Author classification as pre-training for pairwise authorship verification

Notebook for PAN at CLEF 2021

Romain Futrzynski[1]

[1]Peltarion, Holländargatan 17, 111 60 Stockholm, Sweden

Abstract

In this paper, we propose to use a standard BERT model for the PAN 2021 Authorship Verification task where two texts must be determined to either have the same or different authors. The model is chiefly trained to classify short sequences of text as belonging to one of three thousand authors selected from the large training dataset. Additional tasks are also used simultaneously during training in order to capitalize on the information available, namely, a masked language model task, a fandom classification task, and an author-fandom separation task. To perform Authorship Verification, an embedding is extracted from the trained BERT model. In order to reduce the computational cost, only a short sample of text is processed by BERT, but the same text is sampled a hundred times at random locations, and the embeddings from each sample are reduced to a single representation using the median. The representations from two texts are compared by cosine similarity, which is rescaled empirically so that most of the ambiguous pairs lie on the 0.5 threshold. Evaluated on authors and topics absent from the training dataset, this model achieved F1=0.832 and AUC=0.798.

Keywords

BERT, similarity

1. Introduction

Authorship verification [1] is one of the shared tasks at PAN 2021 [2]. The purpose of this task is to determine whether or not two texts from a given pair were written by the same author, without any prior knowledge about the authors or the topics of the texts. The texts used for authorship verification are stories extracted from www.fanfiction.net happening within a fandom (i.e., a popular fictional universe such as *Harry Potter*, *Twilight*, *True Blood*), written by fans of this fandom. The particular fandom of a text is known during training, but it is not available to models during evaluation on the test set. Moreover, both the authors and fandoms included in the training set are different from those included in the test set.

The method proposed here is to train a standard BERT [3] model using an author classification task. Since the authors in the unseen test set are different from the authors available for training, the model cannot be used to directly identify specific authors. Instead, a representation of an input text, in the form of an embedding vector, is extracted from the model before its

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

romain@peltarion.com (R. Futrzynski)
https://fromain.medium.com/ (R. Futrzynski)
0000-0002-3194-5141 (R. Futrzynski)

classification layer. By using a large number of authors during training, the model learns to produce representations of text that can be compared *via* cosine similarity to determine if they share the same author. In addition, the model is trained simultaneously on a masked language model task, a fandom classification task, and an author-fandom separation task in order to induce desirable properties in the model.

2. Dataset preprocessing

The dataset used for training is the large training set from the CLEF 2020 authorship verification task [4]. This dataset is composed of text pairs labeled with whether they were written by the same or different authors. An anonymous author identifier and the fandom are also provided for each text of the pair. In order to train the model using a classification task, the dataset is first reorganized as a list of 493,296 unique texts labeled with their author and fandom.

A single author may have written between 1 and up to 30 texts, within 1 to 6 fandoms. In order to provide as much diversity as possible while not slowing down the training time, a training split is created by gathering every text from the authors who have written within exactly 6 fandoms, resulting in a training split containing 72,471 distinct texts, from 3,353 distinct authors, and covering 1,471 distinct fandoms.

A validation split is also created from the training texts. Only authors who have written within 5 fandoms or less may be included in the validation split, ensuring that the authors from the training and validation splits are distinct. The validation split is structured as text pairs in order to to be evaluated in a fashion similar to the end task. To constitute the validation split, 2,500 text pairs are sampled at random ensuring that they were written by different authors, and 2,500 text pairs are sampled at random ensuring that both texts were written by the same author, and furthermore ensuring that the two texts from a given pair are distinct. The validation split therefore contains 5,000 text pairs, for a total of 9,816 distinct texts, within 1,253 distinct fandoms, from 7,216 distinct authors who do not overlap with the authors of the training split.

3. Model description

The model used relies on the BERT [3] architecture and English pretraining as provided by the `bert-base-cased` [1] model from the `transformers` Python module.

3.1. Model input

This model is given a tokenized text as input. The texts contained in the dataset are often full stories, commonly reaching over 5,000 tokens. Since the computation time required by transformer models grows quadratically with the length of the input sequence, the model is only given 28 consecutive tokens, picked at a random location in the text which is independent for every text and every epoch. Besides decreasing the computation time, it is expected to force the model to focus on brief writing patterns. The size of 28 aims to promote focusing on such

[1] https://huggingface.co/bert-base-cased

short patterns, while still providing enough tokens to let the model leverage the contextual understanding of self-attention.

The sequence of tokens is also prepended by a CLS token whose embedding is intended to be used for classification tasks, and appended with a SEP token which brings to the total model input size to 30 tokens. Although the model is always given texts one at a time, the SEP token is added as a way for the model to store information without affecting any of the classification or language modeling tasks used during training, and to resemble the pretraining setup more closely.

