Authorship Identification using Deep Learning

Suraj Panwar
Mtech DESE
SR No : 14644
surajpanwar@iisc.ac.in

Abstract

Authorship identification allows us to identify the most likely authors of documents, news or articles. It can also be applied to identification of ghost authors and many other applications, which makes it a important field of Natural Text Processing. In this project we tackle this task using both linear classifiers such as SVM, Bag of Words classifier, as well as Neural Network based classifiers such as LSTM, CNN and MLP based classifier's with different learned embeddings.

1 Introduction

Every person has a unique writing style in his works. Some features which bleeds into each of their works irrespective of the topics. Some of these features can be identified by the reader, however most writing styles have such trivial variations that the user is not able to distinguish between these features, due to this readers are unable to identify the authors of the document. This can be a big problem specially nowadays with the advent of social media where snippets of documents make their way to the masses without any author credibility leading to both plagiarism as well as creating hoax.

The problem that we wish to undertake in this project is to identify the author from a group of candidates based on their writings as well as generate a reliable parameter to compare the writing styles of different authors for building a recommendation system for users to select authors based on their favorite writing styles.

2 Prior Work

Authorship Identification problem has been tackled from multiple frontiers. In (Koppel et al., 2007) the authors proposed a unmasking method for the identification which gives satisfactory results. Support Vector machines (SVM) classifier is used in this attribute questioned application. This method even though valid for long documents, does not gives good results on small documents. In (Aida-zade and Talibov, 2016) the author has proposed a n- gram model for Author Identification. In (Hagara and Pucik, 2013) the authors have conducted a feasibility study on a large on line scale using Nearest Neighbor Classifier.

Work in the field of Deep Learning has also been carried out by (Mohsen et al., 2016) where they have used Stacked Noise Auto-Encoders for the classification problem on RCV1 dataset.

3 Identified Datasets

In our work, two different datasets will be used. Reuters 50 dataset offers 5000 news articles by different 50 different authors. Proper preprocessing(i.e. proper tokenization and noise removal) will produce a fine dataset for our work. The second dataset, that is already preprocessed and ready to use is the one provided by Kaggle through its Spooky Authorship Detection competition. The dataset is a collection of works by 3 authors namely Edgar Allan Poe (EAP), HP Lovecraft (HPL) and Mary Wollstonecraft Shelly (MWS). It is already divided into training and test dataset where the training dataset has 19579 and test dataset has 8392 complete lines extracted from books by the above mentioned three authors, all tagged.

The analysis on both the datasets will also give us a measure of extent of the application of these methods on large as well as small training bases, with Reuters providing us with a idea of model performance on relatively less data with but a large number of candidate authors which presents

a more real depiction of the real world, and spooky author detection where the data is relatively large but with low possible candidate authors to choose from.

3.1 Reuters 50 Dataset

The Reuters 50 50 (C50) dataset is a subset of RCV1. RCV1 (Reuters Corpus Volume I) an archive of over 800,000 manually categorized newswire stories recently made available by Reuters, Ltd. for research purposes.These corpus has already been used in author identification experiments. In the top 50 authors (with respect to total size of articles) were selected. The training corpus consists of 2,500 texts (50 per author) and the test corpus includes other 2,500 texts (50 per author) non-overlapping with the training texts.In our work, we re-organize these 5000 texts into non overlapping 7:2 ratio as train set to test set.

3.2 EMH Dataset

EMH dataset is a three class classification dataset which includes snippets of works of three authors which write in horror/mystery genre. The dataset has almost 20000 excerpts of writing of the three authors. The dataset is divided into 7:2 ratio for train vs test data.

4 Baseline Performance

The model baseline was decided upon a Bag of Words model which was made using the training data and with no discounting or normalization. The model baseline metrics computed were :

- RCV1 : 11.21 %
- EMH : 40.31 %

The metric performance might seem higher for the EMH dataset but the fact remains that the EMH algorithm is a 3 class problem compared to a 50 class Reuters classification problem due to which the baseline performance metric is higher for the EMH algorithm.

5 Preprocessing of Data

The data available from the official Reuters is a collection of individual files of different authors which are available as separate documents. The first task undertaken during the preprocessing of the data was to combine all the documents to create a general purpose corpus which was correctly labeled and train, test split was performed on the dataset.

