Author Identification of Reddit Users

ANLY 590 Neural Network and Deep Learning
12.15.2018

Katherine Schulz
MS in Analytics
Georgetown University
ks1533@georgetown.edu

Kendra Gedney
MS in Analytics
Georgetown University
kg729@georgetown.edu

April Chung
MS in Analytics
Georgetown University
ahc72@georgetown.edu

Abstract

We apply author identification, the task of classifying an author based on previously known samples of their text, to the social media domain. We create a new dataset of 100 Reddit users and their comments on which we train five models: one baseline SVM and four deep learning models. The simplest model performs the best. We also test the effect of removing Named-Entities from the corpora and find that accuracy decreases.

1 Introduction

Author identification has applications in digital forensics, anti-terrorism, and anti-plagiarism where it assists investigators in tracking the work of specific people (Stamatatos, 2009). Author identification relies primarily on the "extraction of stylometric features" that represent an author's general writing style, and models tends to perform better when trained on larger samples of an author's text (Markov, Stamatatos, & Sidorov, 2017; Qian, He, & Zhang, 2017).

In this research, we train text classification models on a set of cross-topic comments from 100 distinct Reddit authors. We take inspiration from Qian et. al. (2017) who used deep learning models for sentence-level and article-level author identification.

We train one support vector machine (SVM) and four deep learning models and test for classification accuracy. We also conduct an experiment with Named-Entities to determine how they affect the classification results.

2 Related Work

Author identification traditionally relies on training models on one or more of the three main types of extracted stylometric features: lexical, syntactic, and content-specific. Lexical features may include tokenizing text and examining character-based or word-based patterns, while syntactic features may include part of speech tagging, sentence structure, or word counts at the sentence level. Content-specific features may include words, phrases, or writing structures based on domain expertise in a specific topic (Ghanem, El-Makky, & Mohsen, 2016).

Prior to the widespread use of deep learning, researchers typically performed author identification with SVM classifiers, which were successful when trained over longer documents, but unsuccessful on shorter ones (Qian, He, & Zhang, 2017). Deep learning models have since replaced SVM classifiers, as deep learning models can uncover multiple layers of stylometric features without supervision (Ghanem, El-Makky, & Mohsen, 2016).

Successful approaches to author identification include Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) networks trained on word embedding initialized with Global Vectors for Word Representation (GloVe); however, these

models trained on cross-topic texts that were not pre-processed to minimize the effects of content-specific words. The researchers found that article-level models attained higher accuracy rates (between 60 and 70 percent) than the sentence-level models (between 40 and 45 percent) (Qian, He, & Zhang, 2017).

To deal with cross-topic texts, Markov, et. al. (2017) successfully demonstrated that pre-processing techniques, such as replacing digits and named entities using the Stanford Named Entity Recognizer (NER), improved classification accuracy rates by about 10 percent on several SVM and multinomial naïve Bayes (MNB) classifiers. They note, however, that in the case of single-topic modeling, the pre-processing does significantly improve the models' accuracy rates.

A model proposed by Joulin, et. al. (2016), presents a simple, efficient model for text classification that achieved high accuracy results. Though they did not test author identification specifically, their model can be applied to this task.

3 Datasets

Our dataset is a large collection of Reddit posts and the authors who wrote them. Reddit is a social media platform designed as discussion board, for which posts, or comments, are generally casual and short in length.

We collected our dataset using the Python library *PRAW*, which is a Reddit API wrapper. First, a set of authors was collected from the *r/PoliticalDiscussion* post about the 2018 US midterm elections. We collected all posts, across subreddits, written by each of those authors. We then filtered by post length, such that all posts are between 60 and 1,000 characters. We also filtered by authors that had written at least 500 posts. Finally, we selected 100 authors randomly using the *numpy.random.choice* function. The final dataset contains about 72,600 posts from 1,844 different subreddits. A summary of the final dataset is shown in Table 1.