3.2. Model output

For classification tasks during training, and for the authorship verification task during validation and test, the pooler vector is used. This vector is the result of passing the embedding of the CLS token through a linear layer followed by a hyperbolic tangent activation function.

The first half of this pooler vector, i.e. its first 384 values, serves as an embedding of the author, which is used for author classification during training and for similarity evaluation during validation and test. The second half of the pooler vector, i.e. its last 384 values, is used only for fandom classification during training.

4. Training procedure

The model is trained using the four tasks described below. The optimizer used is AdamW [5] with a learning rate of 2e-5, and a batch size of 250. The linear layers used for the author and fandom classification training tasks use a different, higher learning rate of 1e-3. The model is trained for 20 epochs.

4.1. Language model task

During training, 10% of the input tokens are randomly replaced by a mask token. This differs from the 15% rate used in the original pretraining in order to avoid masking too much of the relatively short token sequences. The language model head of BERT is run on the sequence of token embeddings, and its ability to recover the original token for every masked token is tuned using a crossentropy loss.

The purpose of the language model task is to prevent catastrophic forgetting of the original pre-trained weights, and to promote learning of author-specific words and idioms.

4.2. Author classification

The model is simultaneously trained to identify the author of every text in the training split as a classification task. For this purpose, the model uses the first half of the pooler vector, passed to a linear layer with an output size corresponding to the number of unique authors in the training split, that is, 3,353. No activation function and no bias term are used in this linear layer, in order to promote the learning of 384-size embedding vectors that are more directly suitable

for comparison using cosine similarity.
The performance of the author classification task is tuned using the crossentropy loss function.

4.3. Fandom classification

Similarly, the model is trained to identify the fandom of every text in the training split as another classification task. For this task, the second half of the pooler vector is used. This is passed to another linear layer with an output size of corresponding to the number of unique fandoms in the training split, that is, 1,471. No activation function is used in this layer, although the bias term is enabled. The purpose of this task is to promote fandom awareness, and may possibly contribute to improving the language model in conjunction with the author classification task.
The performance of the fandom classification task is tuned using the crossentropy loss function.

4.4. Author-fandom separation

Since the end task is to verify authorship independently from fandom, it is undesirable that the model uses its predictions about fandom in order discriminate different authors. To counter this phenomenon, a simple network is trained to classify fandoms from the first half of the pooler vector, which is normally intended to embed authors only. This network is made of a linear layer with bias and output size of the full pooler vector, i.e. 768, followed by the SELU [6] activation function, a 10% dropout layer, and a last linear layer projecting to the number of fandoms without bias.

This network is trained using a crossentropy loss in parallel but independently from the main model, therefore using the same training batches but having its own gradient updates. Then, the crossentropy loss from this network is recalculated as part of the main model training. This loss is scaled by a coefficient of 0.1 and subtracted from the sum of the losses from three other tasks. This creates the total loss that is used for training. The reason for scaling down the loss of this task is that since every author in the training split has contributed to only 6 fandoms, good fandom classification could reasonably be expected even if the authors were embedded using style consideration only.

5. Validation procedure

The validation task is to receive two texts, and return whether or not the author is the same. As this differs from any of the training tasks, a different procedure is used.

A single text may easily range in the thousands of tokens, whereas the model is trained using a sequence length of 30 tokens. To proceed, 100 short sequences of 28 tokens are sampled at independent random locations within a text, and the CLS and SEP tokens are added similarly to the training format.
The model then runs a forward pass, and the first half of the pooler output vector is stored for each of these 100 sequences as an embedding of the sequence. These embeddings are then reduced to a single 384-component vector using the median of each component.
The same steps are repeated for the second text of a pair, yielding another median-reduced vector.

Table 1
Comparison of using a 10 and 384 embedding size, measured after 20 epochs on the validation set.

	AUC	c@1	F1	F0.5u	Overall
10-component embedding	0.830	0.760	0.771	0.737	0.781
384-component embedding	0.864	0.802	0.817	0.757	0.810

Finally, the two reduced vectors are compared using cosine similarity. Since the task rewards models that answer 0.5 as an uncertain result, it is desirable to scale the cosine similarity scores so that similar and dissimilar authors lie on each side of 0.5. For this purpose, the ROC curve on the validation split is plotted every epoch, and the threshold corresponding to the halfway step is monitored, giving an approximation of the middle of the curve. After 20 epochs, the threshold is 0.6592 which is rounded to 0.65. Therefore, during the test run, the values of cosine similarity between 0 and 0.625 are linearly rescaled between 0 and 0.5; the values between 0.675 and 1 are rescaled between 0.5 and 1; the values between 0.625 and 0.675 are collapsed to 0.5. The rescaled cosine similarity is reported as the answer to the test task.

