Authorship Identification in News Articles using Transformers

Can Koc
UC Berkeley
cankoc@berkeley.edu

Joshua Dunn
UC Berkeley
joshdunn@berkeley.edu

Abstract

Authorship identification is the task of identifying the writer of a particular text by analyzing the patterns in the text itself. Many approaches to author identification have been developed over the decades including statistical modeling, Bayesian inference, and even deep learning. We show improved accuracy performance compared to previous deep learning models, including those which used long short-term memory (LSTM) neural network or gated recurrent unit (GRU) neural networks. We achieve this performance increase by using transfer learning on pretrained bidirectional encoder representations (BERT) models. However, both our model and the previous model may be performing topic identification rather than author identification due to the nature of the data used and the design of the tests.

1 Introduction

Authorship identification is the task of identifying the author of a given piece of text through the content of the text itself. Every author or speaker has a unique style of sentence structure and vocabulary. This forms an identifiable ”fingerprint” which can be identified through statistical approaches (stylometry) and machine learning. Author identification has real-world applications where identifying an author of a text is important. Examples of this include identifying the anonymous author behind historical documents such as the Federalist Papers, ghostwriters of presidential speeches, detecting cheating within college admissions or standardized test essays, combating plagiarism in research papers or news articles, and even aiding criminal investigations of hate speech in social media.

Natural language processing (NLP) has seen a renaissance of powerful language models built with deep learning of neural networks. The state of the art used to be LSTM networks which retain information over a series of data. However, the advent of transformer neural networks has led to large increases in NLP performance over LSTM models. By using transfer learning on pretrained transformer networks, we hope to have greater success in author identification than previous deep learning approaches.

2 Related Work

Being one of the most popular and oldest classification tasks in NLP, the vast majority of previous research on identifying authors from a text focused on extracting features to capture both the content and the writing style of an author (eg. use of punctuation, specific grammar, specific way of starting or ending a paragraph), where the latter proved to be challenging. In this context, (Muttenthaler et al., 2019) showed that it's possible to achieve sufficient accuracy (69.5%) on PAN-2019 dataset through the use of term frequency-inverse document frequency (Tf-Idf) on character n-grams with a support vector machine (SVM) classifier. However, their model was unable to capture the stylometric representation of authors and rather focused on the content. To capture the style, (Madigan et al., 2005) used a predetermined set of stylometric features as an input to a multinomial logistic regression model to achieve around 41% accuracy on 114 authors from RCV1 corpus.

With recent accomplishments in language modeling, deep learning started to make a big impact in authorship identification. (Qian et al., 2017) showed that a LSTM network could be used to increase the performance of author identification on the C50 dataset. They showed that article-level Gated Recurrent Unit (GRU) networks performed the best with an accuracy of 69.1% on the C50 data. (Barlas and Stamatatos, 2020) showed that pretrained language models (ULMFiT, ELMo, GPT-2 and BERT) can be very good at classifying authors in cross-domain. This is when the text of an author during training differs greatly in content than in testing thereby showing great promise towards capturing stylometry. Close to the approach presented in this work, (Fabien et al., 2020) showed that fine-tuning a pre-trained BERT$_{BASE}$ model on datasets such as ENRON, Blog and IMDB can yield great results. In this work we take this idea further by fine-tuning a pre-trained BERT$_{BASE}$ and BERT$_{LARGE}$ models on the C50 dataset with AdamW optimizer. To the best of our knowledge this is the first work to study author identification on C50 dataset using pre-trained BERT transformer models.

3 Approach

3.1 Baseline

We use transformer neural networks and transfer learning to apply deep learning to author identification. Because (Qian et al., 2017) has performed this task using LSTM/GRU networks, we use their approach as a framework for ours. Our experiment differs in the type of neural network (transformers) and we use transfer learning from pretrained models. This allows us to use their results as a baseline for comparison of the performance differences between LSTM/GRU networks and transfer learning on transformer networks.

