extraction. Various other studies, such as [19], have used this hybrid approach. However, the results have not shown improvement due to the absence of an attention mechanism. On the other hand, more recently, attention models have been introduced and accomplish the state-of-the-art results.

3 System Model

Figure 1 shows an overall flow for the classification and recommendation systems of research papers based on different CNN and RNN models in this study.

As shown in this figure, the abstract dataset for research papers is collected from the Future Generation Computer Systems (FGCS) journal through web crawling. Next, we perform data separation and preprocessing for an abstract dataset. In this phase, the abstract dataset is divided into training set and test set, both of which will be used to create and evaluate the deep learning models. We then apply the preprocessing data to pretraining using word-embedding algorithms. After completing the word embedding process, we apply the Word2Vec pre-trained data to the embedding layer of the CNN and RNN combination models. Finally, we evaluate the proposed models with test data to clarify whether the models are suitable for this study.

3.1 Dataset and Data Processing

We construct dataset from the abstracts of research papers, which are retrieved from journal websites using web scraping tools such as Selenium Python and Beautiful Soup [20]. A total of 5,659 abstracts are retrieved from the FGCS journal. The dataset is separated into 3,961 training data and 1,698 testing data that are randomly chosen.

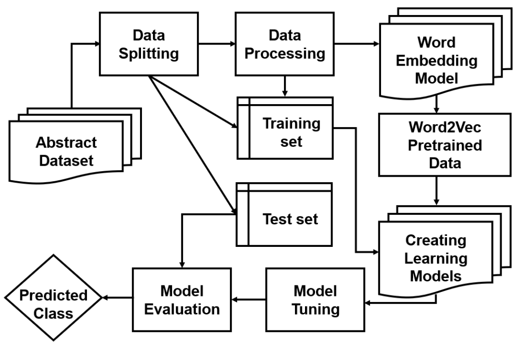


Figure 1. System flow

Text processing is an important component of text-classification systems. First, we split and tokenize the abstracts into words. The unnecessary objects, such as stop words, punctuation, digits, URLs, and website links, are then removed using the Natural Language Toolkit (NLTK) [21]. Finally, we convert verbs and adverbs into nouns using the NLTK function. Table 1 summarizes the data types and the number of items composed after the text processing mentioned the above is applied.

Table 1. Text processing summary

Data type	Number of items
Documents	5,659
Sentences	46,511
Word (vocabulary)	228,407
Unique words	12,969

3.2 Word Embedding Architecture

Word embedding is an essential facet of NLP, which typically represents words in a text using a multidimensional real-valued vector for classification and analysis. The vector represents the meaning of the surrounding words, which are likely to have a meaning similar to that of the target word. The Word2Vec text classification algorithm can be used in conjunction with two learning algorithms, Continuous Bag-of-Words (CBOW) and Skip-gram (Sg), as shown in Figure 2.

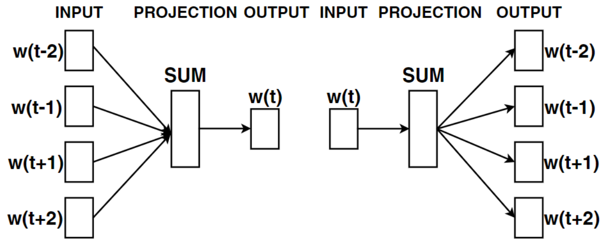


Figure 2. CBOW and Sg model architectures

The CBOW approach in Word2Vec focuses on predicting a target word based on its context. By considering the probability distribution of words in the given context (words surrounding the target word), CBOW employs a neural network with a hidden layer to learn word embeddings. In this architecture, the input layer represents the context words, and the output layer predicts the target word. Throughout the training process, the model adjusts its weights to minimize the disparity between the predicted and actual target words, ultimately resulting in learned weights that serve as word embeddings. In contrast, Sg model, also part of the Word2Vec algorithm, takes a target word as input and endeavors to predict the context words within a specified window. This design allows the Sg to capture the contextual information of a word by predicting the words likely to appear in its vicinity. The neural network architecture of Sg is reversed compared to CBOW. In Sg, the input layer represents the target word, and the output layer predicts the context words. Similar to CBOW, the model adjusts its weights during training to minimize the difference between the predicted and actual context words. The choice between CBOW and Sg often depends on the characteristics of the dataset and the specific nuances of the natural language processing task at hand.