Dataset Summary	
Size (rows)	72,600
Authors	100
Posts per Author	506 - 902
Characters per Post	60 - 1,000
Topics per Author (subreddits)	8 - 172
Vocabulary Size	53,000

Table 1: Description of final dataset collected from Reddit

3.1 Dataset for NER Experiment

For our second experiment, we subsetted the primary dataset to include 14 of the 100 authors and 10,000 posts. We chose to subset for efficiency purposes, since running the Stanford NER algorithm was computationally expensive. It took approximately 10 hours for the processing to complete using a graphics processing unit (GPU) runtime host on Google Colaboratory.

4 Methods

Our research has two parts. First, text classification models are fit on the full dataset to identify the authors. Next, we test the effect of removing Named-Entities on the accuracy of the models using a subset of the full dataset.

4.1 Author Identification

The dataset was randomly split into a training set (80 percent) and test set (20 percent). For the primary author identification task, five models were fit: an SVM as a baseline and four deep learning models.

The four deep learning models fit are:

- Long Short Term Memory (LSTM)
- Convolutional Neural Network (CNN)
- CNN + LSTM
- Simple Pooling

For each of the deep learning models, we set the maximum number of features to be 25,000 for efficiency purposes. The first layer is an embedding layer of size 128. All were trained for 16 epochs using the Adam and RMSprop optimizer and a batch size of 128. The generalized structure for each is shown in Figure 1.

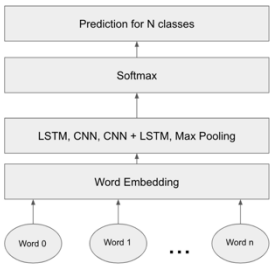

Figure 1: Generalized structure of deep learning models.

The Simple Pooling mode's architecture is based on the simple text classification model proposed by Joulin, et. al. (2016). It contains a word embedding layer followed by a 1D Global Max Pooling layer.

4.2 NER Experiment

Our dataset is cross-topic since it contains comments from over 1,800 different subreddits. Based on the success of Markov, et al. (2017), we applied their NER replacement method, called Stanford NER, to our subsetted dataset of 14 authors. Specifically, we replaced all digits with "0", and replaced all persons, locations, and organizations with "#".

We then fit a baseline SVM and the four deep learning models, with the same architecture as described in Section 4.1, to both versions of this subsetted dataset: one with Named-Entities in the texts intact, and one with Named-Entities replaced with a symbolic token.

5 Results

Results for both experiments are based on test accuracy.

5.1 Author Identification Results

The Simple Pooling model performed the best, as indicated in Table 2**,** which shows the training and test accuracies for the SVM and four deep learning models. The large differences between the training and test accuracies indicate that our models overtrained on the given data. In addition, Table 2 shows the test accuracies for the models at k = 5 and k = 10 when we filtered for the top "k" authors each model produced (in the case of the initial test accuracy, k = 1).

Model	Training Accuracy	Test Accuracy	Test Accuracy (k=5)	Test Accuracy (k=10)
SVM	**0.99**	0.38	0.59	0.68
LSTM	0.91	0.24	0.44	0.55
CNN	0.50	0.23	0.45	0.58
CNN + LSTM	0.52	0.24	0.47	0.59
Simple Pooling	0.89	**0.42**	**0.64**	**0.74**

Table 2: Results by model for the author identification task.

Table 2 also shows that test accuracies for all models increased as "k" increased from 1, to 5, to 10. These results indicate that we can reach up to 74 percent accuracy of finding authorship of the comment from the pool of 10 users we narrowed from 100. For a generally a hard task in identifying authorship from relatively short comments in social media, the best model (Simple Pooling) does well to help narrow our search list by giving the top "k" results.

The accuracy results for the more complicated models, the CNN and the CNN +LSTM, were lower than the simpler models, both SVM and Simple Pooling. For the purposes of plotting accuracy, we trained the model for 50 epochs, and the following graph in Figure 2 shows accuracy in blue and the test accuracy in green for the CNN+LSTM model. The training accuracy continually increases with fitting, with an indication of overfitting at about 30 epochs since the test accuracy stops increasing. Moreover, we see that the test accuracy plateaus a little above 20 percent.