3.2 Model Architecture

We perform transfer learning of models sourced from HuggingFace which is an open-source repository containing code libraries, datasets, and pretrained models. We mainly experimented with BERT$_{BASE}$ and BERT$_{LARGE}$ described in (Devlin et al., 2018). These models are readily available for download through the HuggingFace model hub under the names "bert-base-cased" and "bert-large-cased" respectively. The differences between these two models are explained more extensively in (Devlin et al., 2018). Both models were trained on a corpora containing bookcorpus and wikipedia. The larger model has about $230M$ more parameters and achieves better overall accuracy in most applications.

The pretrained BERT models from Hugging Face has the raw hidden states in their last layer and in order to accomplish the transfer learning, we add several layers after it. The first addition is a dropout layer with a dropout parameter of 0.1 for normalization. Next, we add a fully connected layer to convert from the BERT model output shape to a 50-wide output shape to match the number of authors in the C50 data. Finally, we use a GELU layer for activation. A high-level illustration of the model layer order is shown in 2.

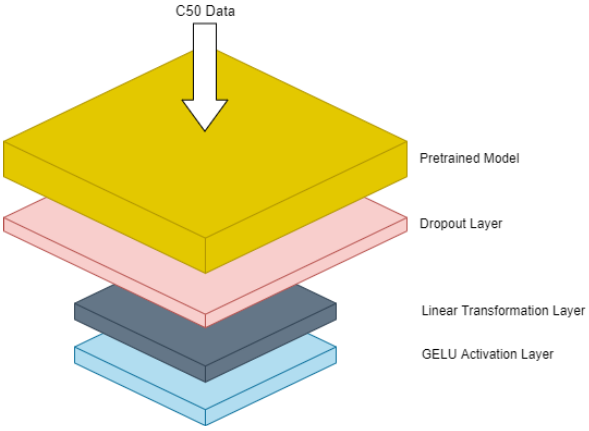

Figure 1: High-level diagram of the model.

Any weights in the added layers are initialized to a random value and so further training is needed to complete the transfer learning.

The AdamW optimizer is used in conjunction with a linear-decreasing weight scheduler. This causes the model to decrease the learning rate every epoch and faster results can be reached without bouncing out of the loss gradient minima.

3.3 Hardware and Performance

The transfer learning of pre-trained neural network models is best performed on GPU-enabled computing resources such as the Nvidia Tesla or the Nvidia K80. We leverage Google Collab Pro+ which gives us access to a Tesla P100 GPU with 15.9 GiB memory. On this platform, we are able to train and evaluate models for 10-15 epochs in under an hour.

4 Experimental Setup

4.1 Data

We use the Reuter_50_50 (also known as C50) dataset. This is a subset of the RCV1 corpus where the top 50 authors are selected with respect to the total size of the news articles. The full RCV1 dataset contains 800,000 manually categorized Newswire articles. The C50 subset has exactly 100 articles per author and 50 authors. Both RCV1 and C50 have been extensively used in authorship identification research and in C50, the authors of texts labeled with at least one subtopic of the class CCAT (corporate/industrial) were selected. This is done to mini-

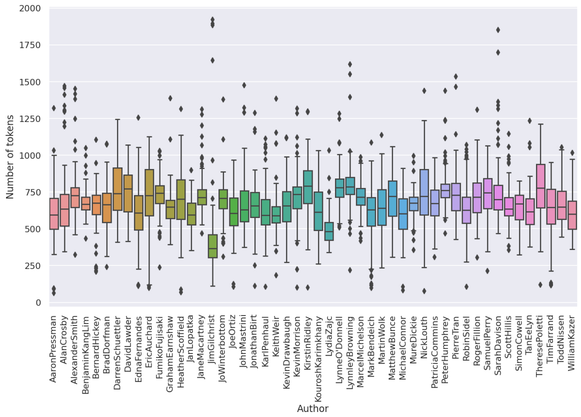

Figure 2: Number of tokens per author.

mize the topic factor in distinguishing authors from the texts. In order to understand the approximate distribution of number of words per author in our dataset we tokenized our data using the pre-trained tokenizer from the cased BERT$_{BASE}$ model. The articles have varying word counts and their distributions by author are shown in 2.