3.3 Pretrained Word2Vec Model

We apply Word2Vec embedding methods (CBOW and Sg) to pretrained the abstract dataset. The Gensim module [22] is used specifically for word embedding preprocessing. This module plays an essential role in generating the Word2Vec pretrained model used in this study. Table 2 summarizes the preprocessing of the word embedding for each of CBOW and Sg models.

Table 2. The word embedding processing

Preprocessing parameters	CBOW model	Sg model
Type of data	Abstract text	Abstract text
Dimensionality	100	100
Window size	5	5
Minimum word count	50	50
Number of workers	3	3
Number of iterations	100	100

4 Basic Structure of Classification Models

4.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specialized models designed for feedforward multilayer neural networks. They are initially developed for image recognition and have made significant breakthroughs in the field of image processing. Interestingly, CNN have found wide applications in text classification as well. The architecture of a classical CNN model for text classification is depicted in Figure 3.

A Deep Neural Network (DNN) comprises two or more hidden layers and stands as a fundamental element in deep learning. While a three-layer structure is commonly used in most applications, certain fields, such as image and voice recognition, to achieve the desired result use multiple hidden layers in deep neural networks.

The neurons in CNNs are inspired by the visual cortex of the human brain, making them particularly effective in image recognition. Unlike traditional pattern classification using neural networks, CNNs differ fundamentally as they serve both for feature extraction and classification. In a pattern classifier using CNNs, the original input data is directly input into the neural network without additional preprocessing. The pattern is then classified through the process of feature extraction.

To address issues of topological constancy, such as movement or distortion, CNNs employ pooling. The basic structure of CNNs involves a series of convolutional layers and pooling layers, as illustrated in Figure 3.

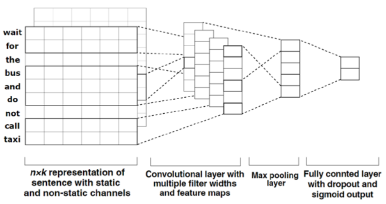


Figure 3. CNN architecture for text classification [4]

Feature Extraction Neural Network: This network, tasked with extracting features from input data, comprises repeated convolutional and pooling layers. The convolution layer extracts local features using a spatial filter, generating a feature map. Multiple feature maps can be obtained by employing multiple filters for various regional features. The pooling layer reduces the size of the feature map, ensuring topological homeostasis and representing surrounding weights.

Neural Network for Classification: Located at the back of the feature extraction neural network, the classification network functions similarly to existing neural network classifiers. It consists of three layers, with interconnected neurons in the input, hidden, and output layerss.

In summary, CNNs play a pivotal role in both feature extraction and classification, making them highly effective for tasks such as image and text classification. Their unique architecture, inspired by the visual cortex, allows them to directly process input data without the need for extensive preprocessing, showcasing their versatility and efficiency in various domains.

4.2 LSTM and GRU Architecture

A. LSTM

Long Short-Term Memory (LSTM) networks represent a specialized type of Recurrent Neural Network (RNN) designed to more effectively handle long-term dependencies compared to simple RNN. Initially introduced by S. Hochreiter and J. Schmid Huber [7], the LSTM architecture has undergone further development by several researchers. LSTM incorporate multiple gates, including the input gate, erase gate, cell state, output gate, and hidden state, to meticulously regulate the flow of information within each node state. This strategic use of gates proves particularly advantageous in addressing the vanishing-gradient problem commonly encountered in training deep networks. The cores unit of the LSTM model, as depicted in Figure 4, comprises these essential components:

Input Gate: Responsible for regulating the flow of new information into the cell state.

Erase Gate: Controls the removal or updating of information from the cell state.

Cell State: Represents the memory or information retained in the cell.

Output Gate: Governs the amount of information released to the output and hidden state.

Hidden State: Represents the output of the LSTM cell and carries information to the next time step.