The program mostly uses the Glove word vectors of 50 length as pre trained word embeddings. The GloVe vectors were used to initialize the word embeddings and the gradient are propagated to the embeddings.

Since we used Glove 50 embedding as a look-up table to represent our words, there could exist rarely-seen words which cannot be represented. Hence, we eliminated the occurrences of numbers and special characters to match the feature of our word representations. During the process of parsing, we trimmed each word to ensure that it does not contain any number or special character at both of its ends.

6 Implemented Models

In this section, we elaborate the models we used in our experiment. In total following models are implemented:

- Naive Bayes based Bag of words implementation
- Doc2Vec embedding based kernel based SVM model
- Document level Glove Embedding based LSTM model
- Document level Glove Embedding based CNN model
- Document level Doc2Vec based CNN model
- Document level Glove Embedding based MLP model
- Document level model based on Fast Text

All of the above models were implemented for both of the datasets and the corresponding observations are discussed.

6.1 Bag of Words

Naive Bayes based bag of words model was implemented for both the corpus and was used as a baseline model for the project. The model implemented uses only word frequency for each class for computation of the posterior probability and the results obtained are mentioned in the above section.

6.2 Kernelized Support Vector Machine

A support vector machine model was implemented using the Doc2Vec embeddings received for the training documents, which was passed along with the corresponding author title to train the model, the model received the following accuracy on the given datasets:

- RCV1 : 40.51 %
- EMH : 48.91 %

The performance observed for the SVM approach gives weak results and hence it's suitability for this task is questionable.

6.3 LSTM Model

In LSTM based model, the documents are represented as a average of the Glove embedding of the content words. The embeddings are propagated through a BiLSTM network with the following configuration:

Layer (type)	Output Shape	Param #
input_13 (InputLayer)	(None, 50, 1)	0
bidirectional_10 (Bidirectio	(None, 50, 400)	323200
dropout_25 (Dropout)	(None, 50, 400)	0
flatten_10 (Flatten)	(None, 20000)	0
dense_31 (Dense)	(None, 48)	960048

Figure 1: Layer configuration for LSTM network.

The network consists of a input layer of size (50,1) which hold the input Glove embedding of the incoming document. That is followed by a Bidirectional layer of size (50, 400) along with a Dropout layer and a Flatten layer of 20000 length this is then connected to the Dense layer of size (48, 1) which is then used to calculate the output by the means of softmax layer and then error back propagation using class cross entropy error. This gives us a output from a class of 48 authors.

The training and test accuracy was computed for different size of the BiLSTM layer and the results obtained are as follows:

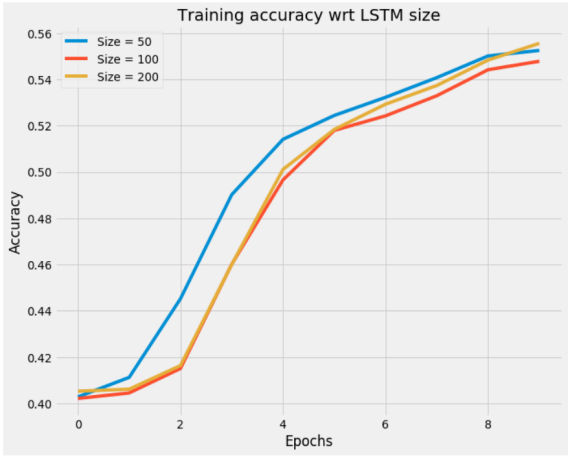

Figure 2: LSTM train accuracy for EMH Dataset.

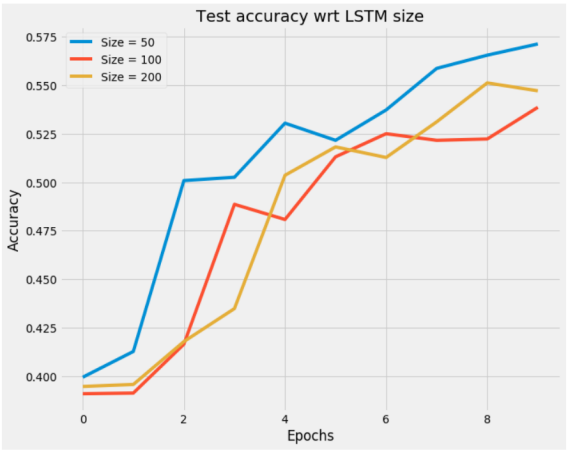

Figure 3: LSTM test accuracy for EMH Dataset.