6. Results

The evolution of the model performance over epochs is reported here. In addition to the model described which uses an embedding size of 384, half the embedding size of BERT, progress is also reported for another model using only 10 components in its embedding. In this model, the first 10 values of the pooler vector are used as author embedding, the next 10 values are used for fandom classification, and the remaining 748 values are simply ignored. Although it appears to perform relatively well, as shown in Table 1, only the model using 384 components was submitted since it has the best overall performance.

Furthermore, the plots show progress over 100 epochs but the model submitted was only trained for the first 20 epochs. The purpose of submitting only the 384-component model after 20 epochs is to preserve as much information from BERT's original pretraining as possible in order to help generalization to unknown authors and fandoms. In order to speed up training, the validation metrics are calculated from only 600 texts pairs sampled at random from the 5,000 text pairs contained in the validation split set. Finally, the plots are smoothed using a gaussian distribution of standard deviation 4 epochs, with the original curves shown as a lighter shade. shows the performance for both the 10-component embedding and the 384-component embedding model.

Table 2 shows the final results of the model, evaluated on the unseen test set using the TIRA platform [7].

7. Concluding remarks

It is interesting to note that the metrics measured on unknown authors reach similar values whether an embedding size of 384 or 10 is used. However, the smaller embedding size has inher-

Figure 1: Validation metrics calculated on a random subset of the the validation split after every epoch of training.

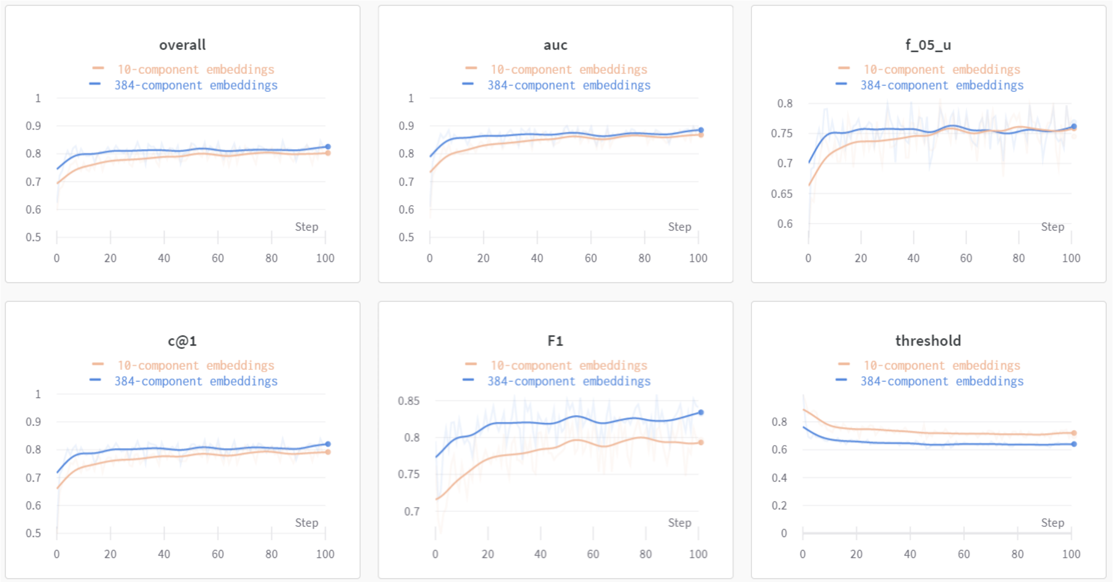


Figure 2: Losses of the first three training tasks as a function of the epoch number.

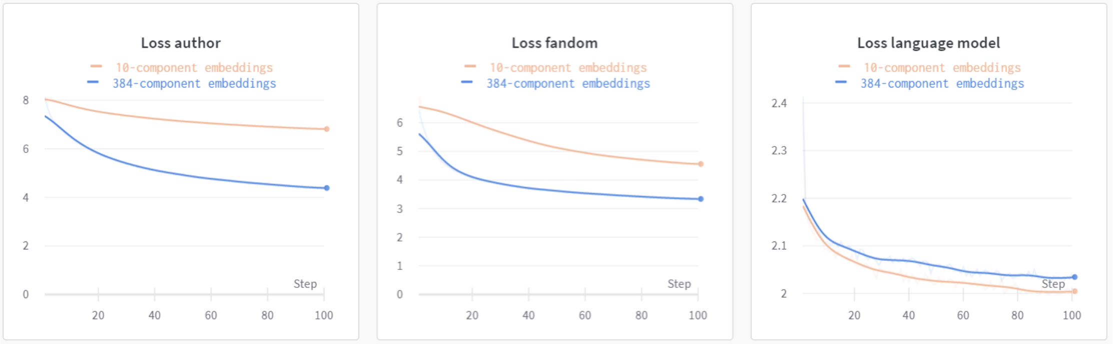


Table 2
Official final results of the 384-component embedding model, with *overall* score calculated as the average of all metrics.