4.1.1 Train, Development and Test Splits

The original C50 dataset contains two mutually exclusive partitions of data with each containing 2500 articles. In order to increase our training data size, we merged the two parts into one and then randomly split the data into training, development and test sets of sizes 4000, 500 and 500 articles respectively. We pick a 9:1 split between training (including development) and test sets because the baseline model we are comparing against in (Qian et al., 2017) also uses this split ratio.

4.2 Fine-tuning BERT

4.2.1 Optimization

We primarily followed BERT fine-tuning instructions in (Devlin et al., 2018) with a few differences and also experimented with different configurations. Just like BERT, we used a dropout of 0.1, but we switched the optimizer to AdamW with a weight decay of 0.01. This allows the model to generalize better with respect to the C50 data size. We also use a linear scheduler without no warm-up and with an initial learning rate of $1e^{-5}$.

4.2.2 Handling Large Texts

The BERT models are restricted to only 512 input words. Most of the articles in the C50 dataset are greater than 512 words. During preprocessing, we either truncated or padded the articles to match the 512 token input size. The padding is done with the special $[PAD]$ token used by BERT for padding input sequences.

4.2.3 Mitigating Overfitting

Early training results over 10 epochs yielded nearly 100% accuracy on our training set and 79.1% and 80.4% accuracies on our validation set for BERT$_{BASE}$ and BERT$_{LARGE}$ respectively. The perfect accuracy on the training set and the significantly less-accurate performance on the validation set indicated the model was overfitting.

To combat this overfitting, we increased the dropout to 0.5 and switched the dropout layer to come after the fully connected layer. However, this caused the model to no longer learn during training and the training accuracy dropped by nearly 50%.

In another effort, we followed the fine tuning instructions in (Devlin et al., 2018) closely by switching our final activation layer from ReLU to GELU, reducing the number of epochs, and adjusting the initial learning rate from $1e{-}5$ instead of $1e^{-6}$. This greatly reduced the difference in training accuracy and validation accuracy without decreasing the final test accuracy.

4.3 Results

We report our best model results in Table 1. We found that training over 5 epochs with 0.1 dropout before the fully connected layer with a AdamW weight decay of 0.0 for BERT$_{BASE}$ and 0.01 for BERT$_{LARGE}$ yielded the best results. For both of the models, we started training from a learning rate of $1e^{-5}$ and used a linear scheduler to lower the rate down to $2e^{-6}$. (Devlin et al., 2018) recommends using batch sizes of 16 or 32 for fine-tuning, however due to the limitations of the CUDA memory available in the Collab Pro+ platform, we were only able to use a batch size of 16 for training BERT$_{BASE}$ but had to use batch size of at most 4 for BERT$_{LARGE}$. BERT$_{LARGE}$ did considerably better than BERT$_{BASE}$ in our validation set and test set and both models were able to beat the baseline accuracy of 69.1% that's reported in (Qian et al., 2017).

Model	Dev Acc.	Test Acc.	Decay
baseline	-	0.691	-
bert-base-cased	0.678	0.77	0
bert-large-cased	0.788	0.808	0.01

Table 1: Our model results compared against baseline.

