Harla Journals

Harla Journal of Engineering and Technology

Journal home page: https://journals.ddu.edu.et/index.php/HJET

Harla journal of Engineering and Technology

Afaan Oromo Document Text classification using Single layer Multisize Filters Convolutional Neural Network

[1]Kasahun Teshome

Computer Science, Mettu University, Oromia, Ethiopia

kasutesh1212@gmail.com

Abstract

Text classification is one of the most widely used natural language processing technologies. It is the technique that classifies unstructured text data into meaningful categorical classes. With the continuously increasing amount of online information, there is a pressing need to classify text for valuable information. Previously, many researchers have done Afaan Oromo text classification using machine learning methods. However, most of these traditional methods use TF-IDF, a Bag of words, to map some representation of the input data to a predefined set of meaningful outputs. However, these methods ignore the context and internal hierarchy of the text. In addition, they treat labels as independent individuals while ignoring the relationships between them, which also leads to a significant loss of semantic information; these deep learning approaches can solve these limitations. So, in this study, we use a Single layer Multi-Size Filters Convolutional Neural Network for document text classification and collect a dataset that contains 6450 documents organized into ten classes. We also look at how preprocessing approaches affect the performance of the model. In conclusion, after hyperparameter tuning the model, the performance of SMF-CNN was evaluated using Fast-Text pre-trained and Word2vec pre-trained word embedding, as well as without using pre-trained word embedding. The experimental results show Single-layer Multi-Size Filters Convolutional Neural Network performance can achieve effectiveness and good scalability of the accuracy of 96.81%, with Fast-Text pre-trained word embedding.

Keywords: Text classification, Document text classification, Afaan Oromo, Convolutional neural network,

1. Introduction

In natural language processing, text classification is a critical task; it is a behavior of dividing a set of input documents into one or more classes. In particular, document text is proliferating due to the increasing amount of information available in electronic forms such as email, blogs, social media, and the world wide web. The presence of so much text in electronic form is a challenge to atural language processing. So, it's better to define a well-defined methodology to analyze and classify this massive data has drawn many communities' attention to this kind of data which is

* Corresponding Author: Kasahun, kasutesh1212@gmail.com
DOI:
 Manuscript received February, 2022; revised March, 2022; accepted June, 2022.

known as unstructured text. This phenomenon has made the importance of document text classification begins to mechanism up.

Text classification (TC) is an automatic mapping of text to a set of predefine categories. It has many important applications like sentiment analysis [1],sentence classification [2], and document classification [3]. Document text classification (DTC) is the most significant and a fundamental task of text classification which automatically assigns a document from a series of documents, based on its content.

A document may contain a sentence, a paragraph or a long text of words. As document contains a large vocabulary size, it has redundant information and more noise as compared to short text [3]. Similarly, multiple paragraphs within the same document semantically belong to multiple categories. Since there are two types of document classification methods which is semantic/concept-based and keyword-based [4]. This study follows concept using representing semantic document text a new concept of distributional representation of words as vectors.

Document classification is the act of labeling documents into categories according to their content. There are three machine learning methods or other technologies are used to automatically classify documents [5]. The first is Supervised machine learning which the categories are known beforehand and given in advance for each training document. The second method is the Unsupervised method, in which documents with similar words or sentences are automatically categorized together by a classifier without any prior training.

The last one is Rules-based method [6], this strategy entails utilizing a system's natural language comprehension capabilities and developing linguistic rules that tell the system to classify a document as if it were a person. Since the authors follows by labeling dataset, which is supervised learning scheme.

TC represented using single word or phrase or sentences with senses etc. For example, in text documents, the word “bank” can classify in to both health and business classes at the same time. Another example the sentence “making daily exercise is good for body strength” can be labeled as sport and health. Each instance is associated with a set of relevant labels. But the sentence has not equal probability to assign each label. The remaining labels are considered as irrelevant.

Previously many researchers have been conducted by using machine learning approaches to create models that allow automatically classify Afaan Oromo TC based on category related details [7] [8] [9]. Traditional text classification methods based on machine learning have many disadvantages such as dimension explosion, data sparsity, and limited generalization ability [10]. Generating efficient features from big corpus is difficulty, because of the high dimensionality and vast number of attributes. The other challenges of text classification include extracting the text features and training the classification models. Traditional ML-based text classifiers methods use TF-IDF, Bag of words to map some representation of the input data to predefined set of meaningful outputs but ignoring the context and internal hierarchy of the text and in addition, the traditional approach treats labels as independent individuals while ignores the relationships between them, which not reflect reality but also leads significant loss of semantic information. Therefore, DL have the ability to handle syntactic and semantic relations of words. So, developing and implementation of some efficient automatic DTC system for the Afaan Oromo language is very imperative.

