Authorship Verification with One Class SVM using BERT Embeddings

Liam Johnston

1 Introduction

Authorship verification is a Natural Language Processing (NLP) task that involves determining who is the author of a given document. It holds significant importance as it has the potential to be used as a defense mechanism against impersonation or unauthorized use of one's identity. To illustrate this, imagine a scenario where an individual gains unauthorized access to your email account and sends an inappropriate message to your employer. An effective authorship verification algorithm would be able to showcase the difference in writing styles between you and the imposter, therefore absolving you from any wrongful attribution.

Given the nature of the task, a one-class support vector machine (SVM) was used as the classifier. As a result, all training data was written by the same author and given a test instance, the model classifies it as either *normal*, meaning it was written by the same author who wrote the training documents, or an *anomaly* that was written by a different author. In this work, the performance of the one-class SVM on two different embedding algorithms, a term-document matrix, which acted as the baseline model, and Bidirectional Encoder Representations from Transformers (BERT), were compared through two different experiments. The first experiment involved solely including *normals* in the test set, whereas in the second experiment *anomalies* were added to the test set.

2 Dataset

2.1 About the Dataset

The dataset used in this work was derived by Schler et al. (2006) to use in their work that was published under the title *Effects of Age and Gender on Blogging*. It consists of 681,288 blog posts written in English from 19,320 different authors that were accessible on *blogger.com* one day in August 2004. The number of posts for a given author ranges from 1 to 4,221 with an average of 35 and a median of 11.

The construction of the dataset is that each author's posts are contained in their own file. An example file name is *9289.male.23.Marketing.Taurus.xml*, which indicates the author's ID, gender, age, occupation, and zodiac sign respectively. Additionally, each post in an author's file contains its posting date.

2.2 Using the Dataset

For the first experiment, the posts in a file were split in two halves where the first half acted as the training set and the second half acted as the testing set. Since the posts were ordered from least to most recent, doing this made sense for an authorship verification model as they have to take their knowledge gained from training documents and apply it to new documents that have been written after the model has been trained.

Test set creation for the second experiment involved hardcoding an arbitrary number of *normals*, then looping through the rest of the dataset and randomly adding *anomalies* from other author's files.

3 Models

As previously mentioned, 2 models were compared in this work, both of which used a one-class SVM classifier. The two embedding algorithms that were used was a term-document matrix, which acted as the baseline model, and BERT.

3.1 One-Class SVM

The training data for a one-class SVM is all of the same class. The model then builds a boundary from the training data as shown in Figure 1. Given a test instance, the model then determines if

its embedding is within the boundary in the vector space that it learned in training or if it is an *anomaly*.

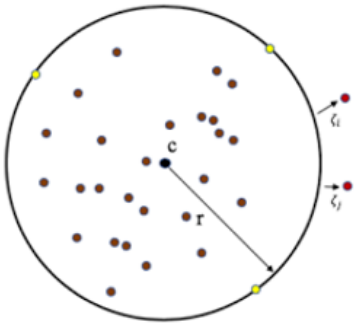

Figure 1: One-class SVM example.

The one-class SVM implemented in this work used a radial basis function (RBF) kernel. Its formula can be viewed in Equation 1, where σ is learned in training to define the boundary.

$$K(\mathbf{x}, \mathbf{x}') = exp(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2})(1)$$

3.2 Creating BERT Embeddings

Figure 2 shows the BERT embedding process for the input "[CLS] hello , world ! [SEP]". Note that "[CLS]" and "[SEP]" are special tokens inserted by BERT's tokenizer that indicate beginning of input and end of sentence respectively.

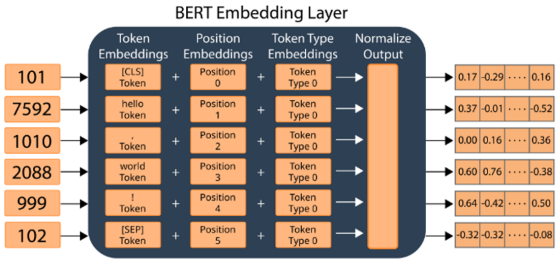

Figure 2: BERT Embedding process for "[CLS] hello , world ! [SEP]".

The first step of the process is to assign a token ID to each token. BERT's tokenizer has a vocabulary of 30,522 tokens. It then assigns a position embedding. BERT has a maximum of 512 position embeddings, so the largest input sequence accepted by a BERT model is 512 tokens long. If the input sequence is longer than 512 tokens, all tokens past the 512th position are removed. An option is to use the extra tokens as an additional input document, but that was not done in this work. There are only two different token type embeddings, 0 and 1, which is there for next sentence prediction, which is one of the tasks BERT was originally trained to solve.

The BERT embedding layer then combines these three embeddings to create the overall embedding for each token. The shape of the output from the BERT embedding layer shown in Figure 2 is (1, 6, 768), where the first element is the batch size, meaning number of input sequences, the second element is the number of input tokens per batch, and the third element is the hidden size, which is the size of the embedding for each token. To ensure normalization for the one-class SVM, all input sequences in this work were padded to a size of 512 tokens if they were shorter.

4 Results

4.1 Experiment 1

The purpose of conducting an experiment where all test instances were *normals* was to gain some preliminary knowledge on how the models perform before constructing test sets with posts from various authors. First, it was determined that the largest training set the computer used to run the models was capable of embedding using BERT was roughly 500 posts. Given this, 16 different files with at most 1,056 posts were handpicked for the experiment (recall from Section 2.2 that the posts in a file were split in half for train-test split in this experiment). Figure 3 shows the results of this experiment.