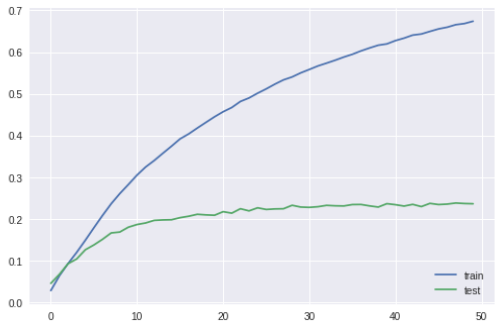

Figure 2: Test and train accuracy by epoch for the CNN + LSTM model. Results looked similar for the CNN model.

5.2 NER Experiment Results

For our second experiment, we replaced Named-Entities with "#" to better generalize our cross-topic data and prevent the models from overfitting on topic-specific words. Table 3 below shows test accuracy by model on a subset of the data with the Named-Entities intact versus replaced. Overall, the test accuracy for each model decreased when the Named-Entities were replaced, indicating that the Named-Entities provided the models valuable information to distinguish authors.

Model	Named-Entites Removed	Named-Entities Intact
SVM	**0.55**	0.58
LSTM	0.37	0.46
CNN	0.38	0.48
CNN + LSTM	0.34	0.44
Simple Pooling	0.51	**0.63**

Table 3: Test accuracy results at k=1 for the second experiment in which Named-Entities are replaced from the subsetted dataset.

The results also indicate that the Simple Pooling and SVM models performed the best at author identification on the given data, similar to the first experiment. With Named-Entities removed, the SVM model attained an accuracy rate of 55 percent, compared to rates of 34 to 51 percent for the other models. With the Named-Entities intact, the Simple Pooling model attained an accuracy rate of 63 percent, compared to the rates of 44 to 58 percent.

6 Discussion

Overall, author identification of Reddit authors proved to be a difficult task, with accuracy rates all under 50 percent. Though it was a difficult task, the simple models, both SVM and Simple Pooling, performed the best in this domain. Using the complex, high-capacity LSTM and CNN models did not lead to any accuracy gains - only losses.

However, we show that deep learning methods can lead to accuracy gains over traditional methods in the case of the Simple Pooling model. This supports the work of Joulin, et. al. (2016), for which they used a similar architecture to achieve high accuracy in text classification accuracy. Since both LSTM and CNN models consider the order of text, these results suggest that order may not be too important in author identification in this domain.

Although our accuracy rates at k=1 may seem low, under 50 percent, our models are successful in reducing a pool of potential authors, which has useful applications in this domain.

Finally, the Named-Entities replacement did not improve author identification results, differing from the results of Markov, et al. (2017) for which they tested on a news corpus. This suggests that in the social media domain, the presence of Named-Entities helps the models learn.

7 Conclusion

To conclude, this research has shown that Reddit author identification based on the text of the post is difficult. However, deep learning models can help improve results. Cross-topic corpora confounds the task. In our case, the Named-Entities helped classification, but that may have negative effects on transferability and generalizability. Future research could further test their effects or consider single-topic corpora.

References

A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, "Bag of Tricks for Efficient Text Classification", Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2016.

Ghanem, N., El-Makky, N. M., & Mohsen, A. M. (2016). Author Identification using Deep Learning. Retrieved November 2018, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7838265

Markov, I., Stamatatos, E., & Sidorov, G. (2017, April). Improving Cross-Topic Authorship Attribution: The Role of Pre-Processing. Retrieved November 2018, from http://www.cic.ipn.mx/~sidorov/CICLing-Markov-Preprint.pdf

Qian, C., He, T., & Zhang, R. (2017). Deep Learning

based Authorship Identification. Retrieved November 2018, from https://web.stanford.edu/class/cs224n/reports/2760185.pdf

Stamatatos, E. (2009, March). Retrieved November 2018, from A Survey of Modern Authorship Attribution Methods: https://pdfs.semanticscholar.org/d25c/27c7a3e9f41f150e8eadbad34c1c05d67510.pdf?_ga=2.101008001.1800055236.1543592516-1748937950.1543592516