The aim of this study is to explore Afaan Oromo document text classification using deep learning approaches. This is the first research of Afaan Oromo DTC utilizing a deep learning model that's are aware of. In this study, the researchers collect text documents and organize into ten classes (Agriculture, Business, Culture, Education, Health, Politics, Social, Sports, Technology and Accident). The researchers proposed one of the neural networks learning algorithm called Single-layer Multisize Filters Convolutional Neural Network (SMF-CNN) to study automatic AODTC.

2. Related Works

2.1.Machine Learning

Previously, researchers performed automatic text classification by using machine learning classifiers such as Decision Tree and SVM method for Afaan Oromo text categorization [7]. The researchers annotated few news texts based on six categories. To design these study text classification and clustering techniques were used. They have trained the system with corpus of size 824 news texts and achieved good results.

Author [8], has designed with similar techniques we mean that text classification and clustering but different algorithms such as J48, Naïve Bayes, Bayes Net, and SMO classifier algorithms were implemented for training text classification model depending on eight main classes of documents. The J48 method outperforms the other classification algorithms by 94.3755%, hence it was chosen to build the classification model.

Recently, [9] used machine learning to perform multi label classification on Afaan Oromo text. They system was trained and tested on datasets collected from OBN from 2008 to 2010 based on nine categories. The researchers compare two algorithms, Naive Bayes and KNN, and conclude that Naive Bayes is the best model for multilevel text classification. In terms of Afaan Oromo text classification/categorization, studies [7] ,and [8] have been proposed on AOTC on single label classification, whereas [9] is neutral in this class. The study aims to developing multilabel TC, specifically for Afaan Oromo news texts.

2.2.Deep Learning

Recently, deep neural networks are so popular and are widely used in lots of domains for the purpose of classification, including text classification. According to [2], he proposed for short texts with fairly balanced class distributions, and gets good results for document classification with a single layer CNN, perhaps with differently sized kernels across the filters to allow grouping of word representations at different scales. This study uses of pre-trained word vectors for classification tasks with CNN found that using pre-trained static word vectors does very well.

According to [11], neural networks outperform classical linear classifiers, particularly when used with pre-trained word embedding. The network's nonlinearity, as well as the ability to easily integrate pre-trained word embedding, frequently result in superior classification accuracy. CNNs are effective at document classification, according to the authors, because they can extract salient features (e.g., tokens or sequences of tokens) in a way that is independent of their position within the input sequences.

When tuning a convolutional neural network for a document classification problem, some hyperparameters are more important than others. Researchers[12] investigated the hyperparameters required to configure a single layer convolutional neural network for document classification. The study is motivated by their claim that the models are configuration sensitive. Their goal was to provide general settings for configuring CNNs on new text classification tasks. They depict the model architecture as well as decision points for properly configuring the model.

To efficiently classify text in the security industry, [13] developed a text classification model based on Word2vec and LSTM. To circumvent the large dimensionality of standard approaches, they developed a pre-trained word2vec. The results reveal that the model can correctly classify these patent texts 93.48% of the time. Fast-Text from Facebook was introduced by [14] with the main goal of improving the Word2Vec model. In contrast to Word2Vec, which attempts to learn vectors for individual words, the Fast-Text framework is trained to generate numerical representations of character n-grams. Because word2Vec cannot generate vectors for words that are not in the dictionary.

Researchers [15], done the in Amharic word embedding to classification purpose he assumes that as known word embedding capture different linguistic characteristics, which are intrinsic, such as word analogy, word similarity, out-of-vocabulary words and odd-word out operations. The author also uses to train classifiers using Fast-Text, a recent method to generate and evaluate word embedding was utilized. Gives benefit to capture sub word information. The results show that the model achieved a 97.8% F1-score in Multiclass text classification, with the result varying depending on the parameters. Researchers [3], designed, document-level TC for Urdu languages. They focused on investigating SMF-CNN and compare its performance with sixteen ML baseline models on three imbalanced datasets of various sizes, and the result showed that SMF-CNN achieved the good accuracy rather machine learning.

