Deep Learning Authorship Attribution

Derrick Xiong, John Schulz

Abstract

Authorship attribution is the process of identifying the author of a given text and from the machine learning perspective, it can be seen as a classification problem.In this report, we outlined the results we obtained for the W266 final project. The project was based on using deep learning models such as Convolutional Neural network, recurrent neural networks model with LSTM(long short-term memory cell). We used this state-of-the-art model for the problem of Victorian authorship attribution. The problem of attributing authorship has been a well-known problem in NLP with a large amount of human benchmarks available.

We used supervised learning algorithms as our baseline and planned to use models better suited for NLP such as LSTM, CNN. The results are very promising and we provided a framework for any similar problems regarding authorship attribution.

1 Introduction

Authorship attribution is not a new trend in human history, nor is it a problem only of recent years. With the spread of literacy, questions of authorship attribution have been raised as early as at the beginning of the last millennia. However, the sense of urgency and demand for high-end technology in attributing authorship has only been increasing because of the real-world applications such as detecting plagiarism, forensics, forged letters for inheritance, information security, and etc. Yet the problem of attributing authorship has remained unsolved, because of difficulty to assess and define the features that compose a writing style of any particular author, and the problem becomes more complicated by the colossal number of authors, each having the same or different features and components of a writing style. With the rise of deep learning techniques and computational power, we can now leverage deep learning to help solve this problem. Interest in the topic is also due to a growth in the volume of text data. Thus, automatic identification of authorship is a growing area of research, which is also important in the fields of forensic science and marketing. The reason why we chose the context of Victorian era literature is that most of the authorship attribution projects center around letters, modern documents and forensic records, and thus we decided to take on a different approach and try to build models on classic literature. Literature works are generally very well written and have meanings or connotations beyond the literal words. The dataset we are interested in is gathered from Victorian era literature work and naturally, a lot of implications and euphemisms are very difficult even for humans to understand. And since our goal is to classify the authorship of these works, it will be important and very difficult for the algorithms to capture the subtitles within the texts and detect the specific styles present in the texts. The complexity in these texts will prove to be the most challenging part of the project.In this project, our goal is to start with the baseline Neural Network model to evaluate the effectiveness of the model in solving the problem of attributing authorship to a number of Victorian era literature works. After applying the baseline model, we will use a more advanced model with a long short-term memory cell (LSTM) and ultimately will use a Convolutional Neural Network to try to achieve better results and gain more insights in using deep learning models for the authorship attribution problem. The article includes the analysis of the dataset we used, an overview of the related works and models we proposed to use, a detailed description of approaches to solving the authorship identification problem, and a discussion of the results obtained.

2 Project Overview

2.1 Datasets used

The dataset was gathered from Victorian era literature. To make sure that the dataset is not biased and reliable, the following criteria have been chosen to filter out authors: English language writing authors, authors that have enough books available (at least 5), 19th century authors. 50 authors were selected with criteria and their works were queried through Big Query Gdelt database. Then the dataset was cleaned to deal with the OCR reading problems present in the original texts. First and foremost all the selected texts were scanned to obtain the total number of unique words and their respective frequencies. During the process of scanning the texts, certain words such as the name of the author, the name of the book and other work specific characteristics needed to be taken out because they would make the classification too easy and almost trivial. Thus the second step is to remove the first 500 words and the last 500 words so as to keep the main body and meat of the work. The third step is to identify the words that each author likes to use. The top 10,000 words that occurred in each author's writings were selected while the rest of the sentence structure remained intact. Thus the entire book was split into text fragments with a length of 1000. Certain shorter text segments with less than 1000 words were filled with zeros to keep those entries in the dataset. Each instance in the training set contains 1000 words which makes up approximately 2 pages of writing, which is sufficient for us to extract features and run models on. Each instance in the training set contains the preprocessed text as well as an author ID while the test set contains only the texts.

