Reproducibility and Robustness of Authorship Identification Approaches

Adam Karásek and Zuzana Nevěřilová

Natural Language Processing Centre
Faculty of Informatics
Botanická 68a, Brno, Czech Republic

Abstract. Authorship identification, framed as a classification task, assigns a digital text to a known author. State-of-the-art algorithms for this task often lack evaluation across diverse datasets. This paper re-implements and evaluates three approaches on three different datasets, exploring the robustness of algorithms on various text types (e.g., emails, articles, instant messages).
Not all the published methods are fully reproducible. However, reasonable parameters were selected if they were not part of the original paper. The evaluation of the ensemble model shows it is somewhat robust on different texts and different numbers of potential authors.

Keywords: authorship identification, evaluation, reproducibility

1 Introduction

Authorship identification is a classification task that assigns a human-written digital text to an author from a known set of authors. There are many different state-of-the-art algorithms for classifying authors of text based on numerous classification algorithms and text processing techniques. However, papers proposing these solutions often provide their evaluation only on one selected dataset. The research question is how robust the distinct algorithms on datasets of different types of text (e.g., emails, articles, or instant messages) are.

In this paper, we re-implemented three different approaches and evaluated them on three different datasets. Apart from robustness, we examined the reproducibility of the published papers. In Section 2, we describe the selected approaches to authorship identification. Section 3 describes the datasets we selected for the evaluation. We aimed to select heterogeneous data in English. In Section 4, we describe our re-implementation. Section 5 describes the evaluation of the count vector ensemble model, and Section 6 draws conclusions about the robustness of the model.

2 Related Work

Authorship identification, also called authorship attribution in some literature[9], is part of a broader field of authorship analysis. There are two other tasks, as

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2023, pp. 57–65, 2023.

stated in [10]. Authorship verification is a mechanism for deciding whether a specific individual wrote an anonymous text. Authorship characterization presumes the author's characteristics, such as gender, age, social background, etc.

Traditionally, identifying the author of an anonymous text was done using stylometric features. Over the years, over 1000 stylometric features of different types, such as lexical, syntactic, structural, content-specific, and idiosyncratic. Nonetheless, there is no consensus on which features or set of features are most helpful in identifying the author of a given text. Different stylometric features can be suitable based on the type and properties of the examined text. Measured features are, for instance, average word length, punctuation rate, occurrence of special characters, etc. [1]

2.1 Count Vector Ensemble Model

The first model we selected for our experiment was introduced by [2] using a count vector for feature extraction and an ensemble of three classification models as a classifier. The count vector calculates the frequency of each word, called *term frequency*, of the input text. This approach measures how many times an author uses specific words. Therefore, the classification model can recognize the author based on their use of words. The authors of [2] used random forest, extreme gradient boosting (xgboost, XGB), and multilayer perception (MLP) as a classification model in the ensemble. With this setup, they have reached 97 % accuracy on 10 authors and 79 % accuracy on 20 authors on a news articles dataset. The dataset was composed from over 140 000 news articles from 15 american news websites.

2.2 Email Detective

As a second model for the experiment, we chose a neural network proposed by [7] with two inputs. The first input processes text using the *word2vec* method for text embedding. Word2vec transforms words into high-dimensional vectors. These vectors reflect the words' meaning so that semantically similar words are close to each other in the vector space [8].

Unconventionally, the authors of Email Detective use the same method to embed characters (excluding spaces), not words. Next, the embedded characters were input into a BiLSTM layer. After processing the text input, it was concatenated with 10 stylometric values gathered from the email header. The full set of text features is afterward classified with a dense layer.

The authors of [7] used the Enron dataset to evaluate their model. Email Detective achieved an accuracy of 98.9 % for 10, 92.9 % for 25, and 89.5 % for 50 authors.

2.3 BertAA

The third implemented algorithm is a transformer-based classification neural network introduced by [6]. Their model consists of a pre-trained BERT fine-tuned on an authorship attribution dataset with a dense layer for classification.

They used Enron emails, IMDb Authorship Attribution, and Blog Authorship Attribution corpus for evaluating the model. For the Enron dataset, the model achieved an accuracy of 99.95 % for 5 authors, 99.1 % for 10 authors, and 98.7 % for 25 authors. The IMDb dataset's accuracy reached 99.6 % for 5 authors, 98.1 % for 10 authors, and 93.2 % for 25 authors. Furthermore, for the Blog dataset, the model attained an accuracy of 61.3 % for 5 authors, 65.4 % for 10 authors, and 65.3 % for 25 authors.

3 Datasets

We selected datasets where authorship is part of the annotation. At the same time, we wanted the evaluation data to be as diverse as possible. We therefore selected emails, social media posts, and news texts.

3.1 Enron Emails

The Enron emails dataset is a publicly available dataset of emails from about 150 authors published by the Federal Energy Regulatory Commission as part of an investigation of Enron Corporation. We used a version of the dataset partially prepossessed by the CMU School of Computer Science. The dataset contains about 500,000 emails from the management of Enron. [5]. All of the emails were sent between 1997 and 2002. [11]

As part of gathering all the email texts into one CSV[1] file, we separated the email body written by the author from the rest of the email file. The start of the email body is deterministic and, therefore, easy to find. The last line of the email header always starts with `"X-FileName: "`, and the next line is already the email body. The end of the email body written by the authors was harder to find. Often, email file contains forwarded messages. This occurs in the dataset in two ways. The first is standardized, probably done by an email client, in which the other author's text starts with one of the following:

- `"--- Forwarded by"`
- `"--- Original Message"`
- `"--- Original Appointment"`

Therefore, we sliced away the part of an email body starting with the phrases listed above.