3. Materials and Methods

3.1.Data Source

Previously there is no publicly available Afaan Oromo document text dataset for classification tasks. To success this research work, data is collected from Oromia Broadcasting Network. For the purpose of this study, data is collected around 6450 text documents in year of 2010-2013 E.C.

from the mentioned sources. In NLP task's role of dataset is very important and having label data has enormous impact on research, success or failure have depended on the quality of correct data. The collected data is in the format of word files. Each document article contains its headline and description of the news. Then the collected data is converted into excel (.csv) format as suitable for preprocessing.

We used Python as a tool to design, implement and test AODT classification models. It is used for various operations like data preprocessing, embedding, and classification. The library which are used is Jupyter Notebook, TensorFlow deep learning library, Keras deep learning library, and Genism library.

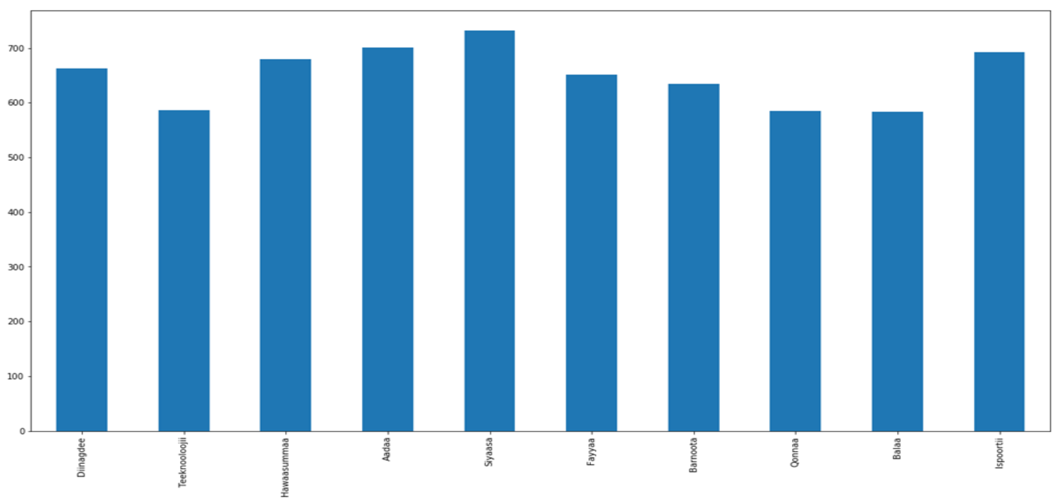

Figure 1.Description of datasets in number of documents

3.2.Pre-processing

The first stage in document classification is pre-processing. Afaan Oromo document text preprocessing includes tokenization, typo error and abbreviation correction, removing punctuation and numerals, stop word removal and different character which non-Latin words are cleaned. But some punctuation marks are reserved such as hyphen (-) and hiccup (called Hudhaa) ('), because they have highly important in Afaan Oromo languages. For example, sab-lammii, Qe'ee, such types of words are reserved. In example [*sabaa fi sab-lammiin kamillee qe'ee isaarraa buqqa'uu hin qabu*] these sentences reserved and written as [*sabaa, sab-lammiin, qe'ee, buqqa'uu*].

To avoid the unnecessary representation of a given word in different forms typo error and abbreviation is required. Because some abbreviation words cause a significant problem for the interpretation of some words (e.g., Afaan Oromoo which is AO). Here “AO” is less than the defined thresholding it will remove, abbreviation converters can help account for these exceptions. In another example “mana barumsa” abbreviated in to “MB”, which is school. After analyzing the Afaan Oromo document items, the authors found that the words whose character length is less three characters cannot be a candidate word to represent the document. So, they are ignored during the feature selection for classification purpose.

3.3.Single Layer Multi-Size Filters Convolutional Neural Network

CNN is multistage trainable Neural Networks architectures sophisticated for classification tasks and it is chosen for classification tasks like document classification and sentiment classification. It's a feed-forward neural network with convolution and pooling layers interleaved. This model consists of a series of filters of various sizes and shapes that convolve (roll over) the original document matrix to reduce it to lower-dimensional matrices.

CNN [16] has been widely used in a long text classification model because of it can be highly parallelized, the model using multiple channels, choose to use different size of the filter, and the Max pooling to select the most influential and lower latitude high-dimensional classification characteristics, and then use a dropout of full connection layer depth of text feature extracting, the final classification results. SMF-CNN was proposed by [2].

