# Deep Learning Classification of Biomedical Text using Convolutional Neural Network

Rozilawati Dollah[1], Chew Yi Sheng[2], Norhawaniah Zakaria[3], Mohd Shahizan Othman[4], Abd Wahid Rasib[5]
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia[1, 2, 3, 4]
Program of Geoinformation, Faculty of Built Environment and Surveying,Universiti Teknologi Malaysia[5]
81310 Johor Bahru, Johor, Malaysia

***Abstract*—In this digital era, the document entries have been increasing days by days, causing a situation where the volume of the document entries in overwhelming. This situation has caused people to encounter with problems such as congestion of data, difficulty in searching the intended information or even difficulty in managing the databases, for example, MEDLINE database which stores the documents related to the biomedical field. This research will specify the solution focusing in text classification of the biomedical abstracts. Text classification is the process of organizing documents into predefined classes. A standard text classification framework consists of feature extraction, feature selection and the classification stages. The dataset used in this research is the Ohsumed dataset which is the subset of the MEDLINE database. In this research, there is a total number of 11,566 abstracts selected from the Ohsumed dataset. First of all, feature extraction is performed on the biomedical abstracts and a list of unique features is produced. All the features in this list will be added to the multiword tokenizer lexicon for tokenizing phrases or compound word. After that, the classification of the biomedical texts is conducted using the deep learning network, Convolutional Neural Network which is an approach widely used in many domains such as pattern recognition, classification and so on. The goal of classification is to accurately organize the data into the correct predefined classes. The Convolutional Neural Network has achieved a result of 54.79% average accuracy, 61.00% average precision, 60.00% average recall and 60.50% average F1-score. In short, it is hoped that this research could be beneficial to the text classification area.**

***Keywords—Convolutional neural network; biomedical text classification; compound term; Ohsumed dataset***

## I. INTRODUCTION

As a result of the increasing growth rate of the online biomedical text, researchers often encounter a lot of tough challenges. Due to the huge amount of biomedical document entries published online every day, the task of organizing the documents is getting more and more difficult. Eventually, researchers might retrieve irrelevant documents from online sources and they have to waste their precious time to filter and check whether the documents they found is the one they need actually. To deal with all these challenges and problems, classification is one of the effective yet efficient ways. Therefore, in this research, classification of the biomedical text by using deep learning neural network, Convolutional Neural Network will be the focus. By using CNN, researchers can save their precious time in searching for all the intended information.

### *A. Problem Background*

Biomedical literature is papers of scientific research which consists of the convincing idea, theories and results of research done in the medical field. The coverage of biomedical literature or journals are very wide, it could be research about the discovery of new drugs and cure for certain diseases, fresh yet useful information about certain diseases, the discovery of new protein and so on. These papers' main function is to allow communication between the researchers and the other researchers, scientist or even people who may not be trained as a scientist or physician such as students and so forth. Most of the time, biomedical literature are very long and complicated, it might be consisted of up to hundred pages and in these papers, there are also a lot of complex scientific terms which make it more difficult for the researchers to read and find their desired information. With the overwhelming number of biomedical literature available nowadays, researchers are facing a lot of difficulties when they wanted to retrieve their desired information in their field of study [1].

Text classification is an important component in many applications, for instance, web searching, information filtering and sentiment analysis [2][3]. By doing text classification, we are assigning predefined categories to the free-text documents [4]. It is also can be known as text categorization or document categorization. The classification of text is mainly done by using machine learning algorithms such as feature engineering, feature selection and so forth. However, text classification using machine learning approaches might have data sparsity problem [5]. To deal with this problem, we apply deep learning principle, Convolutional Neural Network (CNN) instead of the machine learning approach.

Recently, deep learning approaches have also been used to do text classification. Convolutional Neural Network is one of the deep neural networks and it is believed that this neural network can solve the data sparsity problem [6]. It is very useful in extracting information from raw signals, ranging from computer vision applications to speech recognition and so forth [7]. In this research, we will focus on using Convolutional Neural Network to classify biomedical text abstracts and to measure the effectiveness of using Convolutional Neural Network in text classification.