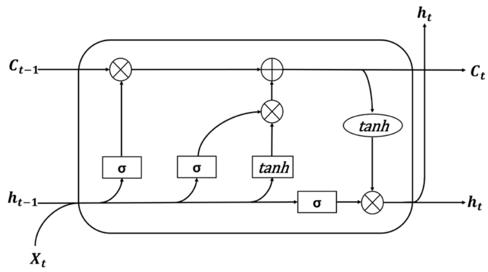


Figure 4. LSTM architecture

The incorporation of these gates allows LSTM to effectively manage the flow of information, mitigating the vanishing-gradient problem that can impede the training of deep neural networks. The LSTM architecture's ability to capture and retain long-term dependencies makes it well-suited for a variety of sequential data tasks, such as natural language processing and time series prediction.

B. GRU

The Gated Recurrent Unit (GRU) stands as a streamlined variant of the Long Short-Term Memory (LSTM) architecture. Distinguished by its simplicity, the GRU comprises solely two gates: an update gate and a reset gate. Despite its similarities to LSTM, GRUs possess fewer parameters, notably lacking an output gate. This reduction in parameters contributes to computational efficiency while still maintaining competitive performance in various evaluations and tasks. Notably, the GRU, introduced by J. Chung and

C. Gulcehre et al. [8], is recognized for its faster learning rate compared to LSTM. The fundamental cell of the GRU model, illustrated in Figure 5, includes the following key components:

Update Gate: Regulates the flow of new information into the hidden state, determining the degree of update.

Reset Gate: Controls the extent to which past information is reset or ignored during the computation of the new hidden state.

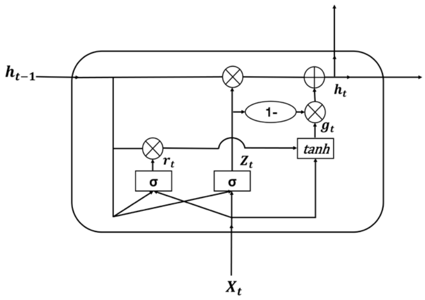


Figure 5. GRU architecture

Simplifying the GRU architecture can be demonstrated while increasing computational efficiency by using fewer gates and is also effective for learning complex temporal dependencies. Its ability to yield results comparable to those of LSTM in various evaluations and tasks has contributed to the widespread adoption of GRU.

LSTM node:

$$\begin{cases} i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\ f_t = (W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ C_t = f_{t^o} C_{t-1} + i_{t^o} g_t \\ 0_t = \sigma(W_{x0}x_t + W_{h0}h_{t-1} + b_0) \\ h_t = 0_t \, o \tanh(c_t) \end{cases}$$

GRU node:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$

$$g_t = \tanh(W_{hg}(r_t o h_{t-1}) + W_{xg}x_t + b_g)$$

$$h_t = (1 - z_t) o g_t + z_t o h_{t-1}$$

where, four sets of weights within each gate: W_{xi}, W_{xg}, W_{xf}, W_{x0} are associated with the input gate, and W_{hi}, W_{hg}, W_{hf}, W_{h0} are associated with the hidden state. Additionally, there are four biases for each gate b_i, b_g, b_f, b_0 . These weights and biases play a crucial role in determining the flow of information and the state of the network at each time step. In this work, the input gate is the gate to store the current information. First, the value at the current time t, denoted as x, multiplied by the weight W_{xi} leading to the input gate, and the hidden state of the previous time $t-1$, denoted as h_{t-1} multiplied by the weight W_{hi} leading to the input gate. These two products are then processed through the sigmoid function, resulting in a value referred to as i_t . Moreover, the current time step, denoted as t, is characterized by the product of the input sequence represented as x and the weight matrix W_{xg} , contributing to the input gate. Simultaneously, the hidden state value from the previous time step, denoted as h_{t-1}, is incorporated through the weight matrix W_{hg} , influencing the input gate. This input gate subsequently undergoes the hyperbolic tangent function, resulting in the generation of a distinctive value termed g_t.

- The input gate value i_t is obtained as follows: $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$, where $\boldsymbol{\sigma}$ represents the sigmoid function.
- The hyperbolic tangent function is applied to obtain the value g_t : $g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$, where ***tanh*** is the hyperbolic tangent function.

