# Will Longformers PAN Out for Authorship Verification?

## Notebook for PAN at CLEF 2020

Juanita Ordoñez*, Rafael Rivera Soto*, and Barry Y. Chen

Lawrence Livermore National Laboratory
{ordonez2, riverasoto1, and chen52}@llnl.gov

**Abstract** Authorship verification, the task of identifying if two text excerpts are from the same author, is an important part of evaluating the veracity and authenticity of writings and is one of the challenges for this year's PAN @ CLEF 2020 event. In this paper, we describe our PAN authorship verification submission system, a neural network that learns useful features for authorship verification from fanfiction texts and their corresponding fandoms. Our system uses the Longformer, a variant of state-of-the-art transformer models, that is pre-trained on large amounts of text. This model combines global self-attention and local self-attention to enable efficient processing of long text inputs (like the fanfiction data used for PAN @ CLEF 2020), and we augment the pre-trained Longformer model with additional fully-connected layers and fine-tune it to learn features that are useful for author verification. Finally, our model incorporates fandom information via the use of a multi-task loss function that optimizes for both authorship verification and topic correspondence, allowing it to learn useful fandom features for author verification indirectly. On a held-out subset of the PAN-provided "large training" set, our Longformer-based system attained a 0.963 overall verification score, outperforming the PAN text compression baseline by 32.8% relative. However, on the official PAN test set, our system attained a 0.685 overall score, *underperforming* the PAN text compression baseline by 7.6% relative.

## 1 Introduction

As more of us rely on online sources for our news and information, it becomes increasingly important to vet their veracity and authenticity. One key component of this vetting process is identifying the authorship of the information. Knowing the author of the information can help us better ascertain its trustworthiness which is critical in light of the growing amount of online misinformation propagated by so-called trolls, bots, and other online agitators. There has been a lot of prior work on computational and statistical methods for determining the authorship of text writings based on writing style: word choice, punctuation usage, idiosyncratic grammatical errors, and in more recent digital texts the use of emoticons. One vibrant community in which these computational approaches to authorship identification are developed and evaluated is PAN [1].

* These authors contributed equally.

PAN hosts scientific evaluations for digital text forensics and stylometry, and one of this year's tasks in PAN @ CLEF 2020 is that of authorship verification [12], where the goal is to determine whether two separate text excerpts come from the same author or not. This notebook paper describes our team's final submission system and some of the experiments and alternative systems that we developed for the authorship verification challenge.

PAN @ CLEF 2020 builds off of earlier evaluations in Authorship Identification tasks [13] that used fanfiction [5] as the source material for the challenge. Milli, et al. [18] describes fanfiction as "fan-created fiction based on a previously existing, original work of literature." Fanfiction is derived from the original work but extends or changes certain aspects of the fiction like providing more development of minor characters, adding additional characters, modifying the relationships between characters, altering endings, etc. While fans may strive to write in the style of the original work's author, it is interesting to see if subtle writing style differences still make it possible to distinguish between the original and fan authors as well as between different fan authors. Our goal is to develop a system that can take two excerpts of fanfiction and determine whether they come from the same fan or not. In addition to the text in these excerpts, we are also given the fandom from which these excerpts derive.

Our approach is to use neural networks to learn text and fandom embeddings that are useful for authorship verification. Our final submission system is built using a variant of state-of-the-art transformer models [26] that have recently been setting the pace on a wide variety of natural language processing tasks such as translation, question-answering, cloze tasks, and language modeling [8]. We use the Longformer model [3] pre-trained on about 6.5 billion tokens of text from multiple online sources including English Wikipedia, real news outlets, book versions of movies, and a subset of CommonCrawl dataset. The Longformer is a computationally efficient transformer model for long text excerpts like the ones found in fanfiction. Our system augments the Longformer with additional fully-connected layers for the authorship verification task as well as a complementary fandom correspondence task.