## II. RELATED WORKS

Deep learning has gained attractions from may researchers recently as the deep learning approach in classification can produce good results which can be compared to the machine learning approach. A well known deep learning-based approach, Convolutional Neural Network is proposed to perform granite rocks classification [8]. The results obtained from the experiment showed the performance of some of the networks would be better when they are applied together. Convolutional Neural Network in image classification on the different type of datasets such as remote sensing data of aerial images and scene images from SUN database [9]. The experimental results based on the graphical representation and quality metrics showed that CNN can produce a fairly good result for all the tested datasets. They concluded that a low Mean Squared Error (MSE), high classification accuracy and shorter training time can be achieved by using enough number of epochs, the number of iterations.

CNN was also applied to the biomedical domain text classification to identify the hallmarks of cancer associated with publication abstracts [10]. The data that had been used in this research is the text document from the online source corpus. The experimental results showed that a competitive performance with the state-of-the-art SVM-based classifier can be achieved. A simple CNN with only one convolution layer can perform notably well and an unsupervised pre-training of the work vectors is very essential in deep learning for natural language processing [11].

A recurrent convolutional neural network to perform text classification [5]. The proposed model is outperforming the traditional recurrent neural network and convolutional neural network as the contextual information can be captured with the use of recurrent structure and then the representation of the text is done using the convolutional neural network.

## III. Methodology

This research could be divided into five phases, Ohsumed dataset collection phase, text pre-processing phase which involved feature extraction, text classification phase, deep learning architecture phase as well as evaluation and validation phase. The details of each phase will be discussed as follow:

### A. *Ohsumed Dataset Collection Phase*

The Ohsumed datasets might contain different levels from the first level until the fourth level and this research will focus on 11,566 abstracts of biomedical journal which are from first and second levels only. All the categories and the number of abstracts for each category used in this research is stated in Table I.

TABLE. I. List of Selected Categories with Document Numbers

| **Category Name** | **Number of Documents** |
|---|---|
| Arrhythmia | 1,173 |
| Coronary Disease | 4,235 |
| Heart Arrest | 513 |
| Heart Defects, Congenital | 718 |
| Heart Failure, Congestive | 1,295 |
| Heart Valve Diseases | 335 |
| Myocardial Diseases | 355 |
| Myocardial Infarction | 2,942 |
| **TOTAL ABSTRACTS USED: 11,566 ABSTRACTS** | |

### B. *Text Pre-processing Phase*

Text pre-processing is an important process to extract the important and informative features before we can classify the biomedical text [12]. In this phase, feature extraction is done to retrieve the most relevant information from the training set and represent that information in a lower dimensionality space. Feature extraction is performed by using the GENIA tagger tool, a tool which is specially designed to extract information from biomedical text. Fig. 1 shows the output produced by the GENIA tagger tools, all words in the text are labeled with post-of-speech (POS) tags, chunk tags and named entity tags. The outputs produced by GENIA tagger tools will be processed again by selecting only the noun phrases which all the phrases are labeled with NP tag. After that, the stop words, general terms and duplicate terms will be removed from the text as it might be able to influence the result of classification afterward. All the remaining features will be grouped to form a vocabulary with the total number of 22,176 features and all these features will be added into the multiword tokenizer lexicon to be used in the process later.

### C. *Deep Learning Architecture Phase*

Generally, deep learning text classification model architecture used in this research consists of several components and all these components are connected sequentially. For instance, the components in the architecture included the input, biomedical abstracts, word embedding layer, deep network which is made up of convolutional layers and max-pooling layer, fully connected layer and the output which is the classification result. The deep learning text classification model architecture used in this research is shown in Fig. 1.

The input which is the biomedical abstracts will go through the process of tokenizing using the multiword tokenizer instead of the single word tokenizer. The reason why multiword tokenizer is used instead of the single word tokenizer is because the dataset used in this research is all biomedical abstracts which contain many biomedical terms, all these biomedical terms mostly are compound terms which are built by combining two or more single term and these compound terms carry different meaning with the single term.

Next, the word embedding layer must be set up before the pre-processed text input passes through. The purpose of this layer is to transform all the words in the text that have the same or similar meaning to have a similar representation in the form of a vector, it is a good technique which can be used to acquire continuous low-dimensional vector space representation of words [13]. Keras, a deep learning framework used in this research provides the embedding layer to handle this word embedding; it stores a lookup table for mapping between the words in the biomedical abstracts and the dense vector representations. In this research, a pre-trained BioASQ word vector is used [14].