These computations are crucial in the context of recurrent neural networks, where they contribute to the determination of the input and memory content at a given time step. The sigmoid function restricts the values of i_t between 0 and 1, while the hyperbolic tangent function processes g_t . With these two values, we determine the amount of information to remember the chosen time. When the output value f_t of the erasure gate becomes 0, the value C_{t-1} of the cell state at the previous time becomes 0, and only the result of the input gate determines the value C_t of the cell state at the current time.

4.3 Bidirectional RNN Architecture

Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU) are RNN architectures primarily applied in natural language processing and the processing of long data sequences [8]. These models are designed to capture relationships between future and past words within a sequence. Unlike standard LSTM and GRU architectures, BiLSTM and BiGRU facilitate bidirectional information flow, enabling them to leverage information from both preceding and succeeding elements. This bidirectional capability proves to be instrumental in modeling sequential dependencies between words and phrases in both directions of a sequence.

In BiLSTM and BiGRU, additional LSTM and GRU layers are incorporated to reverse the direction of information flow. This means that the input sequence is processed in both the forward and reverse directions in these supplementary layers [9]. The outputs of these bidirectional layers are then combined through various methods, including averaging, summation, multiplication, or concatenation. Figure 6 illustrates model of the bidirectional architecture.

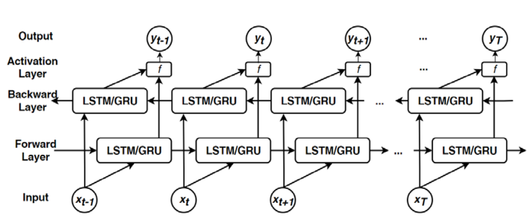


Figure 6. Architecture of bidirectional RNN

As depicted in Figure 6, the input sentence consists of sequential elements x_{t-1}, x_t, x_{t+1}, …, x_T. The Bidirectional RNN implemented operates bidirectionally, encompassing both the future (forward) and past (backward) directions. The hidden state of the Bidirectional RNN is utilized in both the forward and backward directions, with LSTM and GRU directions aligning with the forward and backward RNN. The Bidirectional RNN model employs an activation layer and generates predicted outputs y_{t-1}, y_t, y_{t+1}, …, y_T. This bidirectional architecture allows the model to leverage information from both preceding and succeeding elements in the input sequence, enhancing its ability to capture contextual dependencies and relationships within the data. The predicted outputs are obtained through the activation layer, reflecting the model's learned representations and predictions at various time steps in the sequence.

4.4 CNN and RNN Combine Models

In recent applications of text classification, Recurrent Convolutional Neural Networks (RCNN) have gained prominence [15]. The fundamental concept behind this technique is to capture contextual information through a repetitive structure and formulate a textual representation using a CNN. The RCNN architecture ingeniously merges the strengths of RNN and CNN to leverage the benefits of both techniques within a single model. Figure 7 illustrates the methodology of combining RNN and CNN in this hybrid architecture.

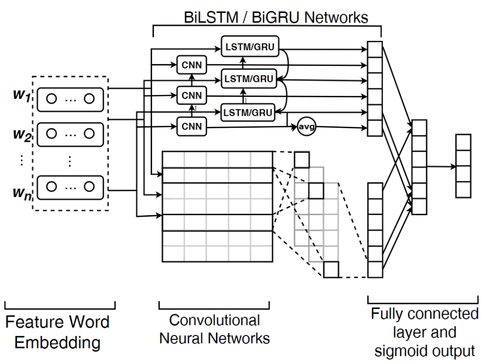


Figure 7. Architecture of CNN with BiLSTM and CNN with BiGRU combination models

As illustrated in Figure 7, the utilization of this model encompasses two distinct approaches. Initially, the model was implemented in the forward direction, wherein CNN with LSTM and CNN with GRU architecture. Additionally, we employ a combined of CNN with BiLSTM and CNN with BiGRU to enhance classification performance. Furthermore, traditional LSTM and GRU models are independently applied to compare with CNN-combined models. These experiments are specifically tailored to assess and compare the performance of these models.

