CEUR-WS.org/Vol-3180/paper-215.pdf

Text-to-Text Transformer in Authorship Verification Via Stylistic and Semantical Analysis

Notebook for PAN at CLEF 2022

Maryam Najafi*[1], Ehsan Tavan*[1]

[1]*NLP Department, Part AI Research Center, Tehran, Iran*

Abstract

Authorship verification has gained much attention in recent years, due to the emphasis placed on PAN@CLEF shared tasks.In authorship verification, linguistic patterns are analyzed to reveal information about the author of two or more texts in order to determine if they are written by the same author. We describe in this paper our authorship verification submission system and the deep neural network approach that will allow us to learn the stylistic and semantic features of authors in the contributors to the PAN@CLEF 2022 event [1], [2], [3]. The system uses the T5 language model as a base embedding layer, followed by CNN and an attention mechanism to extract local and contextual features. As a result of studying multiple language models and deep architectures, we obtained an accuracy of 91.79% on our test dataset which was manually created from a PAN-provided dataset. However, on the official PAN test set, our system obtained a 58.7% overall score.

Keywords

authorship verification, language models, T5, Convolutional Neural Networks (CNN), stylistic features,

1. Introduction

Authorship Verification (AV) is a branch in digital text forensics that deals with comparing the stylistic, and linguistic patterns of two or more texts in order to determine whether they were written by the same author. In other words, the question of whether a documents were written by same author is commonly called AV. The digital library, online journalism, and social networks provide access to an incredible amount of digital texts. Social media plays an integral role in expanding access to AV. In various settings, it is important to verify document authorship automatically.

Researchers, for instance, are judged and compared according to the impact and quantity of their publications, and public figures are exposed by their posts on social media platforms. Massive amounts of textual data are being uploaded to the Internet. Online crimes are rising along with textual data. In order to reduce the problems raised by the Internet, many researchers have turned to the authorship detection. AV is a type of authorship detection that verifies that a document is written by the author by determining the authorship information of the

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2021, Bologna, Italy

maryam.najafi@partdp.ai (M. Najafi*); ehsan.tavan@partdp.ai (E. Tavan*)

0000-0001-5025-2044 (M. Najafi*); 0000-0003-1262-8172 (E. Tavan*)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

*These authors contributed equally to this work document. In addition to its many applications it has also been used in other investigations such as phishing emails and plagiarism detection. Personality traits such as the author's text, genre, temperament, sentiment, native language, gender can be determined by stylistic features. In other words, AV is the process of determining whether documents have been written by the same author.

The accuracy of AV majorly depends on the features that are used for distinguishing the style of writing followed in the documents. In the previous works of AV, the researchers proposed various types of stylistic features to distinguish the author's writing style. The researchers analyzed that the performance of AV was poor when the stylistic features were used alone in the experiment.

A variety of successful technical approaches have been proposed for this task, many of which are based on traditional linguistic features, which include spelling mistakes, grammatical inconsistencies, and stylistic features to distinguish the author's writing style. These features are well suited to long documents, such as books and novels. AV accuracy can largely be attributed to the features that are used to define the style of writing used in documents. The lack of feature extraction by traditional approaches becomes apparent when dealing with short messages and datasets such as tweets and social media posts. So, a disadvantage of ML is that its reliability is greatly compromised when it comes to short and topically diverse social media texts. on the other hand, ML algorithms traditionally relied on so-called stylometric-features [4].

As opposed to stylometric-feature-based systems, several papers have recently integrated the feature extraction task into a deep learning framework. Generally, author-specific writing styles also depend on the form of the text, e.g. whether it's a blackmail note, an Amazon review, a tweet, or a WhatsApp message.

Text-To-Text Transfer Transformer (T5) architecture [5] is shown to exhibit high performance in various Natural Language Processing (NLP) applications. The idea behind this study was to extract the semantic context of embedded text using a language model. In order to do this, we employed the last hidden state of T5-large. In addition to identifying the semantic context of the author, the purpose of this study is to identify stylistic, grammatical, and writing style characteristics of the author. In order to accomplish this, we extracted Part of Speech tagging (POS), emoji, punctuation, author-specific and topic-specific information from the text and provided each as a separate feature to the model. So, we can obtain both semantic and context information as well as stylometric features of the author.

We present a simple and effective approach to AV for similarity learning that significantly improves the performance on the dataset provided by the PAN@CLEF [1], [2] organization. We investigate similarities in the writing styles for two different texts with authors and get the reasonable result of 91.79% from our own-created test set from the PAN@CLEF[1] official dataset. Our code is available at GitHub[2] for researchers.