AUC	c@1	F1	F0.5u	Brier	Overall
0.798	0.663	0.832	0.668	0.796	0.752

ently less capacity to store information so that its generalization performance on significantly different data can be questioned.

Further studies, notably regarding other embedding sizes, the length of the input sequence, and the amount of short sequences sampled from a large text, would provide interesting information about the performance of models trained for classification on similarity-like tasks.

Figure 3: Accuracy of the first three training tasks as a function of the epoch number.

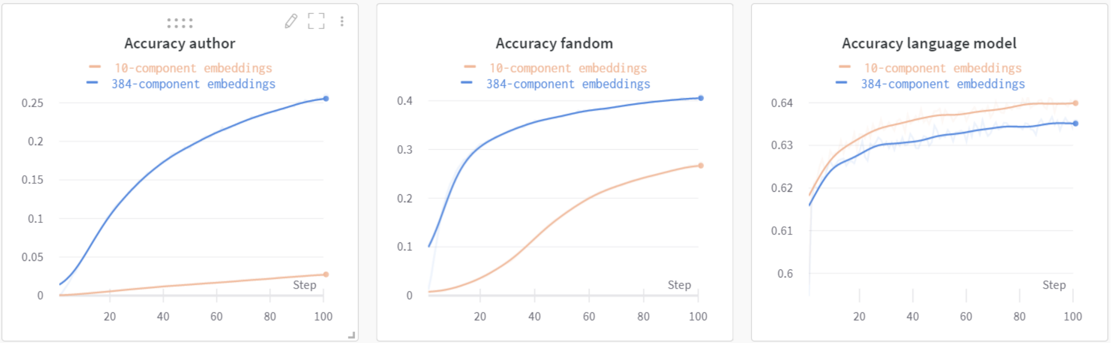
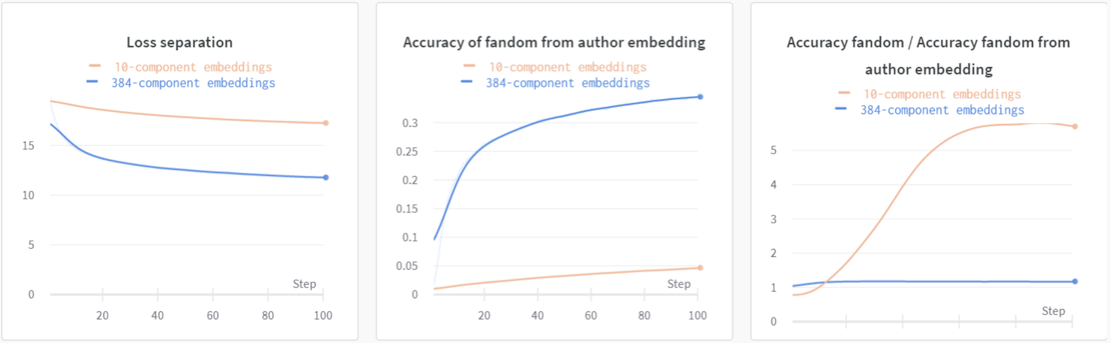


Figure 4: Left: loss of the network classifying fandoms from the author embedding. Center: accuracy of the network classifying fandoms from the author embedding. Right: ratio of the accuracy obtained by the main fandom classification task over the same accuracy obtained by the network working from the author embedding.



While the model only processes short sequences of text, this must be repeated several times in order to get more reliable results. As a result, using the model as it is to regularly process large amounts of text pairs may be problematic, especially on hardware that isn't specifically designed for tensor operations.

References

[1] M. Kestemont, I. Markov, E. Stamatatos, E. Manjavacas, J. Bevendorff, M. Potthast, B. Stein, Overview of the Authorship Verification Task at PAN 2021, in: CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[2] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. arXiv:1810.04805.

[4] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, B. Stein, Overview of the Cross-Domain Authorship Verification Task at PAN 2020, Working notes of CLEF 2020 - Conference and Labs of the Evaluation Forum (2020) 22–25. URL: https://pan.webis.de.

[5] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017. arXiv:1711.05101.

[6] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, 2017. arXiv:1706.02515.

[7] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.