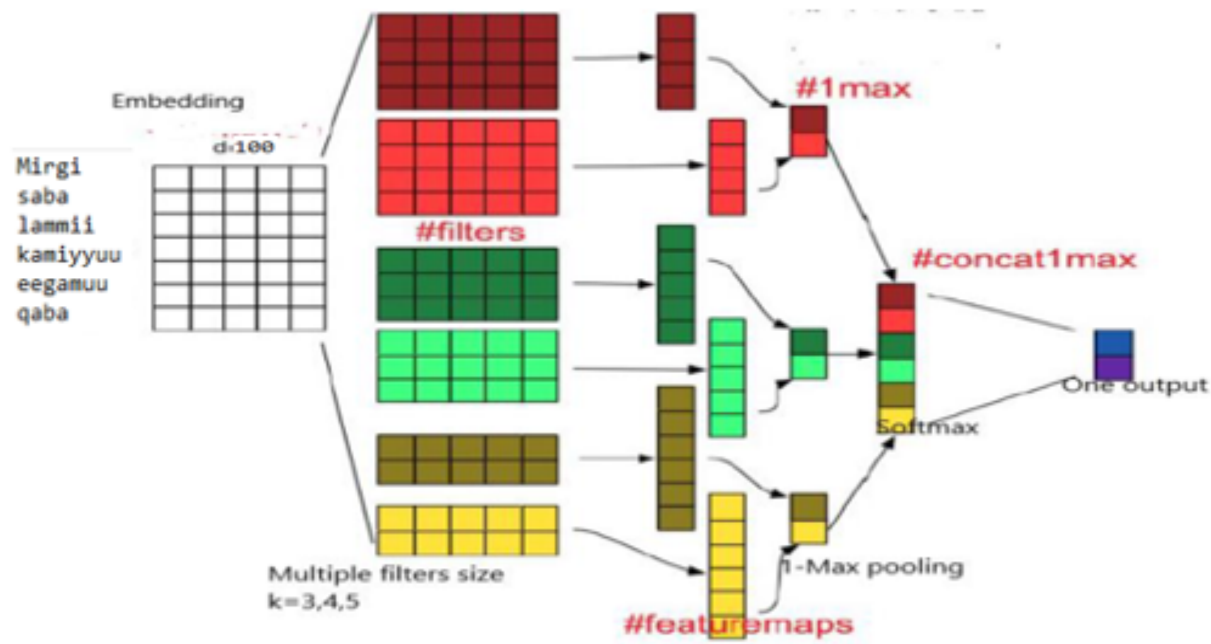

Figure 2. Single-layer Multisize Filters Convolutional Neural Network [2]

Figure 2. Illustration of a SMF-CNN architecture for document text classification. Here depict different filter region sizes (i.e., 3,4,5) each of which has two /more filters. Filters apply convolutions to the document matrix and generate (variable-length) feature maps; max pooling is applied to each map, which means that the largest number from each feature map is recorded. As a result, all maps generate a univariate feature vector, which is then concatenated to form a feature vector for the penultimate layer. This feature vector is then fed into the final Softmax, which uses it to classify the document into one of the multiple classes.

3.4.Proposed Flow work

The architecture of Afaan Oromo document text classification has the following basic components. These components are pre-processing, word embedding, SMF-CNN training, and finally the classification module.