The remaining of this paper is organized as follows: Section 2 reviews related work. Section 3 describes both tasks and the provided dataset. Section 4 presents the theoretical background of the proposed neural model. Experiments and results are presented in Section 6. Section 7 contains paper conclusions.

[1] https://pan.webis.de/clef22/pan22-web/author-identification.html

[2] https://github.com/MarSanTeam/Authorship_Verification

2. Background

The AV using linguistic analysis identifies whether two or more texts were written by the same author based on the similarities between their language patterns. In this section, we present some of the approaches discussed in AV.

This paper's approach is based on a hierarchical recurrent Siamese neural network (HRSN). A recurrent neural network (RNN) topology is said to automate the extraction of sensible and context-independent features. Using a similarity analysis, it was possible to draw reasonable conclusions about unknown authors' writing styles [6]. Writing well involves finding the right words to convey your message. Language analysis of the internal attention weights of the network in [4] shows that the proposed method can indeed latch onto some traditional linguistic categories. In GLAD there was a framework for AV in four languages of Dutch, English, Spanish, and Greek [7]. Both known and unknown documents are used to train a binary linear classifier. These features included character ngrams, lexical overlaps, visual text properties, and compression measures.

In this field, Language Models (LMs) and Graph Neural Networks (GNNs) are commonly used. LG4AV claim that incorporating a LMs and GNN eliminates the need to manually extract features, and allows for the validation of relationships between authors [8].

In PAN@CLEF 2014, Using a graph representation that captures a syntactic sequence of texts and a graph similarity measure, [9] evaluates the similarity between an unknown document and the known documents. An unknown document can be classified as written by the same author if the majority of comparisons exceed a predetermined threshold. In [10] there was a use of content-based features in experiments. By using the term weight measure, the researcher computed the importance of each term, and these weights were used to calculate the document weight. To verify the test document, document weights of training and test documents were compared. [4] propose Siamese network on the large-scale corpus of short Amazon reviews and an analysis of the internal attention weights of the network shows that the proposed model outperforms state-of-the-art approaches that were built upon stylometric features.

3. Data Description

The AV task dataset consists of 12264 samples in English. Each sample contains two texts belonging to different Discourse Types (DT). Each sample also has a tag that indicates whether the two texts are written by the same author. The samples are from the Aston 100 Idiolects Corpus in English covering the following DTs: essays, emails, text messages, and business memos. In order to train the deep learning model, we considered 70% of this dataset as train data and 15% as evaluation data. We also used the remaining 15% of the data as test data to evaluate the use of various features on the model performance.

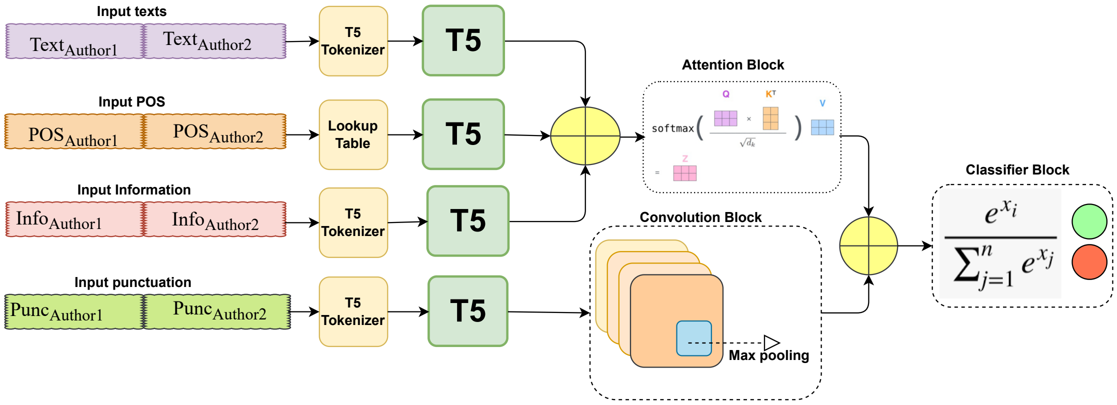

Figure 1: The Proposed model architecture. In this model, the T5 language model is used as feature representation vector. We use T5 with sharing weights between all features. The input text is the input tokens of text pairs. The input PoS is the PoS tags of input texts. The input information is the author-specific and topic-specific information. The input punctuation is the sequence of input punctuations.