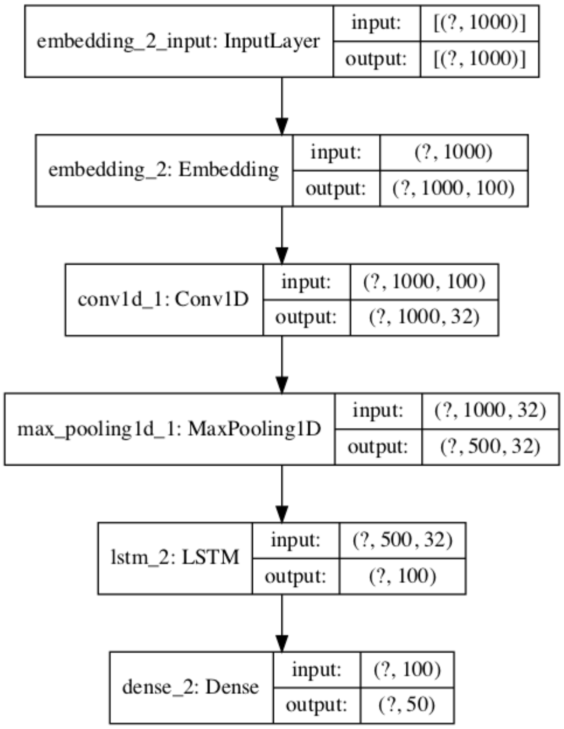
To further process the dataset and ultimately improve performance in our model, we removed stopwords and lemmatised the text for each element in our dataset. Stopwords were removed using the NLTK corpus library and resulted in text without common words that don't provide any context to the provided text or the author who wrote it. Moreover, more noise was removed from our dataset by reducing words with the same root but different conjugations to their stem or base form. We deliberately chose to lemmatise our text instead of stemming it because lemmatization considers the context of the word and reduces it to its lemma or most meaningful base form. All together, these data processing procedures helped reduce noise in our system and improve accuracy for predicting authorship.

2.2 Baseline Model

Before building a deep learning model, we constructed a baseline model using the bag of words and a multi class Gaussian Naive Bayes classifier. Gaussian Naive Bayes was selected for our baseline model for its ability to calculate the probability of each author being the writer of each text in the dataset. The bag of words was constructed by using the built in CountVectorizer function to effectively build a vocabulary for our dataset and tokenize each portion of text. The vectorizer was only fitted on the train data to ensure that no information in the test data set would be used to generate our bag of words. This procedure was completed on both the raw text and the cleaned text elements with no stopwords and lammentization applied. These two models constructed with both forms of the dataset provided a baseline accuracy score for us to build upon in later models. The accuracy of the raw data was .98 on the train dataset and .41 on the test dataset while the accuracy of the cleaned data set was .96 on the train dataset and .44 on the test dataset. These results suggest that the model was overfitting the train dataset and additionally support our assumption that the cleaned data would likely generate better results in later models.