The key contributions that we will cover in our PAN Notebook paper are the following:

1. We train our neural network models to incorporate auxiliary fandom information using a multi-task loss function that combines both authorship and fandom correspondence classification losses.
2. We investigate the effectiveness of large-scale text pre-training for authorship verification by building a system using the state-of-the-art Longformer model [3], a transformer-based model [26] that efficiently models long text excerpts such as the fanfiction data in the PAN 2020 Author Verification Challenge.
3. We compare our word-based Longformer system with two PAN-provided baselines and a character-based convolutional neural network ($(CN)^2$) with self-attention system.

## 2 Related Work

Authorship verification and authorship attribution are related subfields within the larger field of Forensic Authorship Analysis. Authorship verification seeks to determine if two different writings are from the same author, while authorship attribution's goal is to identify who wrote a given writing. Both of these fields rely on the extraction of useful text features for discriminating between different authors. Traditionally, researchers have relied on linguistic style or stylometric features, such as the counts and frequency of function words, average length of sentences, part-of-speech, characters, punctuation, whitespace usage, and other low-level features [25].

More recently, with the advent of many successful end-to-end deep learning systems in computer vision, speech recognition, and natural language processing, researchers have begun exploring the idea of learning what features are most useful for both verification and attribution. Many researchers have explored using convolutional neural networks (CNNs) and have successfully used them to extract text features [15] that are helpful for attribution [10,23,24,2], where the CNNs learn author discriminative n-grams of words and characters. In [10], researchers learn embeddings for both words and parts-of-speech (POS) tags and show improved generalization performance. Word level features are good at capturing an author's word usage style and oft used phrases, but they ignore other writing nuances such as punctuation, whitespaces, abreviations, and emoticons. Character-based CNNs excel at modeling these aspects; [23] shows that character based CNN perform especially well for large scale authorship attribution as the number of authors increase. One of the systems that we explored for PAN, our Character-based Convolutional Neural Network $(CN)^2$, extends the work on character-level CNNs by combining CNNs with self-attention [26] layers to hone in on the most discriminative combinations of character-level n-grams.

Computational and neural network-based approaches for Natural Language Processing (NLP) have seen a spike in the growth of their popularity mostly due to the effectiveness of their usefulness across a wide-range of NLP tasks. Simple word-embedding techniques like Word2Vec [17] and GloVe [20], learn to "embed" or project words into continuous feature vector spaces such that related words are proximal in feature space. Subsequent classifiers can then use these pre-trained embeddings for NLP tasks such as sentiment analysis, syntax parsing, semantic role labeling, etc. More recently, sophisticated language models, some built using recurrent neural networks, like ELMO [21], and others derived from the self-attention based Transformer models [26] making them well-suited for efficient training on massive amounts of data, have attained state-of-the-art performance on diverse sets of NLP tasks [9,8,16,3]. These models, trained on increasingly large "Internet Scale" data (as in the case of [8]), learn features that can represent long sequences of text, and these features are then used in downstream NLP tasks. For PAN, we wanted to see if these state-of-the-art language models pre-trained on large amounts of data could be used for the authorship verification task. In section 3.4, we describe our word-based system that uses the Transformer variant specifically developed to efficiently model long text excerpts like fanfics called the Longformer [3] which is available through the Hugging Face's excellent Transformer repository [27].

Many authorship verification systems have explored architectures other than CNNs and Transformers. Some have sought to learn features using autoencoder-inspired ar-

chitectures from a variety of word and character n-grams and POS [11]. Recurrent Neural Networks (RNN) have also been successfully used. One particularly promising approach is the work of Boenninghoff, et al. [7], who use a Siamese Network setup for transforming pairs of text excerpts to extract authorship features that can be compared for determining whether the two excerpts are from the same author. Their Hierarchical Recurrent Siamese Neural Network uses two Long Short-Term Memory (LSTM) layers trained using a modified contrastive loss function that seeks to project text from the same author to nearby locations in feature space.