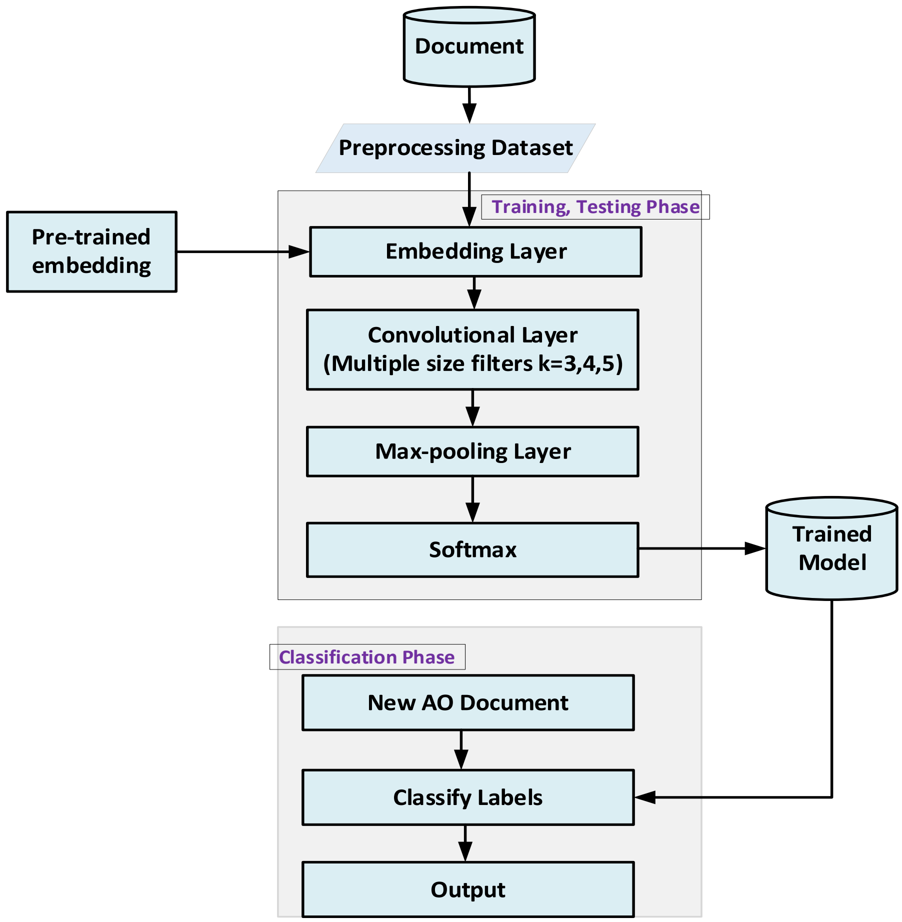

Figure 3. Proposed Flow work

The components of a documents (the words) must to be encoded before input to the models. The variability of the document length depends on the number of words in the document that need to be addressed, as models need constant input dimensionality. To achieve the maximum length of all documents in terms of dimensionality, the padding technique is used, which involves filling the document matrix with zeros. Next step, the encoded documents are converted into matrixes for which each row corresponds to a single word. The produced matrices pass through the embedding layer where each word (row) is converted into a low-dimensional representation by a dense vector.

Pre-trained word embedding: The first step is pre-trained word embedding, which is distributed representation of words as real-valued vectors learned from a text corpus using neural networks and techniques of matrix factoring. Those are low-dimensional, dense vector representations of words in a continuous embedding space where in semantically related words appear near each other and similarity represent vectors. In addition, word embedding preserves both semantic and syntactic words based on their contexts, and has proven to be very effective in NLP applications.

Word embedding got its popularity by the groundbreaking works of [14] using Fast-Text to improve the Word2Vec model. In this research authors uses Fast-Text word embedding. In general, the collection of text is from OBN, and Afaan Oromo documents which published papers for word embedding collected 68,944 words with their vectors.

Input Layer: The input layer will receive the pre-processed data. There are 61456 features in total, with a maximum sequence length of 2836 from 32759 after preprocessing. As a result, the input matrix has the following shape (61456, 2836).

Embedding Layer: Embedding Layer, for text classification problems, is a special component of the CNNs. Embedding are word representations into a type so that words with similar meanings have similar representations. Individual words in word embedding are represented as real-valued vectors with a few hundreds of dimensions. The number of words in the document that need to be addressed determines the document length variability, as CNNs require constant input dimensionality. To achieve the maximum length of all documents in terms of dimensionality, the padding technique is used, which involves filling the document matrix with zeros.

Convolution Layer: Convolutional Layers, are major components of the CNNs [17]. A convolutional layer is made up of a number of kernel matrices that perform convolution on their input and output a feature matrix to which a bias value is added. The learning procedures aim to train the kernel weights and biases as shared neuron connection weights.

The convolution layer applies filters of different width to the document embedding matrix to extract distinctive features as vector corresponding to each filter and creates a feature map. Each convolution layer is established by filters of different sizes in the result of obtaining multiple feature maps. Convolving the same filter through embedding of every word of a document extract features that are independent of word positions. Filters at higher layers capture syntactic or semantic associations between phrases that are far apart in a documents text.

The filter is applied to the window in every possible way to produce a feature map. To obtain the feature map, then add the bias term and apply activation function. The ReLU is function applied to a layer to inject non-linearity to the system by making all the negative values to zero. It helps in fast training of the model without making any significant difference in accuracy. Then the feature map is the forwarded to the pooling layer to down sample the feature map.