5 Experiment and Results

In this section, we explore the performance of CNN and RNN models trained with Word2Vec techniques (CBOW and Skip-gram) for the classification and recommendation of research papers. For performance evaluation, we use precision, recall, and F1-Score as performance metrics [23].

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives}. \quad \textbf{(1)}$$

$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives}. \quad \textbf{(2)}$$

$$F1\;Score = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad \textbf{(3)}$$

where, *True Positives* is the number of instances predicted by the model to be in the positive category among the actual positive category data, and *False Negatives* is the number of instances predicted by the model to be in the negative category among the actual positive category data. On the other hand, *False Positives* is the number of instances predicted by the model as positive category among the actual negative category data. In these equations, *Precision* denotes the percentage of research paper abstracts that are real positives among abstracts classified as positive by the models in this study. *Recall* denotes the proportion of abstracts in which the models are classified as real positive abstracts, and *F1 Score* denotes the average of the weighted precision and recall scores. Furthermore, we use the Scikit-learn library [24] to separate research paper data into a training dataset and a testing dataset and evaluate the proposed models using these datasets.

5.1 Model Parameter Values

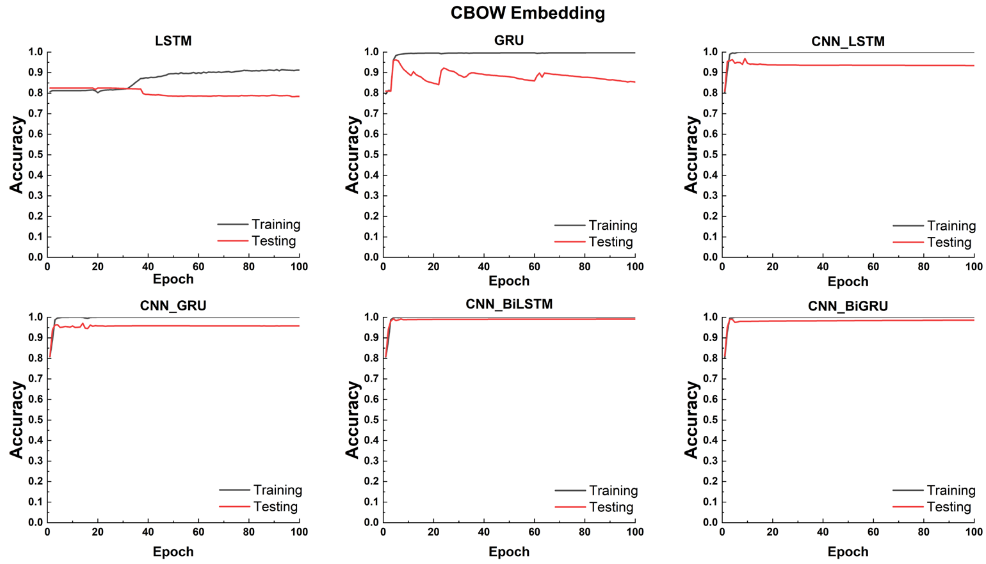
Word2Vec: In this study, we construct the Word2Vec model with all words in the abstract of research papers within the recent three years of FGCS journal. For this purpose, the number of dimensions of the embeddings is set to 100. The window size is set to 5, and 100 number of iterations are performed only for words that appeared more than once. The number of workers is set to 3. Moreover, the remaining parameters are applied as mentioned in Table 2.

CNN and RNN models: The application of pre-trained CBOW and Sg Word2Vec methodologies is integrated into the parameterization of CNN and RNN architectures [21]. Within the CNN layer, three distinct kernels (3, 4, 5) are employed as parameter values, each contributing to the convolutional operations. The number of feature maps for each layer is individually set to 256 filters in first layer, 128 filters in second layer, and 64 filters in third layer, with a maximum pooling size of 2 utilized to aggregate high-level feature maps effectively. Moreover, a dropout parameter value of 0.2 is applied in the convolutional layer to enhance the generalization capabilities of the model. Conversely, in the feed-forward LSTM and GRU layers, 64 memory units are employed to facilitate improved memory capacities. For the BiLSTM and BiGRU layers, 100 memory units are utilized, allowing the model to capture intricate sequential dependencies from both forward and backward directions. The convolutional and sigmoid filter weights are taken uniformly from the interval [0, 1]. The dense layer, also known as the fully connected layer, is incorporated to compute class probabilities, contributing to the final classification decisions.