Finally, this year's PAN baseline systems are two simple, yet effective approaches. The "TFIDF" baseline computes the term frequency-inverse document frequency normalized counts of character tetragrams of the pair of text excerpts, and then uses the cosine similarities between them as an authorship verification score. The "Compress" baseline is an adaptation of [6] to the verification task and uses a text compression technique based on Prediction by Partial Matching to compute cross-entropies between the text pair for attributing authorship. According to PAN [1], "The mean and absolute difference of the two cross-entropies are used by a logistic regression model to estimate a verification score in [0,1]". This technique follows similar work using text compression for building authorship profiles in [14].

## 3 Methodologies

Our work explores neural networks as a means for extracting discriminative features directly from the text without any explicit feature engineering. Towards this end, we explore two models: Character Convolutional Neural Network $(CN)^2$, and Longformer. More detail about each model will be given in Section 3.3 and Section 3.4.

At a high level, there are three components to our approach: text feature extraction, topic embedding, and final classification. Of these three, only the text feature extraction changes depending on which of the two models is being used. To embed the fandoms[1], we apply an embedding layer $E_{topic}$ that maps each fandom identifier to a $512$ dimensional vector. Then, the topic embeddings are combined with the text features and passed to two multi-layer perceptrons $M_{author}$ and $M_{topic}$ for authorship verification and topic correspondence respectively. The whole process is outlined in Figure 2 for the $(CN)^2$ model.

### 3.1 Tokenization

Depending on which model is being used for text feature extraction, our tokenization differs:

1. **$(CN)^2$** - This model looks at the text from the standpoint of characters, including all punctuation and white-space as tokens. This model can thus focus more on the syntactic quirks of each author rather than the semantic meaning. The size of the

[1] In this paper we refer to fandoms as "topics" interchangeably since the fandom of a piece of writing can roughly be considered to be its topic.

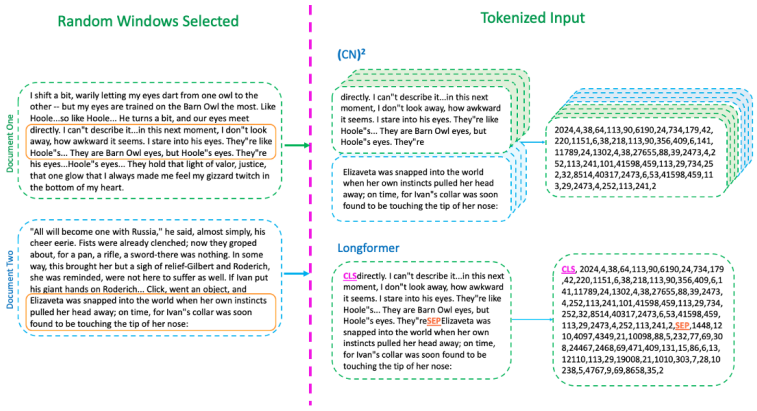


**Figure 1.** This is an example of how we prepared the input for each network. In $(CN)^2$, we take 5 random windows from each pair, map each character to its corresponding index, and stack all windows on top of each other. In Longformer, we take one random window from each document, map each word to its corresponding index, concatenate both windows side by side separating it with the <SEP> token and prepend the <CLS> token.

character vocabulary is $2,500$ and $4,800$ for the small and large version respectively. In the case where a character in the validation set in unseen, we use an "unknown" token.

2. **Longformer** - This model looks at the text from the standpoint of words and can be thought of as placing more emphasis on the semantic meaning and particular word combinations. Here, we tokenize our document using RoBERTa's tokenizer which has a vocabulary of size 50,265 tokens. More detail about this tokenizer can be found in [16].

### 3.2 Model Input

Because the average length of the fanfiction excerpts is about 20,000 characters long making it difficult for our models and their intermediate feature representations fit in GPU memory, we chose to use multiple random windows from each excerpt as input. The number and length of random windows varies depending on the model being used:

1. **$(CN)^2$** - Uses 10 random windows, 5 from each pair, each composed of 1000 characters.
2. **Longformer** - Uses 511 words from each pair, these are then separated using the special <SEP> token, and the classification token <CLS> is prepended to learn the document features that are most beneficial for authorship verification. See Figure 3 and Section 3.4 for more information.

Figure 1 shows an example of how we prepare the fanfiction documents for input to our authorship verification systems.