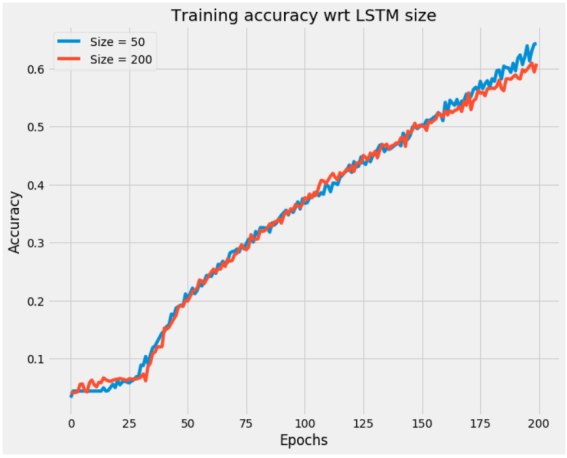

Figure 4: LSTM training accuracy for Reuters Dataset.

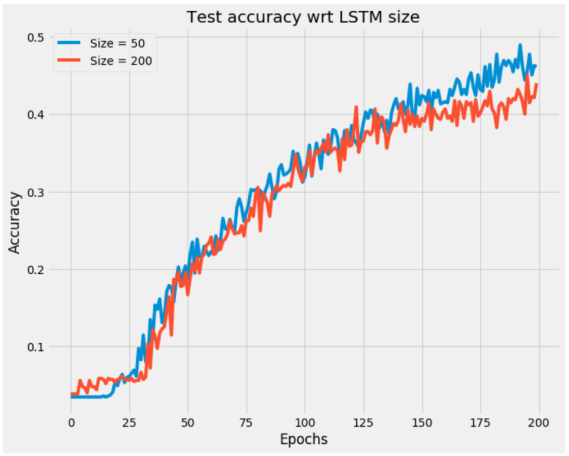

Figure 5: LSTM test accuracy for Reuters Dataset.

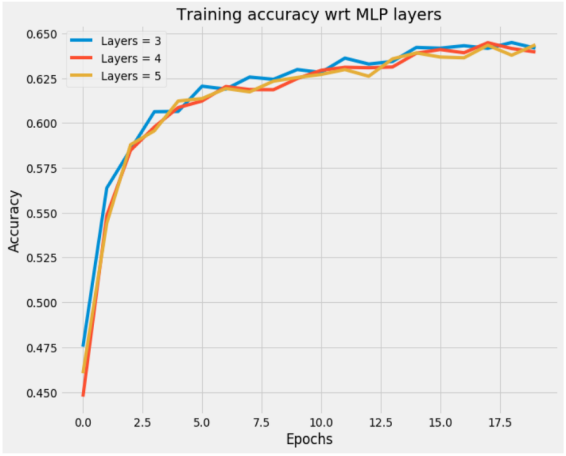

Figure 7: MLP based model train accuracy for EMH Dataset.

6.4 MLP based Model

In this MLP based model, the documents are represented as a average of the Glove embedding of the content words. The embeddings are propagated through a variable dense layer network with the following configuration:

Layer (type)	Output Shape	Param #
dense_22 (Dense)	(None, 512)	26112
dropout_17 (Dropout)	(None, 512)	0
dense_23 (Dense)	(None, 512)	262656
dropout_18 (Dropout)	(None, 512)	0
dense_24 (Dense)	(None, 512)	262656
dropout_19 (Dropout)	(None, 512)	0
dense_25 (Dense)	(None, 512)	262656
dropout_20 (Dropout)	(None, 512)	0
dense_26 (Dense)	(None, 512)	262656
dropout_21 (Dropout)	(None, 512)	0
dense_27 (Dense)	(None, 48)	24624

Figure 6: LSTM test accuracy for Reuters Dataset.