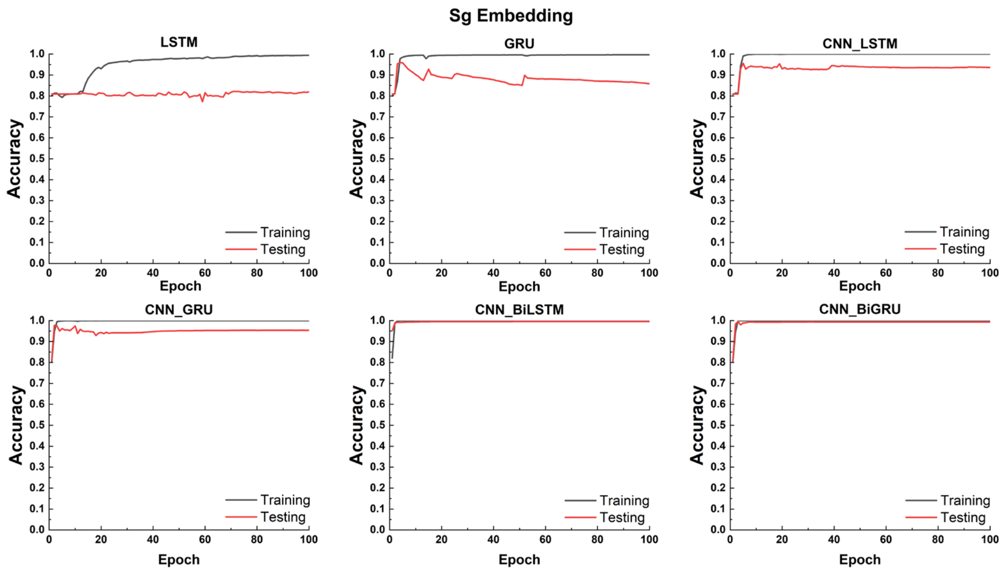
Training is conducted using the "Adam" optimization algorithm with 100 epochs per model. Additionally, a checkpoint is implemented through the callback function to save the model with maximum accuracy during the training process. This meticulous configuration ensures the models are well-equipped to learn and generalize from the input data, showcasing a systematic approach to parameter selection and optimization. Using these parameters, we conduct a comprehensive comparison of overall performance for the six types of variant CNN and RNN models. The following sections describe the results of these comparisons.

5.2 Comparison Results of Pretrained Word2Vec with CNN and RNN Models

Figure 8 shows the accuracies of pre-trained word embeddings with various types of CNN and RNN models for training data and testing data. First, we compare CNN with RNN combination models to traditional RNN models (LSTM and GRU). As we can see in Figure 8, traditional LSTM model with CBOW and Sg pre-trained embedding shows immutable results during training the model to capture long-term dependencies information. On the other hand, the traditional GRU model with CBOW and Sg pre-trained embedding provides slightly better accuracies than the traditional LSTM model, because the GRU model is effective in capturing short-term dependencies in sequences and potentially faster to train on parallel processing units.



(a) CBOW accuracy



(b) Sg accuracy

Figure 8. Comparison of accurary for CNN and RNN models with Word2Vec

Compared to traditional LSTM and GRU models, the CNN models combined with LSTM and GRU have better accuracies for training data by leveraging both CBOW and Sg pre-trained embeddings. The CNN models combined with LSTM and GRU can lead to improving generalization accuracy, because typical CNN models learn generic spatial features from input data and the recurrent units can adapt to diverse temporal patterns. Furthermore, CNN models combined with BiLSTM and BiGRU show superior accuracies to the other models, because the bidirectional processing in recurrent layers is especially effective in capturing long-term dependencies in sequential data.