### 3.3 Character Convolutional Neural Network + Recursive Self Attention

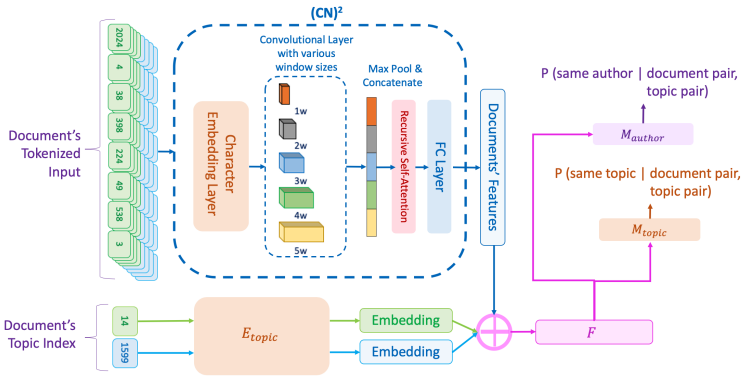


**Figure 2.** Architecture diagram for the (CN)$^2$ system. Five sets of random 1000-character windows are sampled from each input text excerpt and sent into the (CN)$^2$ network. We embed the topic (i.e., fandom) information and fuse it with our text feature representations learned by the (CN)$^2$ network. Finally, the multilayer perceptrons, $M_{author}$ and $M_{topic}$ predict author verification scores and topic correspondence scores respectively.

As mentioned previously, the (CN)$^2$ model looks at the document from the standpoint of characters. For input, we take in a set of 10 random windows, 5 from each document. When each random window is embedded, we stack them on top of each other along the channel dimension. Then, 1-D convolutions are applied over n-grams of size one through five after which we maxpool over the length dimension. For each n-gram, there are 512 different convolutional filters, and these filters essentially learn the character n-grams that are most useful for authorship verification. Letting the output of each 1-D convolution and maxpool be called $C_1, C_2, \cdots, C_5$, where $C_1$ is the output of the 1-D convolution and maxpool that looks at unigrams and $C_2, \cdots, C_5$ are the ones from higher order n-grams, we define an operation we call Recursive Self Attention:

$$out = SA(C_5 \oplus SA(C_4 \oplus SA(C3 \oplus SA(C_2 \oplus C_1)))) \quad (1)$$

Where $\oplus$ is the concatenation operation, and $SA$ is the Self-Attention operation. This allows the network to learn features in a hierarchical manner starting from the smallest n-grams to the highest. Finally, the output of this process is passed through a fully connected layer to generate the final documents' feature vector of size 512. The documents' features are then concatenated with the fandom embeddings forming a composite feature vector $F$, which is then sent to two separate two-layer multilayer perceptrons: one for learning the probability that the two documents are from the same author and the other for the probability that they come from the same topic (i.e., the fandom of the fanfic excerpt). The system architecture is shown on Figure 2.

### 3.4 Longformer

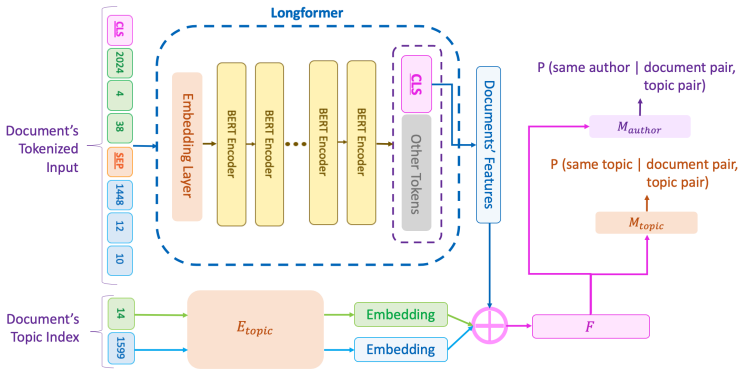


**Figure 3.** Architecture diagram for the Longformer system. Two sets of random 511-word windows are sampled from each input text, concatenated together with a separator token and prepended with a classification token. These are sent into the Longformer network. We embed the topic information and fuse it with our text feature representations embodied by the classification token output by the Longformer network. Finally, the multilayer perceptrons, $M_{author}$ and $M_{topic}$ predict author verification scores and topic correspondence scores respectively.