2.3 Better models

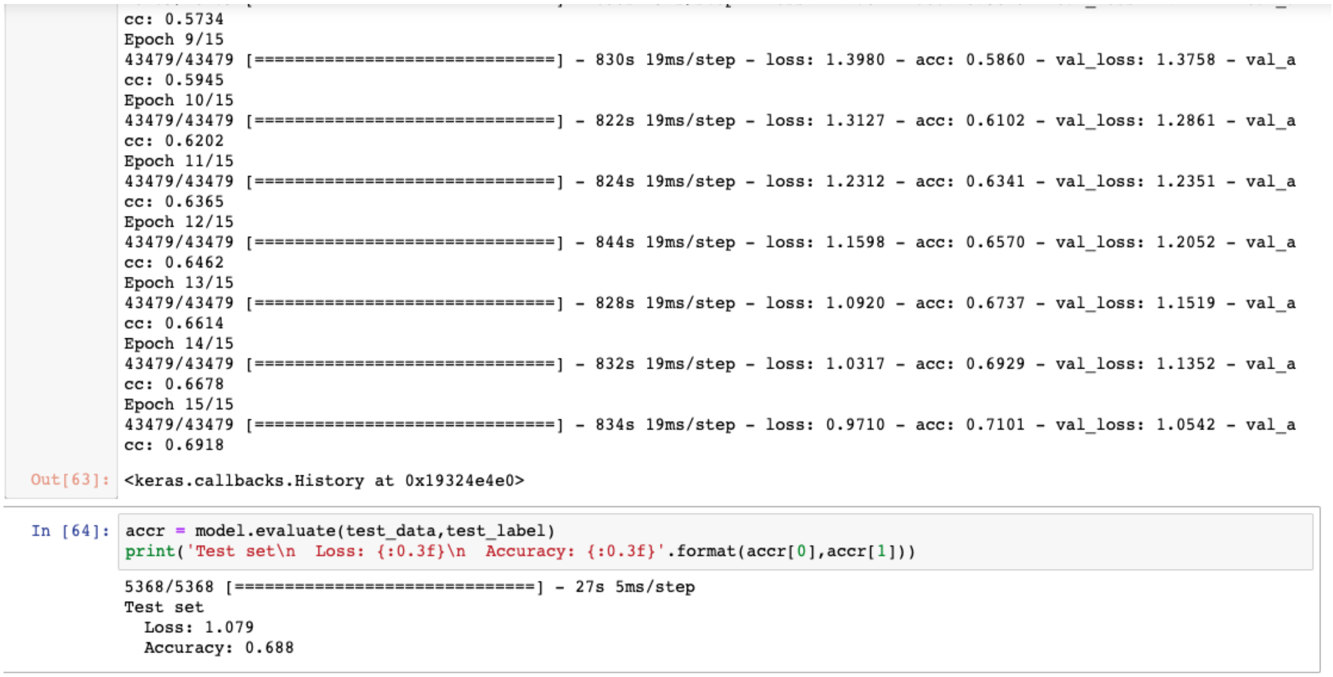
Using the datasets of the 50 Victorian authors, we then built a model of LSTM RNN. Our first step is to represent the words in each text and we do this by translating texts into real valued vectors. We planned to use the keras Tokenizer to map words into integers. The number of top words in the Tokenizer is a hyperparameter we intended to fine tune as the development goes on. And then we use pad_sequences function from keras to make sure all the entries in the datasets are of the same length. We chose a train test split ratio of 0.1. Our original idea was to make two separate models with CNN and LSTM but after trying out both models, we decided to combine both models to achieve maximum results. We would place a convolutional layer and max pooling layer before the LSTM to reduce dimensions and extract features.



- The first layer is the embedding layer and we initially decided to use length 100 to represent the words.
- The next layer is the 1-dimensional convolutional layer with 32 filters and kernel size 5.
- The third layer is a max-pooling layer with pool size 2.
- The fourth layer is a LSTM layer with 100 memory units and dropout rate of 0.2 and recurrent dropout rate of 0.2.
- The output layer is a fully connected layer with 50 output values because we have 50 authors we try to classify.
- For the loss function, we chose categorical_crossentropy with the adam optimizer.

