CNN based Authorship Identification

Mentor- Nishit Asnani
nishit@iitk.ac.in

Hariom Panthi(150263)
phari@iitk.ac.in

Rahul Kumar(150548)
krrahul@iitk.ac.in

Rishav Raj(150587)
rishraj@iitk.ac.in

Abstract

Authorship identification is the task of identifying the author of a given text from a set of given authors. The main concern of this task is to define an appropriate characterization of texts that captures the writing style of authors. From machine learning perspective, it can be viewed as multiclass single-label text classification task where author represents a class (label) of a given text. There are three main problems in author identification: Closed-class,Open-class and Profiling[1]. We will be working on Closed-class author identification. We will achieve this using CNN . The input will be a piece of text and the output will be the author of the given piece of text from a fixed group of authors.

1 Introduction

Nowadays, a large number of articles, paragraphs and other forms of text materials are published. All of them are written in different styles. This is because each author has a different set of vocabulary, style and various other latent features which may be very difficult for humans to recognize. Authorship Identification has a wide range of applications in today's world. It can be used to find authors of various articles and also find plagiarized paragraphs. It can also be used to find authors of various ancient texts that are found.First, we used text analysis for author identification followed by Convolutional Neural Network.
We have made a comparison between the CNN implementation and text analysis implementation. We could clearly see that CNN proved to be much better than text analysis for this task.

2 Related Works

Authorship attribution using statistical methods has a long history. One of the first published works is Mosteller and Wallace which used statistical language-modeling techniques to attribute the true known author to disputed Federalist Papers. Since then AA has continued with using a broad spectrum of features and modeling approaches. We divide approaches to two main sets, features used and methods used. Features include lexical, syntactic and Content-Specific.

- Lexical Features Viewing a text as a sequence of tokens grouped into sentences. Those features could be divided into character-based and word-based lexical features.
- Syntactic Features The usage of syntactic information to fetch the unconscious syntactic patterns used by the authors at the sentence level, such as: part-of-speech, sentence structure, function words frequency and typos.
- Content-Specific Features A set of features to be defined for a specific domain (topic) by domain experts. Those features are often avoided due to its inability to generalize in cross-topic settings.

- CNN implementation for authorship attribution of short texts using n-grams[2].Character and word n-grams help determine the author of a document by capturing the syntax and style of an author.

3 Approach

3.1 Text Analysis Implementation

In this case, we manually selected features which we thought would help us distinguish between various authors. We also used various features which had already been used in previous works on authorship attribution. We used the following features:

- Average sentence length(avg. word count)
- Average sentence length(avg. character count)
- Average word length
- Number of unique words
- Counting parts of speech in each paragraph
- Number of punctuation in each paragraph

Next, we used various multi-class classifiers for training. The classifiers that were used are the following:

- SVM
- Perceptron
- SGD Classifier
- Gaussian NB
- Bernoulli NB
- MLP Classifier

3.2 CNN

Our model is based on Yoon Kim's paper on CNN for sentence classification[3] which performs sentiment analysis. We have modified this model and used it for Author Identification.

3.2.1 Data Preprocessing

The data has been cleaned before feeding into the input layer. This involves the following tasks:

- removing spaces between lines
- padding each sentence to a maximum length.
- We have built a new vocabulary for our data set while training. Each sentence is mapped to an integer between 0 and the vocabulary size. Thus, each sentence becomes a vector of integers.

We are taking input as batches of lines from the novel where each batch contains 64 lines.

3.2.2 The Model

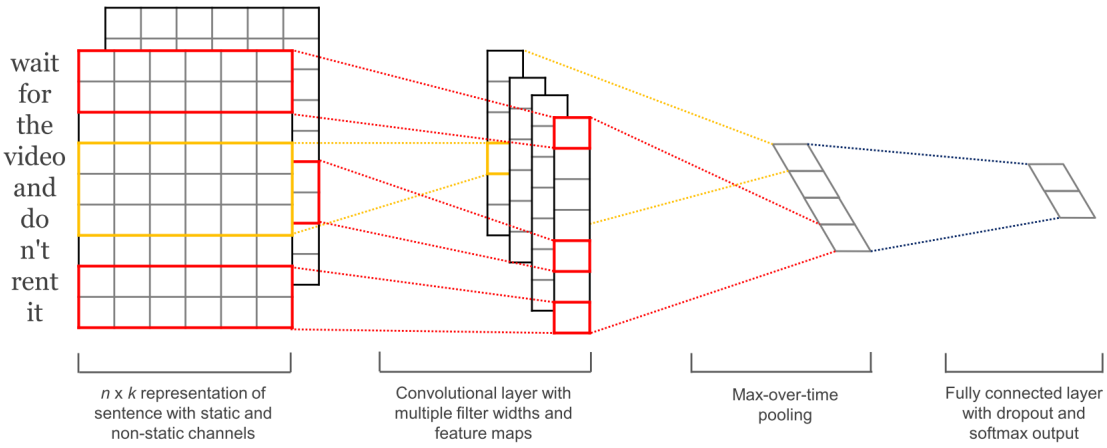

Figure 1: source:http://d3kbpzbmcynnmx.cloudfront.net/wp-content/uploads/2015/11/Screen-Shot-2015-11-06-at-8.03.47-AM.png