The Longformer [3] is an extension of the Transformer [26] which has achieved state-of-the-art results in many natural language tasks. The Longformer has the advantage of being pre-trained on approximately 6.5 billion word tokens and sets state-of-the-art results on Wiki-Hot and Trivia-QA [3]. Whereas the original Transformer architecture computed self-attention in $O(n^2)$ time, where $n$ is the length of the input sequence, the Longformer architecture introduces sliding window self-attention which scales linearly instead of quadratically. The main insight is to compute the self-attention locally instead of globally. Given a fixed window size of $w$, each token attends to $\frac{1}{2}w$ on each side thus resulting in an operation that can be computed in $O(nw)$. Both the global self-attention mechanism and the sliding window self-attention can be combined so as to integrate both global and local context into the computation. This allows the Longformer to take inputs that are much larger than those previously possible. See Figure 3 for a diagram of the architecture.

We continue training from the weights of the RoBERTa model released by Beltagy, et al. [3]. The weights of the embedding layer and the weights of the first ten encoder stacks are frozen, only leaving the last two Encoder stacks for fine-tuning. We use a local attention pattern with $w = 512$ for every token except the CLS token for which we use a global attention pattern as in the original Transformer architecture. We take a random window consisting of 511 words from each author, these words are then tokenized and the input to the model is "<CLS> <Excerpt 1 Tokens> <SEP> <Excerpt 2

Tokens>". Where CLS is the classification token and SEP is the separator token found in the original BERT [9] architecture. Finally, the embedding of the CLS token of size $N \times 768$ is taken as our documents' features, and the rest of the model flow follows as in the (CN)$^2$ approach. Our Longformer system is shown in Figure 3.

### 3.5 Multi-Task Loss Function

During training, our models optimize two objectives: authorship verification and topic correspondence. The joint loss function may be written as follows:

$$L(F, a, t) = \frac{1}{N} \sum_{i=1}^{N} BCE(\sigma(M_{author}(F_i)), a_i) + BCE(\sigma(M_{topic}(F_i)), t_i) \quad (2)$$

Where $N$ is the number of samples, $F \in \mathbb{R}^{Nx3D}$ are the features extracted from the model, $a \in \{0, 1\}$ are the author labels, and $t \in \{0, 1\}$ are the topic labels. The label $a_i$ is 1 if the text pair is written by the same author and 0 otherwise; $t$ is analogous to $a$ but for topic correspondence. Given the features $F$, we map them to authorship verification scores and topic correspondence scores using separate two-layer multilayer perceptrons $M_{author}$ and $M_{topic}$ respectively. These scores are then bounded between zero and one through the use of the sigmoid function $\sigma$.

BCE is the binary cross-entropy loss function here defined for one sample:

$$BCE(x, y) = -y \log(x) - (1 - y) \log(1 - x) \quad (3)$$

Where $x$ is a probability, and $y \in \{0, 1\}$ is the label.

### 3.6 Scoring

After training, the output of the neural networks $M_{author}$ and $M_{topic}$ approximate the posterior probabilities of the two excerpts being from the same author and being from the same topic respectively. We use the same-author posterior probability, the output of $M_{author}$, as our authorship verification score and can be thresholded at $0.5$ to decide whether the pair of fanfic excerpts are from the same author or not.

## 4 Experimental Setup and Results

### 4.1 Metrics

To evaluate our model, we used the script provided by the PAN organizers. This script implements four metrics: AUC, F1-Score, C@1, and F_0.5u. C@1 [19] and F_0.5u [4] are relatively new and have different properties. C@1 is a version of F1-score that rewards the system for leaving difficult problems unanswered, i.e., when the verification probability to be exactly $0.5$. In the case where there is a balance between correct answers and non-decisions this metric approximates the traditional accuracy computation. If the model is unsure about too many samples, the score tends towards zero.