The MLP network employs multiple layers of densely connected layers with a output softmax layer of size 48 which is optimized using a class cross entropy error for backpropagation. The number of the layers was varied and the corresponding results were as follows:

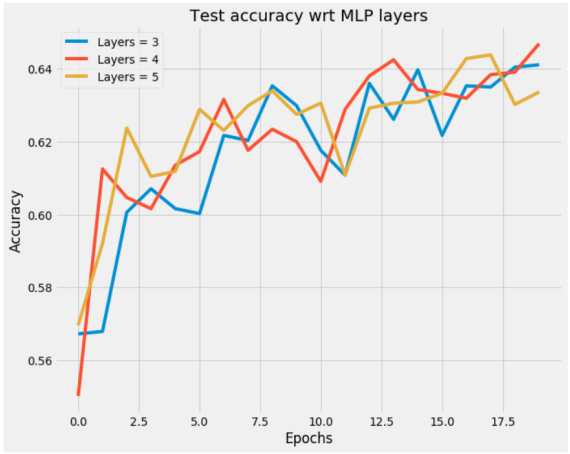

Figure 8: MLP based model test accuracy for EMH Dataset.

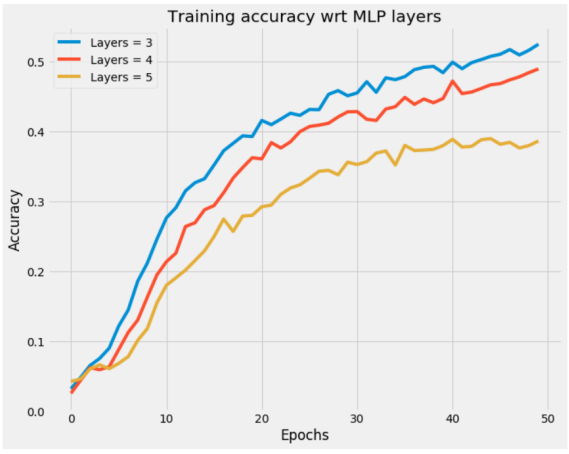

Figure 9: MLP based model training accuracy for Reuters Dataset.

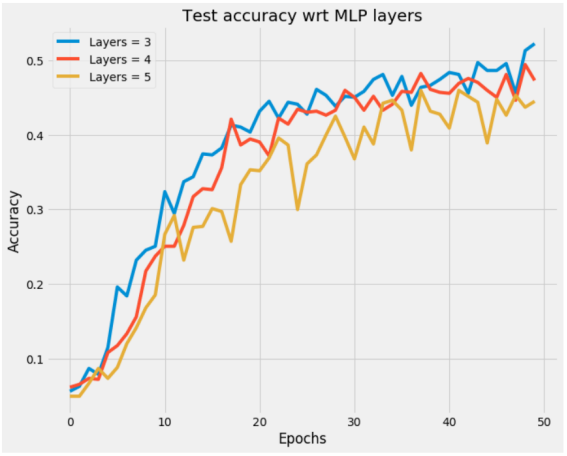

Figure 10: MLP based model test accuracy for Reuters Dataset.

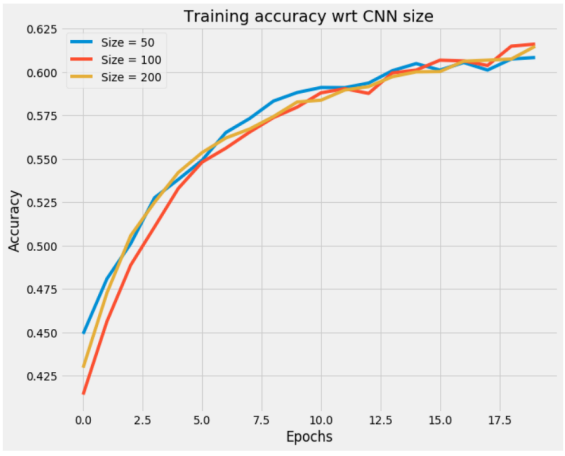

Figure 12: CNN based model train accuracy for EMH Dataset.

6.5 CNN based Model

In this CNN based model, the documents are represented as a average of the Glove embedding of the content words. The embeddings are propagated through a combination of convolution layers along with pooling layers combined with dense layers for calculating the output class. The configuration for this network is as follows:

Layer (type)	Output Shape	Param #
input_7 (InputLayer)	(None, 50, 1)	0
conv1d_5 (Conv1D)	(None, 50, 200)	800
max_pooling1d_3 (MaxPooling1	(None, 25, 200)	0
conv1d_6 (Conv1D)	(None, 25, 200)	120200
global_max_pooling1d_3 (Glob	(None, 200)	0
dense_10 (Dense)	(None, 48)	9648

Figure 11: LSTM test accuracy for Reuters Dataset.