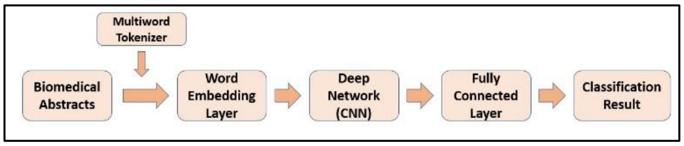

Fig. 1. Deep Learning Text Classification Model Architecture.

The sequence of embedding vectors obtained from the previous process will be converted into a compressed representation with all the information in the sequence of words in the text captured completely and afterward the stack of convolutional layer and max-pooling layer take it as the input. First, the convolutional layer consists of trainable kernels which also known as filters which is functioned to detect any specific features in the input. They will slide or convolve across the one-dimensional input and produce an activation map. A set of activation maps is produced from the different convolution process which detects different features and passed to the max-pooling layer.

The activation function used in the proposed model is the Rectified Linear Unit (ReLU). ReLU function does not activate all neurons at the same time and hence it is very efficient in term of computational time. Next, the max-pooling layer will take the transformed output from the convolutional layer as input and this layer functions to reduce the computation complexity and the spatial dimension without dumping the momentous information. In this research, the momentous information mentioned before this is the significant features. The output from the max-pooling layer will be taken as the input of the fully connected layer.

Next, the fully connected layer is where the classification performed based on the features extracted from the stack of convolutional layers and max-pooling layers. In this research, softmax function with categorical_crossentropy loss function is used as we are solving a multi-class classification problem. Softmax function here functions to apply a transformation to the output obtained from the previous layers so that the final output can be interpreted as a probability vector for each class or class scores.

During the training process of the model, the cross-validation method is used to reduce the problems like model overfitting and give an insight on how well the model can generalize to each independent dataset. In this research, stratified k-fold cross-validation with *k* equals to 10 are used to evaluate the performance of the model as 10 folds is believed to have the smallest mean squared error as well as variance in the estimation of prediction errors [15]. Stratified 10 folds validation split the entire dataset into 10 parts and it will take one part of the dataset to test the model while the other parts of the dataset will be used to train the model. Besides, it takes at least *m* instances from each of the classes for the training process at each fold to prevent a situation where the data from the certain classes are over-represented. Cross-validation is important to ensure that every single part of the dataset is used is to train the model. The parameters used for the proposed model is shown in Table II.

### D. Performance Evaluation and Validation Phase

During the training process, the dataset is split into the training set and validation set. In this research, the training set consists of 80% of the whole dataset while the validation set consists of 20% of the whole dataset. The training set is purposely used to train the model while the validation set is used to evaluate the model's performance. Metrics on the training set like the training loss and training accuracy allow us to know the progress of the model in terms of training while the metrics on the validation set like the validation accuracy and validation loss allow us to measure the quality of the model in terms of the capability to make prediction based on the new data.

TABLE. II. DESCRIPTION AND VALUE OF USED PARAMETERS FOR THE PROPOSED MODEL

| Parameter | Description | Value |
|---|---|---|
| Epoch | Number of the forward and backward pass of all the training samples | 10 |
| Filter size | The dimension of the filter in the ConvNet | 128 |
| Kernel size | Size of the convolutional filter | 3 |
| Batch size | The total number of datasets that will be propagated through the network | 128 |

Generally, a good model should have perfect fitting where the training loss is roughly the same as the validation loss. To reduce the effect of overfitting, dropout can be inserted after each max-pooling layer. This can also decrease the training time and result in better performance. In this research, there are other evaluation methods such as precision, recall and F1-measure also have been used to evaluate the proposed model. In this research, these three evaluation methods are performed by using the scikit-learn classification report. The formula to calculate the precision, recall and F1-measure are as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{F1- score} = 2 * \frac{precision*recall}{precision+recall} \quad (3)$$

Where TP is true positive, FP is false positive and FN is false negative.

## IV. EXPERIMENTAL RESULTS

There are many sets of experiments have been carried out to evaluate the performance of the proposed model. All these experiments are evaluated based on the classification report, the values of precision, recall, and F1- score. The manipulated methods used to test the proposed model included the utilization of single word tokenizer or the multiword tokenizer as well as the number of sets of convolutional and max-pooling layers used in the Convolutional Neural Network architecture.