3 Results

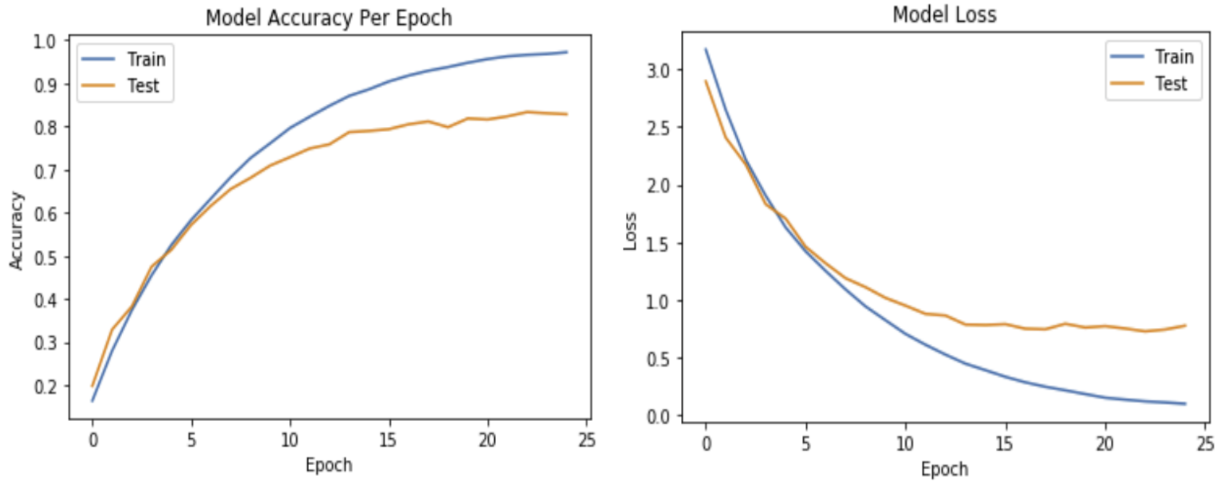
We experimented with different hyperparameters such as maximum number of words in Tokenizer and embedding dimension in the embedding layer. The preliminary results we got without removing stop words and stemming is very underwhelming with validation accuracy around 0.25 and testing accuracy of 0.23. Then we implemented the removal of stop words and stemming, and increased the number of eochs to 15. The results were much more promising, shown follows:



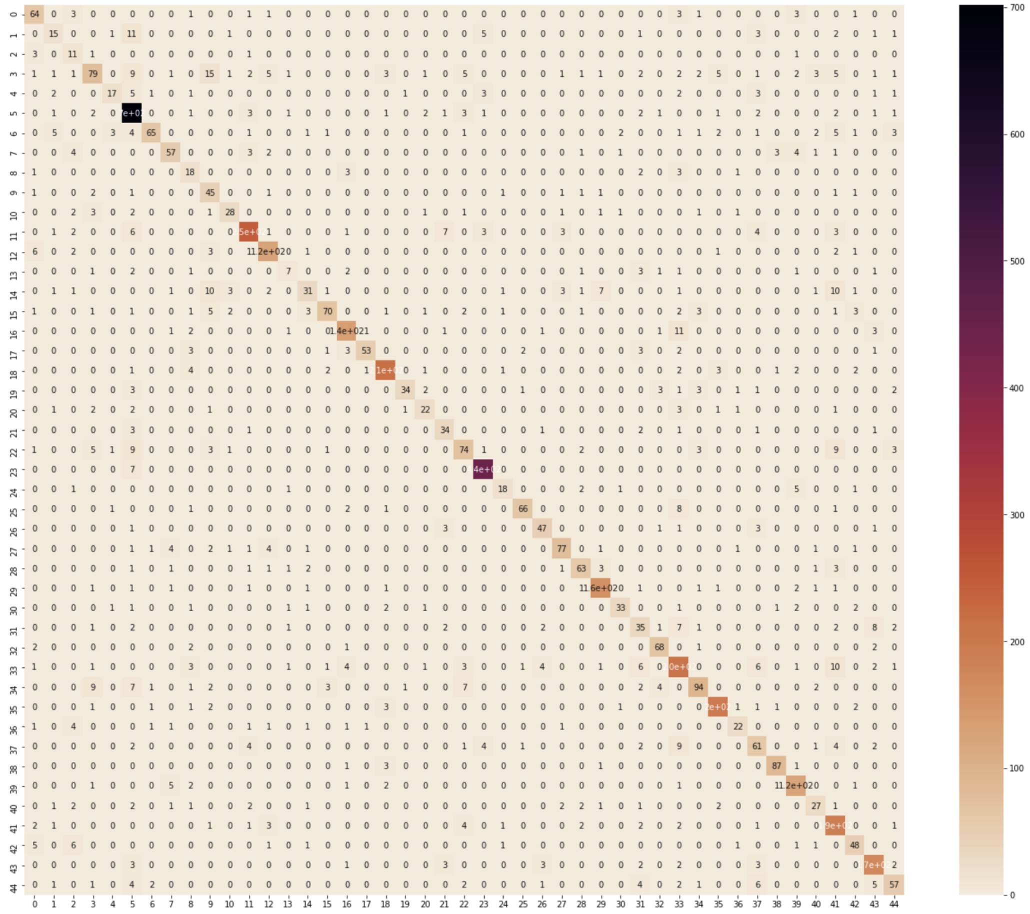
```
         cc: 0.5734
         Epoch 9/15
         43479/43479 [==============================] - 830s 19ms/step - loss: 1.3980 - acc: 0.5860 - val_loss: 1.3758 - val_a
         cc: 0.5945
         Epoch 10/15
         43479/43479 [==============================] - 822s 19ms/step - loss: 1.3127 - acc: 0.6102 - val_loss: 1.2861 - val_a
         cc: 0.6202
         Epoch 11/15
         43479/43479 [==============================] - 824s 19ms/step - loss: 1.2312 - acc: 0.6341 - val_loss: 1.2351 - val_a
         cc: 0.6365
         Epoch 12/15
         43479/43479 [==============================] - 844s 19ms/step - loss: 1.1598 - acc: 0.6570 - val_loss: 1.2052 - val_a
         cc: 0.6462
         Epoch 13/15
         43479/43479 [==============================] - 828s 19ms/step - loss: 1.0920 - acc: 0.6737 - val_loss: 1.1519 - val_a
         cc: 0.6614
         Epoch 14/15
         43479/43479 [==============================] - 832s 19ms/step - loss: 1.0317 - acc: 0.6929 - val_loss: 1.1352 - val_a
         cc: 0.6678
         Epoch 15/15
         43479/43479 [==============================] - 834s 19ms/step - loss: 0.9710 - acc: 0.7101 - val_loss: 1.0542 - val_a
         cc: 0.6918

Out[63]: <keras.callbacks.History at 0x19324e4e0>

In [64]: accr = model.evaluate(test_data,test_label)
         print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))

         5368/5368 [==============================] - 27s 5ms/step
         Test set
           Loss: 1.079
           Accuracy: 0.688
```