The model consists of a input layer of (50,1) dimension for accepting the Glove embeddings of size 50, that is followed by a convolution layer consisting of 200 filters which is then followed by a max pooling layer, this configuration is repeated again and finally a dense connected layer consisting of 48 nodes is used as softmax layer for computation of the output class. The results obtained are as follows:

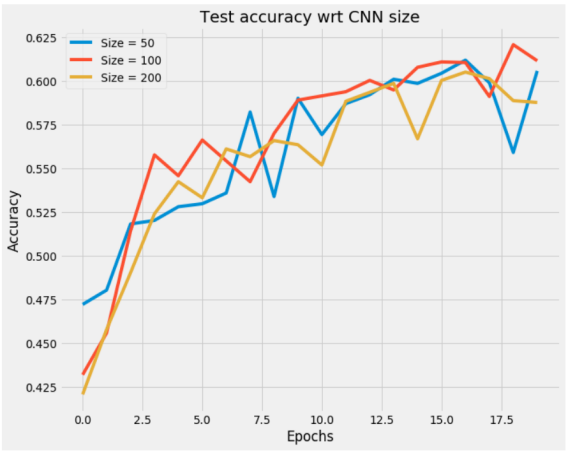

Figure 13: CNN based model test accuracy for EMH Dataset.

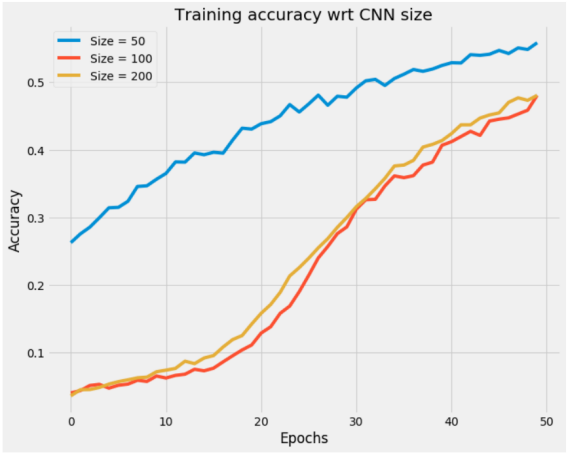

Figure 14: CNN based model training accuracy for Reuters Dataset.

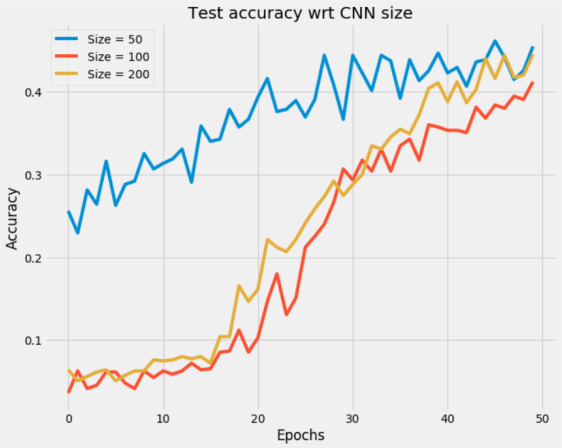

Figure 15: CNN based model test accuracy for Reuters Dataset.

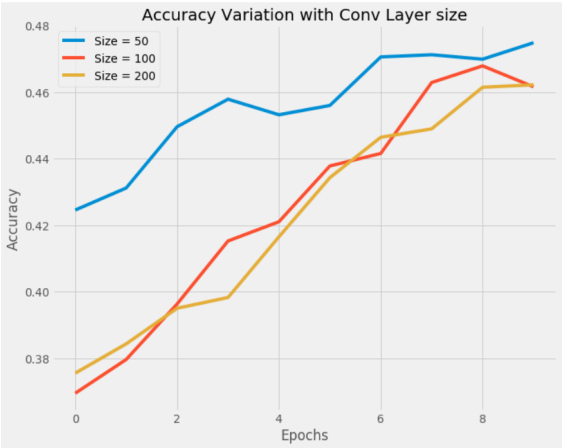

Figure 16: CNN based model accuracy with variable filters for EMH.