In conclusion, our comprehensive analysis of model accuracies leads to affirm that the combination model outperforms the traditional baseline models. This superiority not only underscores its efficacy but also positions it as a promising choice for applications in research paper classification and recommendation systems. The amalgamation of features from different model architectures enhances its capability to capture intricate patterns and relationships, demonstrating its potential for advancing the state-of-the-art in these specific domains.

5.3 Comparison of F1-Score for CBOW and Sg Models

In this section, we present the results of performance evaluation in terms of F1-Score. Figure 9 shows the F1-Score results of six types of models utilizing both CBOW and Sg Word2Vec pre-trained embedding methods.

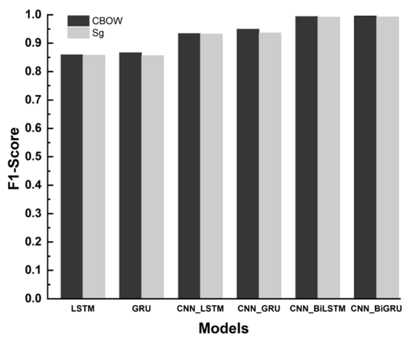


Figure 9. F1-Score results

Based on the results of Figure 9, it is evident that the traditional LSTM and GRU baseline models exhibit inferior performance in terms of the F1-Score when compared to the CNN models combined LSTM, GRU BiLSTM, and BiGRU. In-depth analysis, it is observed that these CNN models deliver enhanced performance when applied with the CBOW pre-trained embedding method. This is because the CBOW pre-trained embedding method is more predictive of a target word using the surrounding words as the input than Sg one. Meanwhile, the CNN models with BiLSTM and BiGRU surpass the performance of the CNN models with LSTM and GRU. This can be attributed to the proficiency of CNN models in extracting valuable features from input data, while BiLSTM or BiGRU effectively captures both forward and backward temporal dependencies in sequential data. This dual processing capability enables comprehensive feature extraction in both spatial and temporal dimensions.

In conclusion, the evaluation of model performance indicates that the CBOW pre-trained embedding model has superior performance compared to the Sg one, especially when used in conjunction with a CNN model combined with LSTM, GRU, BiLSTM, and BiGRU. Therefore, these combined CNN models can be deemed highly suitable for recommendation systems, due to their particular effectiveness in classifying research paper.

6 Conclusion

In this study, we evaluated the performance of six different types of models based on CNN and RNN architecture with Word2Vec embedding techniques for classification and recommendation of research papers. Six types of models were applied to the classification of research papers in FGCS journals, and the performances of these models were compared and analyzed in terms of accuracy and F1-Score. The combination models, CNN with LSTM, CNN with GRU, CNN with BiLSTM, and CNN with BiGRU, have relatively high accuracy as compared to the traditional LSTM and GRU models. The evaluation results showed that the CBOW pre-trained embedding technique performs better than the Sg one for each of combination models. Specifically, the CNN with bidirectional models (LSTM and GRU) have better performance than the other models. Therefore, the CBOW pre-trained embedding will be fit for the recommendation of research papers when applying to the CNN models combined with BiLSTM and BiGRU.

This study is expected to aid future studies on utilizing various deep learning and machine learning methods in the field of research paper classification and recommendation. In the future, we plan to address other techniques such as ELMo embedding, BERT, and so on.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2C1092934).

References

[1] A. M. Hilal, J. S. Alzahrani, H. Alsolai, N. Negm, F. M. Nafie, A. Motwakel, I. Yaseen, M. A. Hamza, Sentiment Analysis Technique for Textual Reviews Using Neutrosophic Set Theory in the Multi-Criteria Decision-Making System, *Human-centric Computing and Information Sciences*, Vol. 13, pp. 1-16, May, 2023.

[2] K. Park, S. Baek, J. Jeon, Y.-S. Jeong, Music Plagiarism Detection Based on Siamese CNN, *Human-centric Computing and Information Sciences*, Vol. 12, pp. 1-10, August, 2022.