4. System Overview

4.1. Embedding Layer

The embedding module is used to convert input tokens to representation vectors. The embedding of words is fundamental to building a model based on deep learning architecture. As a result, in this module, we have used T5 as the embedding layer to obtain a suitable representation vector. Several recent studies have demonstrated that neural language models trained on unstructured text can implicitly store and retrieve contextualized semantic information. Due to the prominent role of the author's writing structure, the syntactic embedding of the author has also been implemented in this study in addition to semantic embedding.

Along with token representation an innovative PoS structure based on a T5 language model embedding is presented in the paper to effectively encode syntactic writing style. Each word was tagged with its corresponding PoS tag. Since the raw texts were tagged with their corresponding PoS tag, each PoS tag was indexed independently and fed into a T5 language model. The output of T5 contains contextualized and dense embeddings of PoS tags. Embeddings can be used to properly capture syntactic writing style.

The architecture incorporates two other stylistic features, punctuation and the author-specific information, as well as PoS in order to capture an author's stylometric information. The T5 receives all three of these features as well as the original text, as four inputs.

The direct feeding of available texts into a pre-trained transformer architecture eliminates the need for any hand-crafted stylometric features, which are of no use in scenarios where the writing style is, at least partially, standardized.

4.1.1. Semantic Embedding

An essential step in creating NLP models is choosing an appropriate embedding vector. In this research, the T5 Encoder module was implemented as an embedding layer. The main part of this model is the use of the T5 module for all extracted features from the dataset. we use pairs of first and second text as an input of the T5 language model and get embedding from the last hidden state of T5.

4.1.2. Syntactic Embedding

there are three features that were incorporated in this paper. the details will discuss in following sections.

PoS: Some researchers employed NLP tools to extract more complex syntactic and semantic features. The most popular of these features is PoS. The PoS has been extensively researched for its effectiveness in AV [11, 12, 13, 14].

As a PoS tagger, we use the NLTK library [15] and utilize a set of 40 PoS tags. We convert the corresponding word in a sentence into PoS tag, then using a lookup table $T_{pos} \in R^{|Z|}$ we convert each PoS tag into index to use as T5 encoder input.

Both the PoS tags of the input texts were then fed into the T5 language model to obtain dense vector relationships. A benefit of this type of embedding vector is its fixed size of the syntactic embedding lookup table, which makes it less susceptible to out-of-vocabulary problems.

Punctuation and Emoji: A step toward interpretable AV is based on punctuation marks as a syntactic feature that consider grammatical structures and is, therefore, independent of content and topic. We also use emojis used in the text in combination with punctuations. Using punctuation and emoji ngrams can provide richer features to the model.

After extracting the sequence of punctuations and emojis from the input text, one-gram, two-gram, and three-gram of punctuations and emojis can be extracted using Convolutional Neural Networks (CNN) architecture. Our method of identifying punctuation and emoji ngrams allows us to identify the user's punctuation and emoji habits which reflect their writing style. Using these features, which can be called author punctuation writing style, can be very helpful to the model to achieve higher accuracy in AV task

Author-specific and topic-specific information: There are several special tokens used in this dataset. Special tokens include the <new> tag for indicating message boundaries, the <nl> tag for indicating new lines within a text, and Author-specific and topic-specific information, such as Named Entities Recognition (NER), were also used. Location, subject, job_title, day, and others are all included in this author-specific information. As the use of new lines, the number of times they are used, and how to express some information, such as named entities, can differ between people, using these features can aid the model in the AV task. By extracting different features from topic-specific and author-specific information, the deep learning model can perform better in AV tasks.

This section introduces features that can be used to extract author information. As can be seen from Figure 1, each of these features obtained from the data is then given to the T5 encoder, separately so that they can be incorporated into the next layer.

4.2. CNN Module

A CNN architecture was designed to extract the punctuation and emoji ngrams from their sequence. Using input text, CNN can extract local context and ngram features. From a sequence of punctuations and emojis, one-gram, two-gram, and three-gram features have been extracted using one-, two-, and three-dimensional convolution filters. Moreover, convolution filters are followed by a max-pool layer to extract rich features. The final ngram features are derived from combining the features extracted from the max-pool layer.

4.3. Attention Module

Scaled dot-product attention[16] uses different weights to extract rich stylomeric features from the text. Because attention is focused on the tokens that best convey the style of a writer, using the attention mechanism is a viable method. The writing style of the author is determined by focusing on the most crucial words by considering the occurrences of these (POS and author-specific information) tokens by feeding them to attention.

Many tasks, including question answering, machine translation, speech recognition, and image captioning, have been successfully completed by attention mechanism. By assigning different weights to each token, we are able to extract rich features using a scaled dot-product attention system.