The sequence of embedding vectors obtained from the previous process will be converted into a compressed representation with all the information in the sequence of words in the text captured completely and afterward the stack of convolutional layer and max-pooling layer take it as the input. First, the convolutional layer consists of trainable kernels which also known as filters which are functioned to detect any specific features in the input. They will slide or convolve across the one-dimensional input and produce an activation map. A set of activation maps is produced from the different convolution process which detects different features and passed to the max-pooling layer.

Overall, the performance of the proposed model using the multiword tokenizer is decreasing as the number of sets of the convolution and the max-pooling layer is increasing in terms of the precision, recall as well as the F1-measure. This is because one set of the convolution and max-pooling layer is enough to extract features from the biomedical text and carry out the classification of the biomedical text. As the number of sets of convolution and max-pooling layer increasing, the model tends to extract more features which are irrelevant to be used in the classification task. The performance measure for the experiment using multiword tokenizer with one set of the convolutional and the max-pooling layer is shown in Fig. 1.

The performance of the proposed model using the single word tokenizer has the same trend as the one using multiword tokenizer which is the performance of the model is declining as the number of sets of layers is increasing. However, the performance of the model using the single word tokenizer is better compared to the model that used multiword tokenizer. The performance measure for the experiment using multiword tokenizer with one set of the convolutional and max-pooling layer is shown in Fig. 2.

Looking deeper into the performance measure obtained for both sets of experiments, we can see that the category with the high number of documents is having a better performance compared to the category with the low number of documents. For instance, categories like coronary disease and myocardial infarction which consists of 4,235 documents and 2,942 documents respectively, they have achieved better overall performance compared to other categories. On the other hand, categories like Heart Valve Diseases and Myocardial Diseases that consists of 335 and 355 documents which considered as a very low number of documents, they have the worst precision, recall and F1-score in all three sets of the experiment conducted which are 0 for precision, recall and F1-measure. The model performance on categories like Heart Valve Diseases is completely zero in terms of precision, recall and F1-score. This is because, in each category, there are different biomedical terms, as this category has a smaller size compared to the other categories, the features that can be used to classify the biomedical text will be less as well (see Fig. 3).

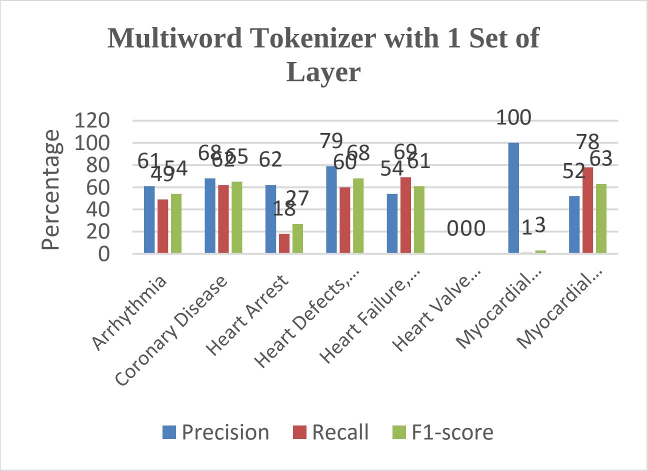

Fig. 2. Result of the Experiment using Multiword Tokenizer with 1 Set of Convolution and Max-Pooling Layer.