Max-Pooling Layer: Pooling Layers, are also integral components of the CNNs [18]. The purpose of a pooling layer is to perform dimensionality reduction of the input text features. Pooling layers make a subsampling to the output of the convolutional layer matrices combing neighboring elements. The most common pooling function is the max-pooling function, which takes the maximum value of the local neighborhoods. These max-pooled feature values are then concatenated to compose a new higher-level feature vector, which acts as input to the next layer. The major benefit of including such max-pooling layer in the network is that it reduces the number of parameters or weights, and controls overfitting.

SoftMax Layer: The model is topped by a Softmax classifier layer that predicts the probability distribution over classes. The output of the dense layer is passed to a Softmax layer. Softmax layer is used to finally classify the document into one of the multiple classes.

Output layer: By taking this process it predicts the labels document text classification, therefore the output is target text with their predicted labels.

4. Experiment and Result

The Experimental scenarios are divided in to three-part: Preprocessing data set, Word Embedding, proposed SMF-CNN implementation. The first steps are preprocessing like steps of tokenization, stop word Removal, numerals and punctuation removal are discussed. In Afaan Oromo languages all punctuation marks are not removed like another Latin languages which makes differ specific properties of these languages, because they are more important in meaning of words unless the words are loss their meanings, as a result some of them are reserved.

Output of such process consider as input of next layer. In addition, word embedding is converting text to vector form here is main work because good representation of word in distributional manner have effect on research work totally. The prepared pre-trained word embedding from different related documents have given an advantage to contain more words in embedding vocabulary.

Finally experiment done to apply proposed AODT using SMF-CNN approaches by accepting input form Embedding layer combine and classify the text documents to their classes.

Table 1.Shows accuracy performance of models on all class.

Model	Accuracy
CNN	96.40%
SMF-CNN - Fast-Text	96.81%
SMF-CNN – Word2Vec	94.85%

Table 2. Shows accuracy performance of models on six class.

Model	Accuracy
CNN	94.65%
SMF-CNN - Fast-Text	95.22%
SMF-CNN – Word2Vec	93.69%

As seen in the Table1, the SMF-CNN Fast-Text pre-trained embedding is good performance. This implies the pre-trained embedding as features for SMF-CNN have high accuracy compare with Word2vec and not CNN, these experiments are done on all text classification datasets which we have 6450 document texts with ten class. The experiment aims to test what the class categories are increase the accuracy also increase in general as compare to traditional machine learning algorithms.

Two experiments have done, first experiment is done when the number document is ten class and the second one is when the number of classes is six. As described, the ten major classes are ***(Agriculture, Business, Culture, Education, Health, Politics, Social, Sports, Technology and Accident)*** whereas the second experiment six class is ***('Business', 'Social', 'Education', 'Sports', 'Politics', 'Technology')***.

An experiment is done with the different number of class and data size and by using neural word embedding on Fast-Text pre-trained embedding is achieve good result. The results in Table 1 and Table 2 shows that neural word embedding can be used to develop AODTC. Even if a performance improvement through using word embedding from domain perspective are more important for desire result observed from this experiment, this shows that increasing class are also increase the performance. In conclusion, Table1 shows number documents 6450 with class ten and Table 2 is the number of 4422 documents with class six.