Tokens are assigned weights based on their significance and importance in defining text stylometry, regardless of their distance from each other. Therefore, the importance and relevance of tokens can be determined in identifying the writing style of an author. Attention comprises the following elements:

$$\begin{gathered} W_i^Q, W_i^K, W_i^V \in R^{d_{model} \times d_k} \\ Q = XW_Q, K = XW_K, V = XW_V \\ Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \end{gathered} \tag{1}$$

There are three trainable parameters: W_Q, W_K, and W_V. In order to create three matrices (Q, K, and V), input X is multiplied with matrices Q, K, and V. Dot-products between Q and K are divided by $\sqrt{d_k}$ to prevent them from becoming too large.

4.4. Prediction Module

The first step was to establish a fully connected layer, for predicting syntactic and semantic similarities in two texts. The general representation is then constructed by using a max-pooling layer based on the same dimensions of multiple tokens. Equation 2 formulates the max-pooling modules.

$$Z = Max([h_1, ..., h_l]) \tag{2}$$

The softmax classification method is used to determine the probabilities of the labels. Based on input, the module creates probabilities of distributions in terms of softmax classifiers. A softmax classifier is used to predict a label $\hat{y}$ for an input sequence from a set of discrete classes (to be same or not to be).The softmax classifier takes R as input:

$$\boldsymbol{P}(\boldsymbol{y} \mid \boldsymbol{Z}) = \boldsymbol{softmax}(\boldsymbol{WR} + \boldsymbol{b}) \tag{3}$$

$$\widehat{\boldsymbol{y}} = argmax\boldsymbol{P}(\boldsymbol{y} \mid \boldsymbol{Z}) \tag{4}$$

5. Experimental Setup

Model implementation was done in PyTorch on Nvidia V100 GPUs. Training was done through 100 epochs. To train the network, the AdamW optimizer with a learning rate of 2e-5 is used. Early stopping in max mode with 7 epochs of patience ensures a high validation accuracy. The training batch size is set to 8. The T5 tokenizer is limited to a maximum length of 350 tokens. The CNN filter sizes are set to 1, 2, 3 to extract ngram features. Other parameters are initialized at random.

6. Results

According to the previous section, our first challenge was to find the suitable embedded contexts of the tokens through language models. Consequently, we first analyzed the quality of the various language models on the data, and after reviewing BERT, RoBERTa, and T5, we concluded that the T5-large model is the most accurate of these language models.

Table 1 shows the results of the various language models. The results of our tests led us to use the T5-Large language model in subsequent experiments with different features.

Table 1
The results of evaluating different language models

Model	Valid ACC	Valid F1-Score	Test ACC	Test F1-score
BERT-Base	61.23 %	61.09 %	61.02 %	61.11 %
T5-Base	71.09 %	71.19 %	71.23 %	71.33 %
BERT-Large	64.49 %	63.92 %	63.50 %	63.99 %
RoBERTa-Large	63.03 %	62.82 %	62.48 %	62.76 %
T5-Large	82.99 %	82.88 %	82.39 %	82.46 %

Following the determination of the appropriate embedding layer, we had to examine properties that would allow us to determine the semantic and grammatical differences and similarities between the two texts. This information can prove to be extremely valuable to a model when determining the authorship of two texts. As mentioned, extracting context-based and semantic features can be very helpful in AV. To capture context and semantic features, we've used a T5-based embedding layer. Based on the structure and writing of an author, we used punctuation based on ngram, author-specific, and topic-specific information, and PoS to extract syntactic features. The ngram-based punctuation feature was calculated using CNN to ensure superior accuracy. The results of different experiments comparing the validation data and test data are shown in Table 2.

As can be seen, the introduction of new features led to improvements over the original text. From the Table 2, it is evident that incorporating punctuation, author-specific, and topic-specific information has increased the F1-score of the model from 82.46% to 87.52%. Following the addition of the PoS tags to extract the syntactic features used by the author, the F1-score of the model increased from 87.52% to 89.93%.

Finally, using the attention mechanism, we sought to extract the meaning of tokens and particular relationships between them, so that we could further use the meaning of the specific tokens as well as particular relationships within each author's text. Due to the high ability of the attention mechanism to find and prioritize important tokens, as well as to discern the relation between tokens, the F1-score of the model has increased from 89.93% to 91.72% on the test data.