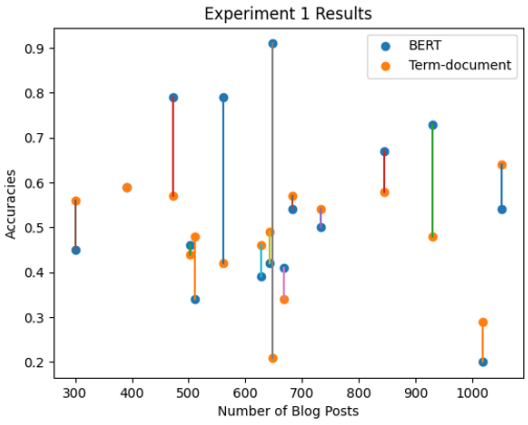

Figure 3: Experiment 1 Results.

Figure 3 indicates that the term-document embeddings had a higher accuracy than BERT across 9 of the 16 selected files (note that for one of

the files, the accuracies of the two embedding algorithms were very close, therefore the term-document point is the one shown as it was plotted last, however it's accuracy was in fact slightly higher). However, Figure 3 also shows that BERT has the 5 highest accuracies and that when it is more accurate than the term-document embeddings, it tends to be by a higher margin.

An interesting takeaway from Figure 3 is that the size of the training sets don't seem to have much of an impact on the accuracy of the models. Additionally, although it is not shown on Figure 3, the demographics of the authors selected for this experiment were varied and no discernable differences were discovered.

4.2 Experiment 2

For experiment 2, *anomalies* were added to the test sets. Three different training authors were used in this experiment, all of which were also selected in experiment 1.

4.2.1 Author 1

In experiment 1, the term-document and BERT embeddings obtained accuracies of 0.21 and 0.91 respectively for author 1. The test set was hard-coded to have 200 *normals*, which resulted in the train set having 448 posts. Note that this is larger than its train set of 324 posts in experiment 1. Running the test set creation algorithm resulted in approximately 1,000 anomalies being added.

Table 1 shows the results of the two models for just one test set after being trained on author 1's 448 training posts. It is worth noting that results were remarkably similar across multiple test sets, as in each metric's value was almost always within 0.01 of the values shown in Table 1. It is notable that the performances of the two embedding algorithms were quite similar, with BERT's being slightly better. Note that precision, recall, and F1 scores were calculated with *anomalies* acting as positives. This means that the high precision and low recall values means that when the models classify a test instance as an *anomaly*, they are almost always correct, but often misclassify *anomalies* as *normals*.

4.2.2 Author 2

Author 2 was selected because it had a similar number of posts as author 1. For this experiment, the train set contained 442 posts and the test set contained 200 *normals* and approximately 1,000 *anomalies*, which is approximately the same test set size as author 1. The accuracies of the term-document and BERT embeddings in experiment 1 for author 2 was 0.49 and 0.42 respectively.

Metric	Term-Document	BERT
Accuracy	0.60	0.63
Precision	0.92	0.98
Recall	0.56	0.56
F1 Score	0.70	0.71

Table 1: Comparison of results in Author 1 of Experiment 2 between the Term-document and BERT embeddings.

Metric	Term-Document	BERT
Accuracy	0.81	0.84
Precision	0.90	0.87
Recall	0.87	0.93
F1 Score	0.88	0.91

Table 2: Comparison of results in Author 2 of Experiment 2 between the Term-document and BERT embeddings.

Table 2 shows the results of the two models from one test set after being trained on author 2's 442 training posts. Interestingly, the models performed better on author 2 than author 1 in experiment 2 when the reverse was true in experiment 1. The only metric with values higher in author 1 compared to author 2 was precision. Note that the better performance on *normals* can be attributed to the increased size of the training set and that as with author 1, the results remained consistent across multiple test sets.

4.2.3 Author 3

Author 3 was selected because it had a smaller training set than that of the first two authors. It had a training set of 200 posts, and a test set of 100 *normals* and approximately 450 *anomalies*. The accuracies of the term-document and BERT embeddings in experiment 1 for author 3 was 0.45 and 0.56 respectively.

As with the previous two tables, Table 3 shows the results of the two models from one test set after being trained on author 3's 200 training posts. Although the training set size was less than half of that of author 2, the models performances were almost exactly the same.

Metric	Term-Document	BERT
Accuracy	0.80	0.84
Precision	0.89	0.88
Recall	0.87	0.93
F1 Score	0.88	0.90

Table 3: Comparison of results in Author 3 of Experiment 2 between the Term-document and BERT embeddings.

5 Conclusions

Findings from both experiments reveal a seemingly random variance in performance amongst authors. A possible explanation for this phenomenon is that those who consistently post about the same topics allow the models to obtain higher accuracies, whereas those who post about different topics will yield lower accuracies. Time constraints prevented further testing and obtaining averages across authors with various demographics and number of posts. Another interesting finding from experiment 2 that should be further studied is the consistent performance across different *anomalies* in the test set when the same training data was used.

5.1 Future Work

For future work, more effectively pipelining the process of selecting the training author would allow for experiments to run more efficiently, therefore gaining more empirical evidence for takeaways.

Furthermore, the project could be run on cloud servers to obtain more memory for testing BERT embeddings on larger datasets. However, this is not completely necessary as the goal of authorship verification models is to work well on the least amount of possible training data as one author generally does not generate a massive amount of training data.

Lastly, the impact of adjusting the one-class SVM kernel could be examined. Additional kernel options provided by scikit-learn's one-class SVM implementation which was used in this work, are linear, polynomial, sigmoid, and precomputed.

References

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*. volume 6, pages 199–205.