[3] T. W. Lee, J. S. Park, J. G. Shon, Fake News Detection Using CNN-based Sentiment Change Patterns, *KIPS*

Transactions on Software and Data Engineering, Vol. 12, No. 4, pp. 179-188, April, 2023.
[4] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back, Face recognition: A convolutional neural-network approach, *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, pp. 98–113, January, 1997.
[5] N. Mehranbod, M. Soroush, C. Panjapornpon, A method of sensor fault detection and identification, *Journal of Process Control*, Vol. 15, No. 3, pp. 321–339, April, 2005.
[6] Q. Chen, Q. Xie, Q. Yuan, H. Huang, Y. Li, Research on a Real-Time Monitoring Method for the Wear State of a Tool Based on a Convolutional Bidirectional LSTM Model, *Symmetry*, Vol. 11, No. 10, pp. 1-18, October, 2019.
[7] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, November, 1997.
[8] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, December 2014. https://arxiv.org/abs/1412.3555
[9] R. Fu, Z. Zhang, L. Li, Using LSTM and GRU neural network methods for traffic flow prediction, *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Wuhan, China, 2016, pp. 324–328.
[10] Z. Huang, W. Xu, K. Yu, *Bidirectional LSTM-CRF models for sequence tagging*, August, 2015. https://arxiv.org/abs/1508.01991
[11] A. Graves, J. Schmidhuber, Framewise phoneme classification with Bidirectional LSTM and other neural network architectures, *Neural Networks*, Vol. 18, No. 5-6, pp. 602–610, July–August, 2005.
[12] O. Melamud, J. Goldberger, I. Dagan, Context2vec: Learning Generic Context Embedding with Bidirectional LSTM, *20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics (ACL)*, Berlin, Germany, 2016, pp. 51–61.
[13] L. Xiao, G. Wang, Y. Zuo, Research on Patent Text Classification Based on Word2Vec and LSTM, *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 71–74.
[14] Y. Luan, S. Lin, Research on Text Classification Based on CNN and LSTM, *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, 2019, pp. 352–355.
[15] A. U. Rehman, A. K. Malik, B. Raza, W. Ali, A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis, *Multimedia Tools and Applications*, Vol. 78, No. 18, pp. 26597–26613, September, 2019.
[16] J. Wang, L.-C. Yu, K. R. Lai, X. Zhang, Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp. 581–591, 2020.
[17] H. Xie, W. Lin, S. Lin, J. Wang, L. C. Yu, A multi-dimensional relation model for dimensional sentiment analysis, *Information Sciences*, Vol. 579, pp. 832–844, November, 2021.
[18] A. H. Ombabi, W. Ouarda, A. M. Alimi, Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks, *Social Network Analysis and Mining*, Vol. 10, No. 1, pp. 1–13, 2020.
[19] J. Zhang, Y. Li, J. Tian, T. Li, LSTM-CNN Hybrid Model for Text Classification, *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2018, pp. 1675–1680.
[20] L. Richardson, Beautiful Soup Documentation: crummy.com/software/BeautifulSoup/bs4/doc, Python package for HTML and XML parser documents, Web scraping, *Google Groups*, retrieved Sep. 27, 2022.
[21] S. Bird, E. Klein, E. Loper, *Natural Language Processing with Python*, O'Reilly Media Inc., 2009.
[22] Gensim: Topic of modelling for humans, https://radimrehurek.com/gensim/, accessed Dec. 11, 2022.
[23] D. M. W. Powers, Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation, *Journal of Machine Learning Technologies*, Vol. 2, No. 1, pp. 37–63. 2011.
[24] Scikit-learn: A set of python modules for machine learning and data mining, retrieved Sept. 24, 2022. https://scikit-learn.org/stable/

Biographies

Dipto Biswas received his B.S and M.S degrees in Computer Engineering from Daegu Catholic University, Korea in 2019 and 2021, respectively. He is currently pursuing a Ph.D. on Computer Engineering at Daegu Catholic University. He has recently been researching big data and artificial intelligence for research paper recommendation systems.

Joon-Min Gil received the Ph.D. degree in Computer Science and Engineering from Korea University, Korea in 2000. He is currently a professor at the Department of Computer Engineering, Jeju National University (JNU), Jeju, Korea. Before joining JNU in Sept. 2023, he was with Daegu Catholic University as a professor from Mar. 2006 to Aug. 2023. His recent research interests include cloud computing, big data computing, artificial intelligence, and distributed computing.