Table 2
The results of adding introduced features to base model

Model	Val ACC	Val F1-Score	Test ACC	Test F1-score
T5-Large	82.99 %	82.88 %	82.39 %	82.46 %
T5+Punc	87.16 %	87.33 %	86.50 %	86.19 %
T5+Punc(CNN)	90.05 %	90.09 %	87.12 %	87.17 %
T5+Punc(CNN)+POS	88.36 %	88.32 %	87.17 %	87.09 %
T5+Punc(CNN)+Information	88.85 %	88.72 %	87.60 %	87.52 %
T5+POS+Punc(CNN)+Information	91.35 %	91.10 %	89.67 %	89.93 %
T5+Punc(CNN)+POS+Information(Attention)	91.73 %	91.47 %	91.79 %	91.72 %

In order to evaluate the performance of our proposed model, we used the evaluation platform provided by PAN, which includes the following metrics:

- AUC: the conventional area under the curve score.
- c@1: rewards systems that leave complicated problems unanswered [17].
- F_0.5u: focus on deciding same-author cases correctly [18].
- F1-score: harmonic way of combining the precision, and recall of the model [19].
- Brier: Brier Score evaluates the accuracy of probabilistic predictions [20].

Based on the hidden test set, table 3 demonstrates the performance of the proposed model. This model was evaluated on the TIRA environment for PAN@CLEF 2022.

Table 3
The result of proposed model on the hidden test set.

Model	AUC	c@1	F_0.5u	F1-score	Brier	Overall
Proposed model	59.8 %	57.1 %	57.1 %	57.6 %	61.8 %	58.7 %

7. Conclusion

Our research proposes a model using semantic, grammatical, and stylometric (i.e. punctuation ngrams and topic-specific information) to predict sameness of two text excerpt author. The T5 language model is used to convert these features into representation vectors.

Our CNN neural network extracts ngram features from punctuation sequences. The attention module is also employed to extract the most important features and to determine the relationships between the tokens. We conducted many experiments to evaluate the performance of the proposed model with these newly introduced features. As demonstrated by the experimental results, combining semantic-based, grammatically-based, and writing style features with the proposed architecture provides a reasonable range of results for AV.

References

[1] Efstathios Stamatatos and Mike Kestemont and Krzysztof Kredens and Piotr Pezik and Annina Heini and Janek Bevendorff and Martin Potthast and Benno Stein, Overview of the Authorship Verification Task at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR Workshop Proceedings, 2022.

[2] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Style Change Detection Task at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR Workshop Proceedings, 2022.

[3] 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.

[4] B. Boenninghoff, S. Hessler, D. Kolossa, R. M. Nickel, Explainable authorship verification in social media via attention-based similarity learning, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 36–45.

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv preprint arXiv:1910.10683 (2019).

[6] B. Boenninghoff, R. M. Nickel, S. Zeiler, D. Kolossa, Similarity learning for authorship verification in social media, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2457–2461.

[7] M. Hürlimann, B. Weck, E. van den Berg, S. Suster, M. Nissim, Glad: Groningen lightweight authorship detection., in: CLEF (Working Notes), 2015.

[8] M. Stubbemann, G. Stumme, Lg4av: Combining language models and graph neural networks for author verification, in: International Symposium on Intelligent Data Analysis, Springer, 2022, pp. 315–326.

[9] E. Castillo, O. Cervantes, D. Vilari, B. David, Author verification using a graph-based representation, International Journal of Computer Applications 123 (2015).

[10] P. Buddha Reddy, T. Murali Mohan, P. Vamsi Krishna Raja, T. Raghunadha Reddy, A novel approach for authorship verification, in: Data Engineering and Communication Technology, Springer, 2020, pp. 441–448.

[11] L. Ouyang, Y. Zhang, H. Liu, Y. Chen, Y. Wang, Gated pos-level language model for authorship verification, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4025–4031.

[12] J. Hitschler, E. Van Den Berg, I. Rehbein, Authorship attribution with convolutional neural networks and pos-eliding, in: Proceedings of the Workshop on Stylistic Variation (EMNLP 2017). September 8, 2017 Copenhagen, Denmark, The Association for Computational Linguistics, 2018, pp. 53–28.

[13] A. Kansal, Fake news detection using pos tagging and machine learning, Journal of Applied Security Research (2021) 1–16.

[14] D. Vilariño, D. Pinto, H. Gómez, S. León, E. Castillo, Lexical-syntactic and graph-based features for authorship verification, in: Proceedings of CLEF, 2013, pp. 282–302.

[15] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with

the natural language toolkit, " O'Reilly Media, Inc.", 2009.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[17] A. Peñas, A. Rodrigo, A simple measure to assess non-response (2011).

[18] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, 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, Volume 1 (Long and Short Papers), 2019, pp. 654–659.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.

[20] G. W. Brier, et al., Verification of forecasts expressed in terms of probability, Monthly weather review 78 (1950) 1–3.