F_0.5u places more importance on documents that come from the same author (i.e., true positives). This is in contrast to C@1 which treats both positive and negative examples with equal importance. Because of this, in F_0.5u the unanswered samples are treated as false negatives resulting in worse performance for models that leave a lot of samples unanswered. Finally, F_0.5u weights true positives higher compared to false positives and false negatives. Although both C@1 and F_0.5u interact with unanswered samples differently, we didn't map any scores to $0.5$ and simply used the raw same-author posterior probability as our verification score.

### 4.2 Dataset

The PAN Authorship Verification 2020 datasets come in two flavors: small and large. Each dataset was built by scraping fanfiction writings from fanfiction.net. Pairs of fanfictions along with their respective topic (fandom) were provided for training with the classification of positive (same author) or negative (different authors) as ground truth. For training our models, we used both the small and large versions of the dataset with our final submission being based upon a model that was calibrated on the large dataset.

The differences between both datasets is highlighted in Table 1. There are $52,655$ samples in the small dataset and $278,169$ in the large dataset, thus the large dataset contains about 5.28 times more samples. In terms of document length, the average number of characters is approximately 21,400 and the split between negative and positive samples is close to 50% on both datasets. Finally, both the small and the large dataset share the same $1,600$ fandoms.

To validate our approaches, we split both the small and large dataset into training and validation splits. To construct these splits, we performed the following steps:

1. Separate the positive and negative samples from each set.
2. Randomly choose 70% of the negative samples for the training set, and the remaining 30% for the validation set.
3. We pick 15% of the authors randomly and use their positive samples in our validation set. If two or more positive pairs existed, we used half for our training set and the rest for the validation set.
4. The positive samples of the remaining 85% of authors are used in the training set.

### 4.3 Results

Tables 2 and 3 show our evaluation results of the baseline systems, $(CN)^2$ and Longformer, on the validation set for the small and large datasets respectively. Comparing $(CN)^2$ and Longformer against the Compress Baseline on the small dataset (Table 2), both models see an improvement on every metric except $(CN)^2$ on AUC. Overall, $(CN)^2$ outperforms the baseline by 5% absolute, while the Longformer model outperforms it by 14.5%. On the large dataset (Table 3), the improvement is even more pronounced. $(CN)^2$ achieves an improvement of 12.7% while Longformer achieves an improvement of 23.8%. Both $(CN)^2$ and Longformer benefit from having more data available which highlights the possibility of further improving results by the addition of more data.

**Table 1.** PAN's authorship verification statistics: small and large version.

| Dataset Version | Small | Large |
|---|---|---|
| Average Size of Document (Character) | 21,441 | 21,428 |
| Size of Character Vocabulary | 2,542 | 4,811 |
| Number of Negative Examples | 24,767 | 127,787 |
| Number of Positive Examples | 27,834 | 147,778 |
| Number of Pair Train Split | 37,147 | 196,951 |
| Number of Pair Validation Split | 15,454 | 78,614 |
| Total Number of Pairs | 52,601 | 275,565 |
| Number of Topics | 1,600 | 1,600 |
| Number of Authors | 52,655 | 278,169 |

**Table 2.** Small dataset validation results (%).

| Model | AUC | C@1 | F0.5u | F1 Score | Overall |
|---|---|---|---|---|---|
| TFIDF Baseline | 0.789 | 0.731 | 0.699 | 0.7441 | 0.747 |
| Compress Baseline | 0.806 | 0.74 | 0.701 | 0.782 | *0.757* |
| $(CN)^2$ | 0.803 | 0.804 | 0.809 | 0.812 | 0.807 (+ 0.05) |
| Longformer | **0.898** | **0.897** | **0.914** | **0.898** | **0.902** (+ 0.145) |

Given these results, we chose the Longformer system trained on the large dataset as our submission for the TIRA evaluation system [22]. Unfortunately, as can be seen in Table 4, the model didn't generalize well to the test dataset.