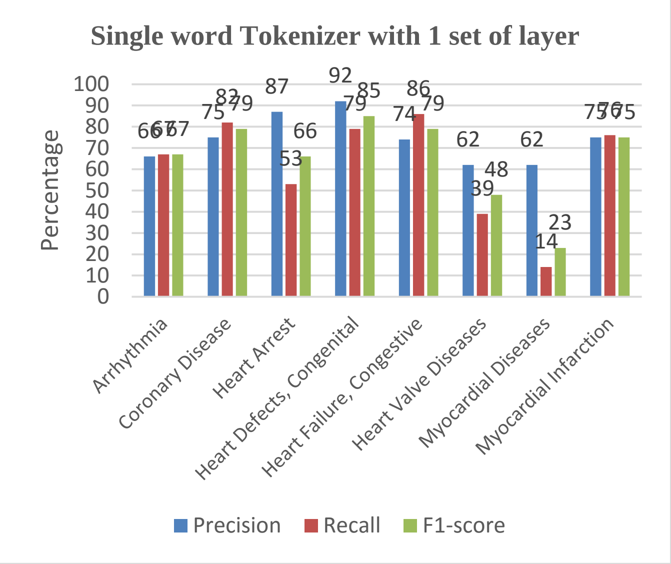

Fig. 3. Result of the Experiment using Single Word Tokenizer with 1 Set of Convolution and Max-Pooling Layer.

TABLE. III. COMPARISON BETWEEN THE RESULT OF THE EXPERIMENT USING MULTIWORD TOKENIZER AND SINGLE WORD TOKENIZER

| | Average Accuracy (%) | Average Precision (%) | Average Recall (%) | Average F1-score (%) |
|---|---|---|---|---|
| Using multiword tokenizer | 54.79 | 61.00 | 60.00 | 60.50 |
| Using single word tokenizer | 70.64 | 75.00 | 75.00 | 75.00 |

Fig. 4 and Table III demonstrate the comparison between the experimental result using the multiword and single word tokenizer. In this comparison, both experiments are using the same architecture, one set of convolution and max-pooling layer since this is the most ideal architecture that achieved the best performance among the others. The model with the use of single word tokenizer is outperforming the model with the used of the multiword tokenizer. This is because the single word tokenizer tokenizes each word in the text into single tokens which carry less or no meaning to pass through the word embedding layer. Unlike what is done by the multiword tokenizer which tokenizes the words in the text according to the words added into the lexicon at the early stage. The outcome would be a list of tokens which included single terms as well as the compound terms. This is important as the dataset used in this research is biomedical text which consists of a lot of biomedical terms that should be taken in consideration when doing text classification.

Overall, the result of experiment II is best among the three sets of experiments as shown in Fig. 5. Although the average precision, recall and F1-score did not show a very satisfying value, but if we compare the average precision and average recall, we can see that in Experiment II, the average precision is 69% and the average recall is 67% has only a difference of

2% which indicates that almost all the documents retrieved are relevant. This is because the dataset B has more documents in each category and hence there will be more documents involved in the training process which in turn give a better performance. Experiment III gives the worst result among the three sets of experiments; it achieved 73% for average precision and 60% for average recall. This indicates that in Experiment III, the number of documents retrieved is less relevant as there is a difference of 13% between the value of average precision and average recall. This scenario is caused by a small number of documents involved in the training process. In conclusion, to have a good deep learning model performance, the size of the dataset used must be large enough.

Fig. 6 illustrates the comparison of training accuracy and validation accuracy for the model using multiword tokenizer and one set of convolution and max-pooling layer as the model architecture and the number of epochs in this experiment is set to 10 epochs. We can observe that both the training accuracy and validation accuracy have the same trend of increasing. Overfitting is a scenario where the model tends to memorize the data instead of learning it. The degree of overfitting is indicated by the gap between the training accuracy line and the validation accuracy line, smaller gap means less overfit and vice versa. In this research, the degree of overfitting is minimized by adding dropout after the convolution and max-pooling layer. The degree of overfitting can also be known by comparing the training loss and validation loss. In a model with the perfect fit, the training loss should always lower and roughly the same with the validation loss. In this case, the difference between the training loss and the validation loss is not more than 0.2 which indicates that the added dropouts have successfully reduced the degree of overfitting.

The results of this research is compared to the result conducted by Hughes *et al.* [16] and it is shown in Fig. 7, we can see that the methods used in this research which is the combination of CNN, Multiword Tokenizer and Word2vec is outperformed compared to the (BOW + LogR) methods, this might be caused by the features used for classification in our proposed method are more informative than the features used in (BOW + LogR) method. On the other hand, the proposed method in our research is less perform compared to (CNN + Word2vec), because using different tokenizer will result in a different list of features. In their research, they use only the single word tokenizer and the features used for the classification might be less meaningful compared to the features used in our research, for instance, features "lung" and "cancer" are less informative than feature "lung cancer" and hence resulting the better performance in terms of accuracy, 68% compared to our proposed method which achieved only 54%. Hence, it can be concluded that a CNN-based approach with the use of multiword tokenizer can be used to conduct biomedical text classification and produce a better performance.