We made sure that our model has not overfitted by analyzing the loss curve and accuracy curve. With the progress, we added the number of epochs and higher embedding dimension. The resulting model produced the plot below with a curve that shows how the model improved at each epoch until the model started to overfit and final testing accuracy plateaued at 0.8. The final results are as follows:

Accuracy Plot:



Confusion matrix:



4 Conclusion

In conclusion, our project examines a wide variety of models to tackle the problem of authorship attribution. Authorship attribution can be thought of as a standard text classification task and many researchers have built models and evaluated them on authorship attribution together with other tasks, such as topic identification, language identification, genre detection, etc. Our model utilized state-of-art models such as CNN and LSTM and achieved very promising results with the limited amount of data in our possession.

Although we made significant progress, we believe there is still room for improvement beyond the use of deep learning models. Further research with pre-training models such as BERT can be of great benefit in solving the problem of authorship attribution. Pre-training models utilize the power of generalized models trained on millions or even billions training examples and using them is definitely an area future research should be looking into to improve our results.

5 References

- https://www.aclweb.org/anthology/R13-1010.pdf
- https://www.aclweb.org/anthology/J14-2003.pdf
- https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001
- https://www.researchgate.net/publication/338033403_Open_Set_Authorship_Attribution_toward_Demystifying_Victorian_Periodicals
- https://scholarworks.iupui.edu/handle/1805/15938
- https://web.cs.dal.ca/~vlado/papers/pacling03.pdf
- https://scholarworks.iupui.edu/bitstream/handle/1805/15938/abdulmecits-purdue-thesis.pdf?sequence=1
- https://towardsdatascience.com/a-machine-learning-approach-to-author-identification-of-horror-novels-from-text-snippets-3f1ef5dba634
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.1634&rep=rep1&type=pdf
- https://cs224d.stanford.edu/reports/RhodesDylan.pdf
- https://machinelearning.technicacuriosa.com/2017/09/04/nlp-powers-revolutionary-authorship-attribution-system/