## 5 Discussion

From the results we see that Longformer clearly outperforms both $(CN)^2$ and the PAN baselines on our own held-out validation set. We surmise that there are three reasons why the Longformer architecture won out over $(CN)^2$ on this held-out validation set:

1. It was pre-trained on about 6.5 billion tokens from multiple online sources.
2. Its deep architecture allows the model to learn more distant relationships of elements in the text than just n-grams. In contrast, $(CN)^2$ is limited to exploiting relationships of n-grams of size 1 through 5.
3. Because the grammar of fanfictions are relatively clean, the word-based Longformer is more suited to the task as it exploits semantic relationships between the words and the structure of the sentences. On the other hand, $(CN)^2$, focuses more on the syntactic features, thus we hypothesize that character-level features would be better in a dataset with shorter, more informal excerpts where words aren't necessarily written correctly.

As of the writing of this paper, we're still unsure as to why the Longformer model didn't generalize on the test set as can be seen in Table 4. We conjectured that our validation set may not have sufficiently exhibited "the significant shift in the relation between authors and fandoms" in the test set from those seen during training. To test

**Table 3.** Large dataset validation results (%).

| Model | AUC | C@1 | F0.5u | F1 Score | Overall |
|---|---|---|---|---|---|
| TFIDF Baseline | 0.779 | 0.723 | 0.691 | 0.759 | 0.738 |
| Compress Baseline | 0.766 | 0.707 | 0.674 | 0.753 | *0.725* |
| $(CN)^2$ | 0.851 | 0.852 | 0.849 | 0.858 | 0.852 (+ 0.127) |
| Longformer | **0.964** | **0.964** | **0.96** | **0.965** | **0.963** (+ 0.238) |

**Table 4.** Test dataset results (%).

| Model | AUC | C@1 | F0.5u | F1 Score | Overall |
|---|---|---|---|---|---|
| Longformer | 0.696 | 0.64 | 0.655 | 0.748 | 0.685 |

this hypothesis, we created a new dataset split such that the (author, fandom) pairs seen in the validation set are unseen in the training set. For example, if an author wrote in 5 fandoms, we used his/her writings from 4 fandoms in the train set and reserved the last for the validation set. This new validation set allowed us to test the most extreme form of (author, fandom) shifts where there is no overlap between (author, fandom) pairs between training and validation sets. Using this new train/validation data split, the Longformer model achieved an overall score of 93.6% on the validation set. While this score is less than that achieved in our previous validation set, it is still high, and thus fandom shift does not fully explain our Longformer model's failure to generalize on the official test set.

## 6 Conclusion

In this paper, we described our approach to the authorship verification task in the PAN @ CLEF 2020 challenge based on using neural networks for learning discriminative features from the text as well as the fandom from which the text derives. We compared two different neural network architectures. The first architecture, $(CN)^2$, uses a convolutional stack followed by a recursive self-attention stack to simultaneously learn useful character n-grams and their combinations that are most useful for determining whether the excerpt pair is from the same author. The second architecture leverages the state-of-the-art text features learned by the Longformer model, pre-trained on text with about 6.5 billion words, and we fine-tune the last two encoder stacks to learn useful features for authorship verification. By leveraging a powerful text modeling architecture like the Longformer, we sought to investigate the effectiveness of a transfer learning approach that has shown great results for other NLP tasks. Both systems learn a separate set of embeddings for the fandoms of both fanfic excerpts. The text based features and the fandom features are concatenated and used to predict authorship and fandom correspondences. We used a multi-task loss function to simultaneously optimize for both authorship verification and topic correspondence, allowing it to learn useful fandom features for author verification indirectly.

For our validation testing, we partitioned each of the PAN provided "small" and "large" training sets into our own training and validation sets. We evaluated our $(CN)^2$

and Longformer systems along with the two simple PAN baseline systems. Both of our systems outperformed each baseline, but our Longformer system was the winner by a wide margin, attaining a 0.963 overall verification score which is a 32.8% relative improvement over the best baseline system. Thus, we chose to submit our Longformer model trained on the "large" training set as our submission system for PAN. Unfortunately, this system failed to generalize on the official PAN test set, only attaining a 0.685 overall score and failing to outperform the baseline systems. Without access to the PAN test set, we have started to diagnose the source of the generalization failure. An initial experiment on a new train/validation split where (author, fandom) pairs do not overlap between train and validation sets has shown that an extreme shift in (author, fandom) pairs may not be the most significant cause of performance degradation. Our future work will further investigate this issue and seek to improve the Longformer system's generalization performance.