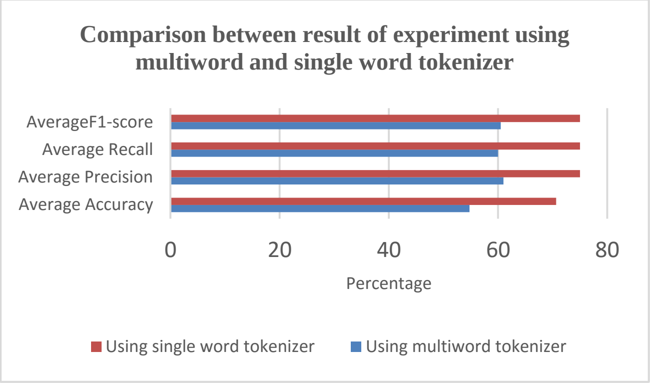

Fig. 4. Comparison between the Result of an Experiment using Multiword and Single Word Tokenizer.

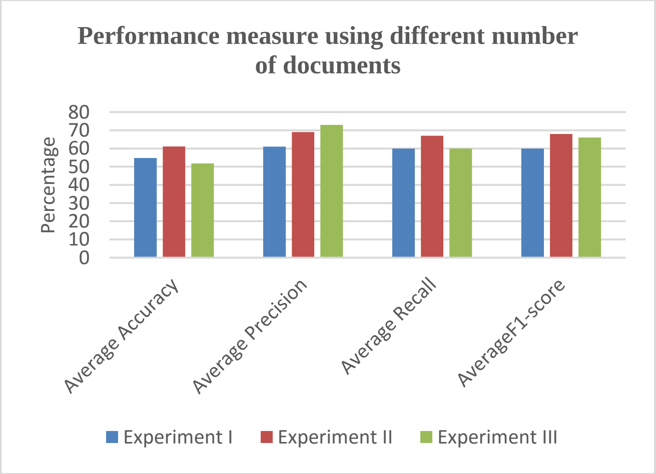

Fig. 5. Performance Measure using different Number of Documents.

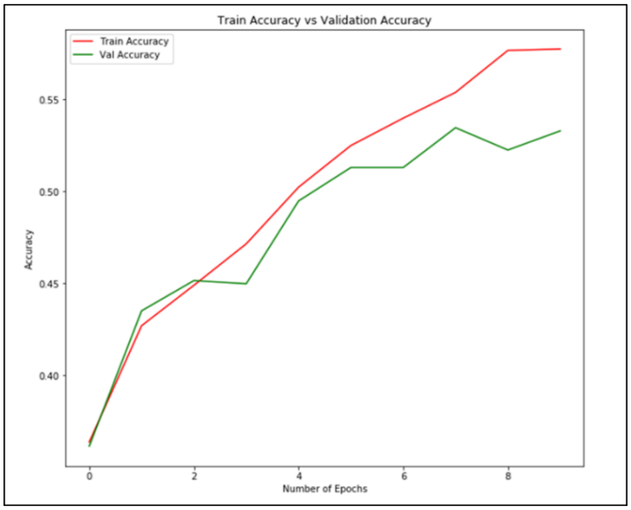

Fig. 6. Comparison of Training Accuracy and Validation Accuracy.

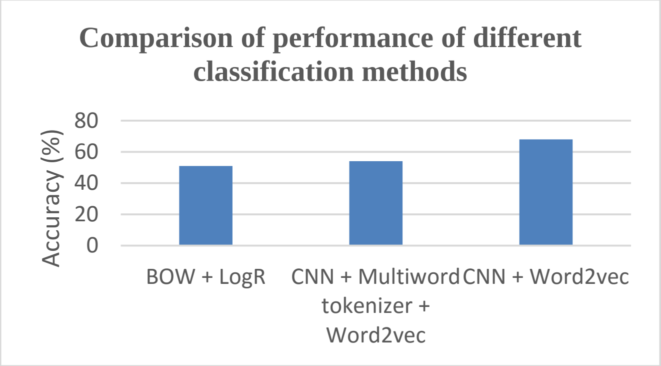

Fig. 7. Comparison of Performance of different Classification Methods.