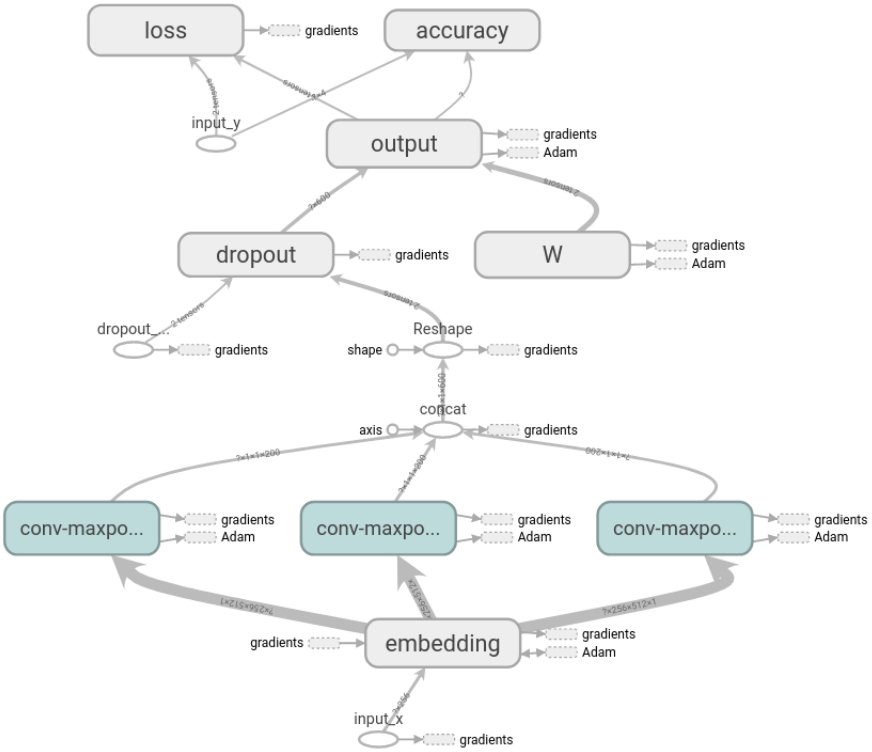

Figure 2: The figure above is the actual and complete representation of our model.

The embedding matrix was randomly initialized and learned during training. The first layer is the embedding layer which maps vocabulary word indices into low-dimensional vector representation.[3]

Let $x_i \in R^k$ be the k-dimensional word vector corresponding to the i-th word in the sentence. A sentence of length n is represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus .....x_n$$

where $\oplus$ is the concatenation operator.
In general, let $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, ..., x_{i+j}$.
A convolution operation involves a filter w $\in R^{hk}$, which is applied to a window of h words to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by

$$c_i = f(w.x_{i:i+h-1} + b)$$

(b $\in R$ is a bias term)
This filter is applied to each possible window of words in the sentence { $x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}$ } to produce a feature map

$$c = [c_1, c_2, ..., c_{n-h+1}]$$

Now we perform max pooling over the feature map.Then, dropout is used with a drop probability of 0.5 to learn the weight of all the edges.Multiple features have been extracted using multiple filters. Finally, we use softmax layer to perform classification of text.

4 Experiment and Result

Text Analysis Approach For this approach, data was taken from 4 novels from 4 different authors chosen randomly.

Approach	No. of Authors	Accuracy
SVM	4	33.97
Perceptron	4	33.96
SGD Classifier	4	38.55
GaussianNB	4	40.03
BernoulliNB	4	40.56
MLPClassifier	4	43.58

CNN Approach

We selected few novels from the Gutenberg dataset[1] randomly. For each author, few paragraphs were taken from different authors for training.Testing was done on paragraphs from different novels of the same authors.

Dataset	No. of Authors	Accuracy (lbl[2])	Accuracy (pw)[3]
Gutenberg	4	42.23	83.33
Gutenberg	8	24.72	70.83
Gutenberg	12	19.89	62.5
Pan12[4]	8	11.49	37.5
Random	4	72.12	100

Random dataset consists of 4 novels from 4 different authors chosen randomly. In this case, the test data was taken from the same novel which was used for training.

5 Discussion

5.1 Limitations

In the text analysis implementation,the linear model could not account for the non-linearity in data. Although we fed various features, we may have missed out on various subtle and important features

[1] source:https://drive.google.com/file/d/0B2Mzhc7popBga2RkcWZNcjlRTGM/edit?usp=sharing

required for classification. The CNN implementation requires lots of computation. The computation increases with increase in the number of authors. We need to increase the number of nodes which makes training extremely slow. Also, space required increases greatly with increase in the number of nodes.

5.2 Future Works

CNN performed well on the task of authorship attribution. This creates a premise for using it for other language modelling tasks as currently RNN is mostly used for such tasks.

6 Conclusion

In this project, we experimented with text analysis methods and CNN for author identification. CNN performs much better than text analysis. This shows that CNN could extract better features than what we fed in our model. However, it requires much more data and proper training.

References

[1] Bagnall, D.: Author Identification using multi-headed Recurrent Neural Networks. In: Cappellato, L., Ferro, N., Gareth, J., San Juan, E. (eds.) Working Notes Papers of the CLEF 2015 Evaluation Labs (2015)

[2] Convolutional Neural Networks for Authorship Attribution of Short Texts by Prasha Shrestha, Sebastian Sierra and Fabio A. González

[3] Kim, Yoon. 2014. Convolutional neural networks for sentence classification. arXiv preprint

arXiv:1408.5882 .