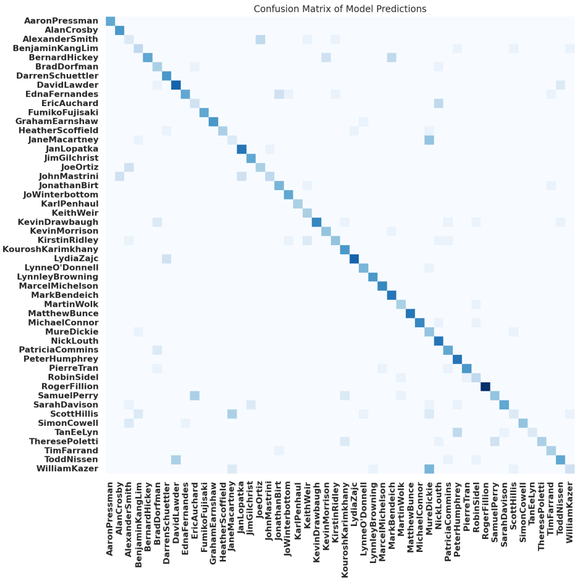

Figure 3: The model's confusion matrix on the test data. The X-axis are the predicted labels and the Y-axis are the true labels.

5 Discussion and Future Work

5.1 Analysis

This work presents how we can fine-tune pretrained BERT transformer models to authorship identification task on the C50 dataset. In our paper, we show that without significantly long hours of training, relying on single Tesla P100 GPU with 15.9 GiB memory, we are able to achieve sufficiently large accuracies on the C50 dataset using transformers and beat previous baseline article level GRU and LSTM based approaches. Although we show marked improvements over the LSTM/GRU, we have concerns about how our model, and the LSTM/GRU model discerned between the different writers. We believe our model does not make the distinction on stylometry or sentence structure. We intended for the model to learn *how* an author wrote rather than *what* an author wrote. The former is more desirable as it would indicate the model could generalize far better than the latter and is based on style rather than topic.

Upon inspection of the confusion matrix of the model prediction, shown in Figure 3, we discovered misclassifications occurred between certain pairs of authors. We then investigated the test data to try understand why the model made these errors.

One example of a misclassified author is Mure Dickie, who's work is often incorrectly attributed to William Kazer by the model. We found the test data for Mr. Dickie's work appears to primarily consist of coverage of events within Asian countries. In particular, his articles cover the political, economical, and social conflicts within China, Berma, Hong Kong, and Taiwan. Mr. Kazer's articles primarily cover economic and human right conflicts within China and the country's dealings with other world powers. Both of these journalists cover the same topics within the same part of the world. This indicates that the model is predicting authorship based on topic rather than style.

We then investigated the test data for some of the authors which the model predicted most accurately. This included David Lawder, Lydia Zajc, and Roger Fillion. Mr. Lawder's work in the test data covers the companies, workers, and unions within the United State's automotive manufacturing industry. Ms. Zajc's work in the test data covers the Canadian stock market and corporate dealings of Canadian companies. Mr. Fillion's work in the test data covers the US telecommunication industry including FTC decisions and their impact on consumers and corporations. Each of these authors cover distinctly niche topics and the model predicts their authorship with the greatest accuracy.

The output of the model produces a 50-wide vector of decimal numbers where each element correlates with a unique author. The GELU activation layer allows for the predicted class to be the maximum value of this layer. However, the other values are non-zero and offer insight into the distinction of the model's prediction distributions over the entire test set. Relationships between authors can be analyzed by each author's output distribution in the output vector space. The test data is processed by the model and we then performed a t-distributed stochastic neighbor embedding (t-SNE) on the final layer outputs to reduce the 50 dimensional output to 2. The results are then further limited to these five interesting authors and plotted in Figure 4.

In this reduced-dimensional plot, Mure Dickie and William Kazer are in close proximity and each have wider distributions (noted by the spacing between contour lines). This shows the model's output did not strongly delineate between the two authors' topics and/or style. The opposite is true for Lydia Zajc, David Lawder, and Roger Fillion who show tight and more separate distributions. This indicates the model had greater distinction in classifying the work by each of these three authors.