## V. Conclusion

In this study, Convolutional Neural Network was used to perform classification of biomedical text and a result of average accuracy of 54.79%, average precision of 61.00%, average recall of 60.00% and average F1-score of 60.50% were obtained. It did take consideration of the biomedical terms in the biomedical text by using multiword tokenizer, biomedical terms which are mostly made up of two or more terms were tokenized into compound tokens and used for the CNN to perform text classification. All and all, the proposed method can increase the recall percentage, in other words, increase the number of documents being retrieved and classified correctly and indeed it is a good approach to be used in the classification of biomedical text.

## VI. Future Works

Firstly, a new word vector could be retrained by using all the biomedical text from the entire online biomedical text repository instead of just using the pre-trained BioASQ word vector which involves only biomedical text from PubMed. This is to ensure that the single term tokens or compound term tokens both can be assigned with meaningful vectors when they are passing through the word embedding layer. Next, a dataset which is greater in size could be used. This is because a deep learning neural network needs a lot of data in the training process. In other words, it needs a lot of data for the learning purpose. All in all, CNN is really an effective yet efficient approach to do classification tasks.

## Acknowledgment

This study is supported by the Fundamental Research Grant Scheme (FRGS) under the Vote No. 4F559 that sponsored by Ministry of Higher Education (MOHE), Research University Grant Scheme (RUG) under the Vote No. 13J94 and UTM Industry/International Incentive Grant (IIIG) under the Vote No. 4C255. The authors are greatly obliged to Universiti Teknologi Malaysia (UTM) and Soft Computing Research Group (SCRG) for the support and motivation.

## References

[1] M. Pavlinek, and V. Podgorelec, "Text classification method based on self-training and LDA topic models," Expert Systems with Applications, vol. 80, pp. 83–93, September 2017.

[2] C. C. Aggarwal, and C. Zhai, "A Survey of Text Clustering Algorithms," Mining Text Data, Springer, pp. 77-128, January 2012.

[3] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili, "A novel multivariate filter method for feature selection in text classification problems," Engineering Applications of Artificial Intelligence, vol. 70, pp. 25-37, April 2018.

[4] M. M. Mirończuk, and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," Expert Systems with Applications, vol. 106, pp. 36–54, September 2018.

[5] S. Lai, L. Xu, K. Liu, and J. Zhao, (2015). Recurrent Convolutional Neural Networks for Text Classification. Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2267–2273.

[6] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," Pattern Recognition, vol. 77, pp. 354–377, May 2018.

[7] X. He, and W. Zhang, "Emotion recognition by assisted learning with convolutional neural networks," Neurocomputing, vol. 291, pp. 187–194, May 2018.

[8] A. Ferreira, and G. Giraldi, "Convolutional Neural Network Approaches to Granite Tiles Classification Convolutional Neural Network approaches to granite tiles classification," Expert Systems With Applications, vol. 84, pp. 1–11, October 2017.

[9] D. Jaswal, V. Sowmya, and K. P. Soman, "Image Classification Using Convolutional Neural Networks," International Journal of Advancements in Research & Technology, vol. 3, Issue 6, pp. 1661–1668, June 2014.

[10] S. Baker, A. Korhonen, and S. Pyysalo, "Cancer Hallmark Text Classification Using Convolutional Neural Networks," in Proceeding of the Fifth Workshop of Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016), pp. 1–9, December 2016.

[11] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751, October 2014.

[12] V. Gurusamy, and S. Kannan, "Preprocessing Techniques for Text Mining," in Conference Paper, India, October 2014.

[13] C. Y. Liang, W. Jin, K. R. Lai, and Z. Xuejie, "Refining Word Embeddings for Sentiment Analysis," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 534-539, September 2017.

[14] I. Pavlopoulos, A. Kosmopoulos, and I. Androutsopoulos, "Continuous Space Word Vectors Obtained by Applying Word2Vec to Abstracts of Biomedical Articles," Word Journal Of The International Linguistic Association, pp. 1-4. March 2014.

[15] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, pp. 1137–1143, August 1995.

[16] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, "Medical Text Classification using Convolutional Neural Networks," Studies in Health Technology and Informatics 235, pp. 246–250, April 2017.