6.6 Doc2Vec embedding based CNN model

In this CNN based model, the documents are represented using a Doc2Vec neural representation. The embeddings are propagated through a combination of convolution layers along with pooling layers combined with dense layers for calculating the output class. The configuration for this network is as follows:

- Input layer of size (50,1) for embedding
- 2D convolutional layers of filter size (2,2) with variable number of filters.
- Corresponding max pooling layers.
- Dense layer with size equal to number of classes for softmax estimation.

For this configuration the number of filter of the convolution layer was varied along with the size of the embedding used. It was generally found that the length of embedding has not much effect on the output, if the corpus on which it is trained is of relatively small size. The results obtained were:

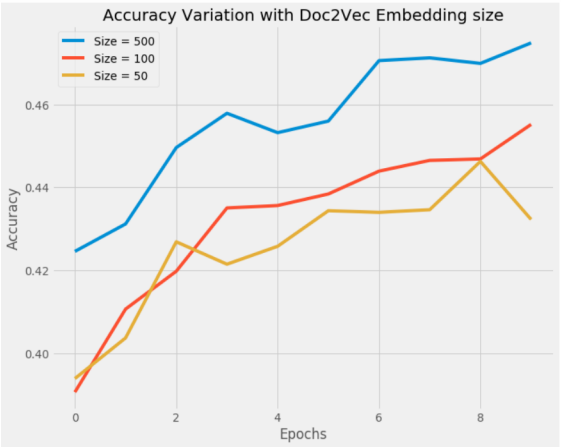

Figure 17: CNN based model accuracy with variable embedding size for EMH.

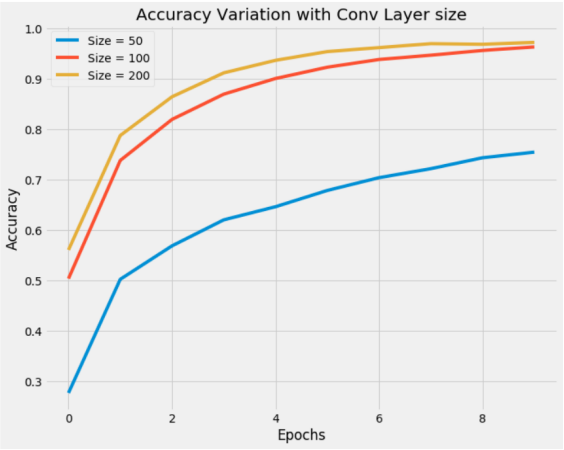

Figure 18: CNN based model accuracy with variable filters for Reuters.

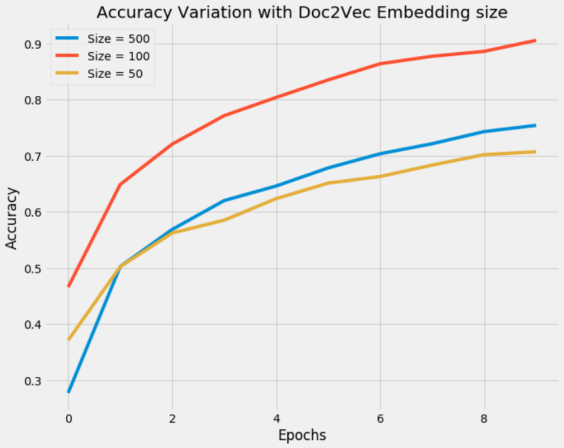

Figure 19: CNN based model accuracy with variable embedding size for Reuters.

6.7 Fast text

Fast Text is a simple linear model with rank constraint (Joulin et al., 2016). The first weight matrix A is a look-up table over the words. The word representations are then averaged into a text representation, which is in turn fed to a linear classifier. The text representation is an hidden variable which can potentially be reused. This architecture is similar to the cbow model of Mikolov et al. (2013), where the middle word is replaced by a label. We use the softmax function f to compute the probability distribution over the predefined class, the configuration is shown below:

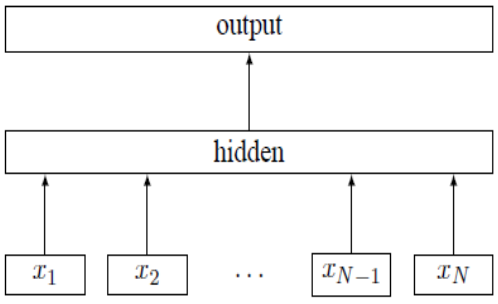

Figure 1: Model architecture of `fastText` for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.

Figure 20: Fast Text Model.