Visualizing Accuracy and Losses

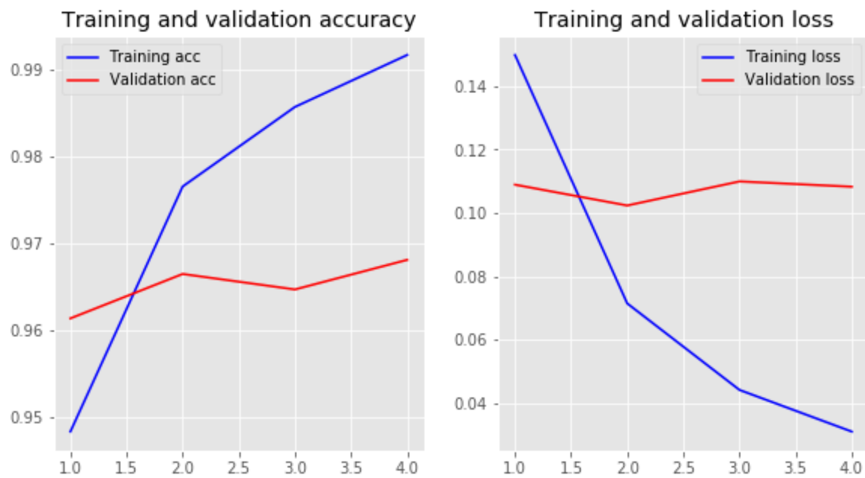

Figure 4.Training accuracy and training loss of SMF-CNN respectively

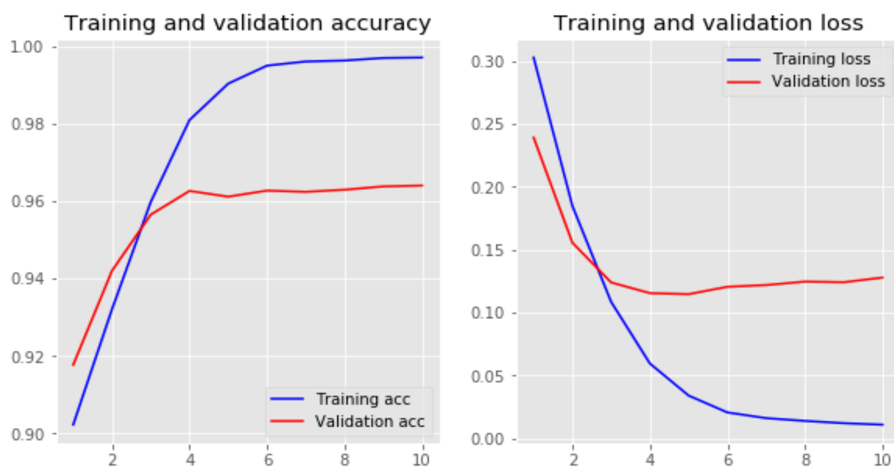

Figure 5. Training accuracy and training loss of CNN respectively.

The model is trained with 5160 files and is validated against 1290 files based on all classes. Fig 4 and Fig 5 shows the graph for training accuracy and training loss of SMF-CNN and CNN of the model respectively.

5. Discussion

In this section, the performance of the SMF-CNN architecture is developed for training and validating AODTC. In our experiments we explored document text classification since each document of our labelled dataset can be assigned to one class. With these experiments we also wanted to see what impact a small dataset like, as we explained in Table 1 and 2.

These models have a large set of hyper-parameters and finding the perfect configuration for problem domain is a challenge. Different configurations of the proposed network were explored and attempted to optimize the parameters based on the validation and training set accuracy. From the collected dataset 20% for testing and 80% for training the proposed SMF-CNN architecture discussed earlier was used the training with around 5160 samples for training, 1290 for validation datasets. Then trained each classifier on each, and fine-tuned the hyper-parameters with the validation set. Once we found the best hyper-parameters, we used the trained classifier to predict the class from the test set.

The experiment is with various parameter values such as the number of epochs, batch size, optimizer selection, and network layers until found the best fit model. In Table 1 and Table 2 the results of experiments are presented, based on experimental analysis. However, by adjusting the network's other parameters, we able to get good performance from the chosen SMF-CNN models. The batch size and optimizer selection are the other network parameter and changed during our experimental analysis. Optimizers are used to update the network's weight and have their own set of behaviors. For our proposed model the Adam optimizer gives better results compared with other optimizers.

After hyperparameter tuning of our model, the performance of SMF-CNN evaluated using in different ways: Fast-Text pre-trained and Word2vec pre-trained word embedding, the other is without using pre-trained embedding. The study demonstrates how to use SMF-CNN to build Afaan Oromo document text classification by employing Fast-Text pretrained word embedding. These studies look into multi-class text classification. As a result, we developed some of word embedding, which any researcher might use to complete further Afaan Oromo NLP tasks like sentiment classification for Afaan Oromo language.

6. Conclusion

We analyzed proposed models for Afaan Oromo document text classification in this study using SMF-CNN. The SMF-CNN models produce optimal features to represent texts by considering their contextual information, which is later used to analyze unlabeled documents. We have compared our proposed models in terms of specified parameter settings against one of the existing models, CNN, and found that the proposed models perform better in terms of accuracy. The proposed SMF-CNN models could help classify Afaan Oromo document text.

Researchers for developing Afaan Oromo document classifiers are highly concentrated on improving the classification accuracy as the number of categories increased by using different learning algorithms. They tried to solve the problem of computation time using machine learning algorithms statically using a different method.