## 7 Acknowledgment

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This document may contain research results that are experimental in nature, and neither the United States Government, any agency thereof, Lawrence Livermore National Security, LLC, nor any of their respective employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply an endorsement or recommendation by the U.S. Government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily reflect those of the U.S. Government or Lawrence Livermore National Security, LLC and will not be used for advertising or product endorsement purposes.

## References

1. PAN: A series of scientific events and shared tasks on digital text forensics and stylometry. https://pan.webis.de/, accessed: 2020-07-03
2. Andrews, N., Bishop, M.: Learning invariant representations of social media users. ArXiv **abs/1910.04979** (2019)
3. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. ArXiv **abs/2004.05150** (2020)
4. Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Generalizing unmasking for short texts. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 654–659 (2019)
5. Bischoff, S., Deckers, N., Schliebs, M., Thies, B., Hagen, M., Stamatatos, E., Stein, B., Potthast, M.: The Importance of Suppressing Domain Style in Authorship Analysis. CoRR **abs/2005.14714** (May 2020), https://arxiv.org/abs/2005.14714
6. Bobicev, V.: Authorship detection with PPM notebook for PAN at CLEF 2013 (2013)

7. Boenninghoff, B., Nickel, R.M., Zeiler, S., Kolossa, D.: Similarity learning for authorship verification in social media. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2457–2461. IEEE (2019)
8. Brown, T.B., Mann, B.P., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krüger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E.J., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. ArXiv **abs/2005.14165** (2020)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv **abs/1810.04805** (2018)
10. Hitschler, J., van den Berg, E., Rehbein, I.: Authorship attribution with convolutional neural networks and POS-eliding. In: Proceedings of the Workshop on Stylistic Variation. pp. 53–58. Copenhagen, Denmark (2017)
11. Hosseinia, M., Mukherjee, A.: Experiments with neural networks for small and large scale authorship verification. ArXiv **abs/1803.06456** (2018)
12. Kestemont, M., Manjavacas, E., Markov, I., Bevendorff, J., Wiegmann, M., Stamatatos, E., Potthast, M., Stein, B.: Overview of the Cross-Domain Authorship Verification Task at PAN 2020. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)
13. Kestemont, M., Tschuggnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the author identification task at PAN-2018: cross-domain authorship attribution and style change detection. In: Cappellato, L., Ferro, N., Nie, J., Soulier, L. (eds.) Working Notes of CLEF 2018 - Conference and Labs of the Evaluation (2018)
14. Khmelev, D., Teahan, W.: A repetition based measure for verification of text collections and for text categorization. In: ACM SIGIR 2003. pp. 104–110 (2003). https://doi.org/10.1145/860435.860456
15. Kim, Y.: Convolutional neural networks for sentence classification. ArXiv **abs/1408.5882** (2014)
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. ArXiv **abs/1907.11692** (2019)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119 (2013)
18. Milli, S., Bamman, D.: Beyond canonical texts: A computational analysis of fanfiction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2048–2053. Austin, Texas (2016)
19. Peñas, A., Rodrigo, A.: A simple measure to assess non-response. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (2011)
20. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)
21. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
22. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World. Springer (Sep 2019)

23. Ruder, S., Ghaffari, P., Breslin, J.G.: Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. ArXiv **abs/1609.06686** (2016)
24. Shrestha, P., Sierra, S., González, F.A., y Gómez, M.M., Rosso, P., Solorio, T.: Convolutional neural networks for authorship attribution of short texts. In: EACL (2017)
25. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology **60**(3), 538–556 (2009)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. ArXiv **abs/1706.03762** (2017)
27. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing. ArXiv **abs/1910.03771** (2019)