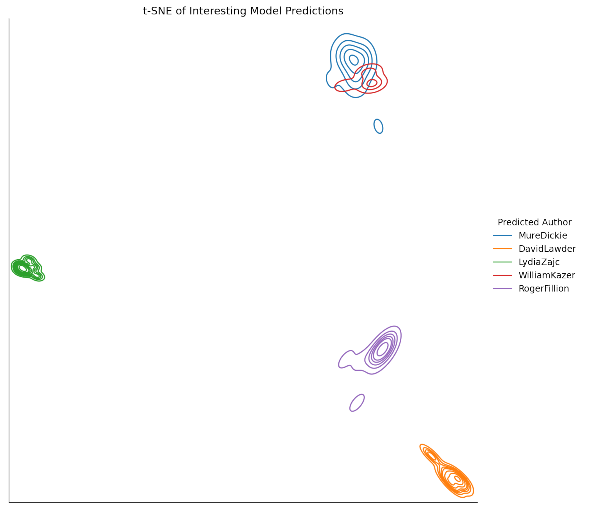

Figure 4: The t-distributed stochastic neighbor embedding of the model's output with only the interesting authors' distributions shown.

5.2 Issues

5.2.1 Topical Discernment

The inspection of the test data and the model output leads us to believe our model does not predict authorship based by style but rather by topic. The model's classification errors offer valuable insight of topical similarity between journalists covering the same field. We can say that the model does very well, and better than previous efforts, to identify authors who wrote the text in the C50 dataset.

Although we outperformed the similar work done by (Qian et al., 2017) using transfer learning of transformer models, we attribute the model's misdirected behavior to our project design. We failed to appreciate early that each journalist tends to cover discrete subjects with geopolitical and topical distinction. Our model learned to distinguish between 50 'topics' in the corpus which happened to correlate with each journalist's coverage of certain fields.

5.2.2 Model Limitations

The BERT models are restricted to an input of only 512 tokens. This is much less than most of the articles in the data. We found that many of the articles have similar exposition structure of giving a brief summary of the event, the background, and then a more full exposition into the specifics. A few authors even used nearly the exact same set of sentences for these background portions across many articles with only minor changes. Because only 512 words were used from each article, these event and background summaries were all the model is exposed to in training and evaluation. Much of the useful and more distinct information found in the later portions of the articles were never used for training or evaluation.

5.3 Proposed Solutions

5.3.1 Topical Discernment

We believe the best solution for the topic-author coupling issue is to preprocess the data to remove or replace topical meaning from the data. A part-of-speech tagger, such as HuggingFace's flair, could be used to preprocess the data to part-of-speech notation. All topical information would be removed and only the sentence structure remain.

Future work would include this part-of-speech preprocessing and potentially using a different base model to accommodate the difference in data.

5.3.2 Model Limitations

The model training and evaluation should be modified to allow for multiple 512-word chunks to be processed before a final classification is made. For a sequence of chunks, the model's output from the fully-connected layer could be aggregated. This aggregation could then be averaged and then ran through the GELU activation layer. Although continuity between chunks would be lost, the model could process and classify the entire document rather than the first 512 words.

A second approach would be to randomly sample a continuous 512-word chunk from each article. However, this would still have the issue where much of the data in each article is left out of training and evaluation.

A third solution is to use oversampling by splitting the articles into 512-word chunks and thereby increasing the number of text documents in the data. This would certainly cause the data to become unbalanced and thus warrant the usage of class weights in training and a more appropriate evaluation metric.

References

Georgios Barlas and Efstathios Stamatatos. 2020. Cross-domain authorship attribution using pre-trained language models. In *Artificial Intelligence Applications and Innovations*, pages 255–266, Cham. Springer International Publishing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA : BERT finetuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

David Madigan, Alexander Genkin, David Lewis, and Dmitriy Fradkin. 2005. Bayesian multinomial logistic regression for author identification. *25th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*.

Lukas Muttenthaler, Gordon Lucas, and Janek Amann. 2019. Authorship attribution in fan-fictional texts given variable length character and word n-grams. In *CLEF*.

Chen Qian, Ting He, and Ren Zhang. 2017. Deep learning based authorship identification.