The experimental analysis is conducted over 6450 AODT categories. We used Python as an experimental environment for word embedding and document classification. The classification accuracy is improved by comparing non-pre-trained word embedding in increased classes. On the AODTC task, the SMF-CNN shows good classification performance on our datasets. Finding the optimized parameters of the SMF-CNN is a time and resource-exhausting process, but it improves the classifier's performance.

Generally, our model can classify documents with an accuracy of 96.81% on the OBN document text dataset. The model can classify documents based on ten classes (Agriculture, Business, Culture, Education, Health, Politics, Social, Sports, Technology, and Accident). The CNN model was less accurate than the proposed model when tested experimentally. Using Fast-Text word embedding along with Single Layer Multisize Filter Convolutional Neural Network helped maintain the relationships between the words. The model has been improved using pre-trained word embedding.

7. Acknowledgments

First of all, I would like to express a deep gratitude to my staff, who is always there during the work of this paper in giving me the support, courage and advice. Next, I would like to thank Oromia Media Network for giving the data and Mettu university for sponsored.

References

[1] Mehmood, K., Essam, D., & Shafi, K, "Sentiment Analysis System for Roman Urdu," *Proceedings of the 2018 Computing Conference ,* vol. 1, no. 10, pp. 29-42, 2019.

[2] Y. Kim, "Convolutional Neural Networks for Sentence Classification.," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* 2014.

[3] Akhter, M., Jiangbin, Z.,Naqvi, Syed I., Abdelmajeed, M., Mehmood, A & Sadiq, M., "Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network," *IEEE Access,* vol. 1, no. 10, p. 99, 2020.

[4] Kumilachew T., Tarekegn, A & Alemu, T , "A Comparative Study of Flat and Hierarchical Classification for Amharic News Text Using SVM," *International Journal of Information Engineering and Electronic Business.,* vol. 9, pp. 36-37, 2017.

[5] F. pascual, "monkeylearn," 21 Nov 2019. [Online]. Available: https://monkeylearn.com/blog/document-classification/#:~:text=Document classificatio,. [Accessed 17 2 2020].

[6] Expert.ai, "Expert.ai Team," 20 September 2016. [Online]. Available: https://www.expert.ai/blog/document-classification-works/. [Accessed 19 2 2021].

[7] Jimalo, K., P, Ramesh & Assabie, Y. , "Afaan Oromo News Text Categorization using Decision Tree Classifier and Support Vector Machine: A Machine Learning Approach.," *International Journal of Computer Trends and Technology. ,* no. 47, 2017.

[8] Naol, B and Mamo.G, "A Two Steps Approach for Afaan Oromo Nonfiction Text Categorization," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 10, no. 4, p. 120, 2018.

[9] Kannaiya R., Naol.B and Barani.S, "Multi-Label Classification for Afaan Oromo Text by using Python Machine Learning Tools," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 3, no. 1, 2020.

[10] Hongping W., Yuling L., Jingwen W, "Review of Text Classification Methods on Deep Learning," *CMC, vol.63, no.3,,* vol. 63, pp. 1309-1321, 2020.

[11] Y. Goldberg, " A Primer on Neural Network Models for Natural Language Processing.," *Journal of Artificial Intelligence Research.,* vol. 57, 2015..

[12] Zhang, Y & Wallace, B., "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.," *Retrieved: arXiv:1510.03820v4 [cs.CL],* 2016.

[13] Xiao, L., Wang, G & Zuo, Y., "Research on Patent Text Classification Based on Word2Vec and LSTM.," *Research on Patent Text Classification Based on Word2Vec and LSTM.,* 2018.

[14] Joulin, A., Grave, Ed., Bojanowski, P & Mikolov, T., "Bag of Tricks for Efficient Text Classification.," *Computation and Language:,* vol. 3, 2016.

[15] Y. Yared, "EXPLORING NEURAL WORD EMBEDDINGS FOR AMHARIC LANGUAGE," *NEAR EAST UNIVERSITY,* 2019.

[16] Liu, Z., Huang, H., Lu, C., & Lyu, S, "Multichannel CNN with Attention for Text Classification.," 2020.

[17] J. Brownlee, "How Do Convolutional Layers Work in Deep Learning Neural Networks," Machine Learning Mastery, 17 April 2020. [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. [Accessed 22 11 2020].

[18] J. Brownlee, "A Gentle Introduction to Pooling Layers for Convolutional Neural Networks," Machine Learning Mystery, 22 April 2019. [Online]. Available: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/. [Accessed 22 11 2020].