The performance of the model was computed over different embedding sizes for both datasets and the performance is shown as:

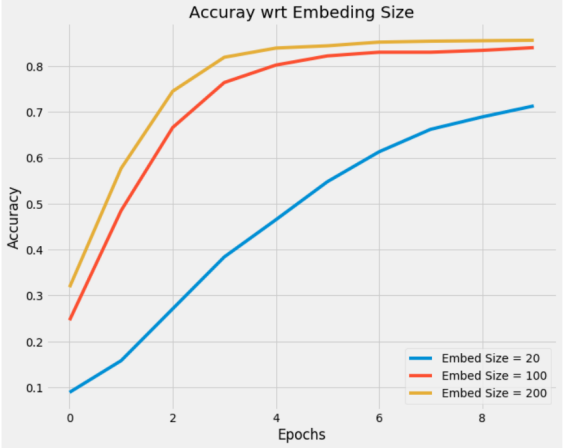

Figure 21: Fast Text Model performance on Reuters.

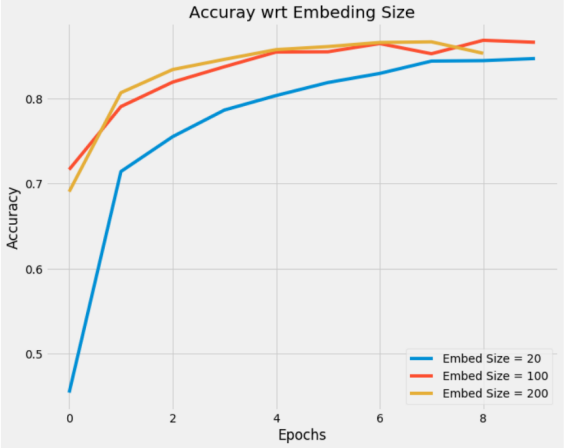

Figure 22: Fast Text Model performance on EAP.

7 Results

The performance observed are as followed:

Performance of LSTM model:

- RCV1 : 59.81 %
- EMH : 57.15 %

Performance of MLP model:

- RCV1 : 57.12 %
- EMH : 65.31 %

Performance of CNN model:

- RCV1 : 67.51 %
- EMH : 47.87 %

Performance of CNN model with Doc2Vec:

- RCV1 : 71.00 %
- EMH : 47.91 %

Performance of Fast Text model:

- RCV1 : 85.41 %
- EMH : 86.19 %

7.1 Limitation

In our Document level model, our strategy of representing a sentence is averaging the word vector, which ignores the temporal relationship among words, and also the inside structure of a sentence which severely limits the model performance. This can be further improved by using some means of establishing relations between documents which could result in model performance increment.

7.2 Future Work

Many future works could be done, such as trying different size of word embeddings, exploiting intra-article attention, and methods of representing an article or sentence instead of simply averaging. In addition, other model such as Siemese network is likely to win over our models on this authorship attribution task, which could be examined. Furthermore, an ensemble of model is promising to give a rise to the accuracy of authorship attribution. Beyond authorship identification, there are also a great amount of meaningful work worth studying. Recalling our proposed models are capable of extracting style features of a given article, further study might depend on the extracted features to mine more information under the article, e.g., the relationship between two financial reports, or the sentiment change under one persons tweets in a short period.

7.3 Conclusion

In this project, we studied different deep learning models on authorship identification. we designed models for authorship identification. Document level Fast text implementation performed best on authorship identification, outputting an accuracy of 85.4% on EMH dataset and 86.1% on Reuters dataset.

References

K. R. Aida-zade and S. Q. Talibov. 2016. In 2016 ieee 10th international conference on application of information and communication technologies (aict). In *Authorship identification of the azerbaijani texts using n-grams*. page pages 13.

M. Hagara and J. Pucik. 2013. In radioelektronika (radioelektronika), 2013 23rd international conference. In *Fingertip detection for virtual keyboard based on camera*. pages 356–360.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. In *Bag of Tricks for Efficient Text Classification*. pages 1–4.

M. Koppel, J. Schlerand, and E. Bonchek-Dokow. 2007. In measuring differentiability. In *Unmasking, pseudonymous authors*.

A. M. Mohsen, N. M. El-Makky, and N. Ghanem. 2016. In 2016 15th ieee international conference on machine learning and applications (icmla). In *Author identification using deep learning*. page 898 903.