The second form of another author's text is also resending someone else's email, but probably due to copying and pasting without the exact structure. We noticed that these parts of text often contain phrases such as `"To: "`, `"From: "`, and `"Send by: "`. So, we cut out the part starting with these phrases. We also removed the signatures. The texts in the dataset have an average of 400 to 500 characters.

[1] Comma-separated values

3.2 Techcrunch Articles

The second dataset comprises articles from `techcrunch.com`, an online newspaper focused primarily on startups and tech companies. We obtained the dataset on Kaggle [4]. There was no need to preprocess the dataset other than deleting all unnecessary columns so that we kept only the author and text columns. The texts in the dataset have an average of 3000 to 3500 characters.

3.3 Telegram Messages

The last dataset is gathered from the biggest Telegram group focused on cryptocurrencies. Data were published by Kaggle user Anton [3]. We chose the "OKEx official group" as the biggest of the five datasets in Telegram. The dataset was published in JSON format, but we transformed it into a CSV file for easier processing and standardization with other datasets and deleted all redundant information, keeping only the author and text. This dataset is characterized by frequent usage of emojis. No other preprocessing was needed. The texts in the dataset have an average of 200 to 220 characters.

3.4 Training and Evaluation Subsets

We constrained all datasets in the following way:

1. The task complexity increases with the number of potential authors. We, therefore, selected k authors with the largest number of documents.
2. Every document from the document set has to be at least 100 characters long. This constraint removes documents (e.g., emails and instant messages) containing only one sentence (such as "I'll be there.") for which it is impossible to assign an author.
3. Since the dataset should be balanced, we select l random documents written by each selected author.

We created three experiment sets for each dataset with $k \in \{5, 10, 25\}$. Parameter l is different for each dataset and parameter k. This is done to ensure the maximal size of experiment sets while keeping them balanced. The number of documents per author is shown in the table below.

Table 1: Number of documents per author in experiment set

Dataset	$k = 5$	$k = 10$	$k = 25$
Enron	$l = 4000$	$l = 2000$	$l = 800$
Telegram	$l = 1000$	$l = 650$	$l = 470$
Techcrunch	$l = 2500$	$l = 1200$	$l = 250$

4 Algorithm implementation

We compare three algorithms because they are relatively recent and reported a high accuracy rate. The re-implementation in Python is available at GitHub[2].

4.1 Count Vector Ensemble

We implemented the count vector ensemble model described in [2]. We created a class to encapsulate the xgboost classifier, multilayer perceptron, and random forest classifier. We separated 10 % of the learning dataset into a validation dataset and used the rest as a training dataset. Then, we transformed the text into a count vector.

[2] did not specify the number of layers and nodes in the MLP classifier. Therefore, we experimented with different settings to attain the best results. We implemented the MLP with three hidden dense layers with 4096, 2048, and 1024 nodes, respectively, and a ReLU activation in a forward direction with a dropout layer set to 0.5 after each hidden layer, including L2 regularization in each dense layer. The count vector size determined the input layer dimension. The output layer is a softmax with the number of nodes = number of authors.

The output shape is a one-hot encoding for the MLP model and a one-dimensional token for random forest and xgboost classifier.

The rest of the hyperparameters were described in the original paper; therefore, we used the values provided by [2]. We set the loss function to categorical cross-entropy, and we used Adam as an optimizer with a learning rate of 0.0001. To find out the right number of epochs, we applied early stopping.

For the random forest classifier, we set the number of trees to 100, minimum sample split to 2, minimum samples in leaf to 1, bootstrap to true, and criterion to Gini.

We set the parameters of the xgboost classifier as follows: eta to 0.3, min-child-weight to 1, max-depth to 6, and scale-pos-weight to 1.

We trained all three classifiers on the same learning dataset. For ensemble prediction, we used each classifier to predict an author, then calculated the average of the three output vectors and determined the most probable author. When there were two or more authors with the same probability, we chose randomly one of them as the predicted author.

4.2 Email Detective

We implemented the Email Detective algorithm without the email header stylometric features, considering no such data are available for datasets other than emails. We set the specification as described in [7] to an extent to which model parameters were specified. The authors of [7] calculated the word2vec vector representation with dimensions set to 256; the iter and window parameters set to 5. The BiLSTM layer was set to a full output sequence; a maxpool layer reduces

[2] `https://github.com/karasekadam/authorship_identification`

Table 2: Ensemble model experiment results for Techcrunch dataset

Techcrunch	$k = 5$	$k = 10$	$k = 25$
Ensemble	0.9624	0.9275	0.7568
Random Forest	0.904	0.8517	0.6096
XGB classifier	0.9616	0.915	0.7536
MLP	0.9728	0.943	0.7504
Training time	894s	1229s	1142s

the output size. After a 0.5 dropout, a softmax layer follows. The classification part consists of a dense layer with 256 nodes, a dropout layer set to 0.5, and a softmax layer as output.

There were parameters not detailed in the original paper. We decided to use global max pooling 1D to reduce batch dimension for the max-pooling layer. The authors of [7] did not specify the length of the input text, so we experimented with different setups and decided to limit the input length to 10000 characters due to a need for a faster training time and limited memory.

4.3 BertAA

We used the TensorFlow hub to obtain the pre-trained BERT model and downloaded `bert_en_cased_L-12_H-768_A-12`. We transformed the input text to the pre-trained vector representation that was part of the downloaded model. A dense layer classified the output of the BERT model with a softmax activation function. All parameters were set to trainable, and we ran the experiment with early stopping.

5 Evaluation

We evaluated the model using *accuracy* and measured the training time. We performed a 72/18/10 split into training/validation/test.

We used a computer with Ubuntu 22.04 LTS, two Tesla T4 GPUs, 65GB RAM, and 32 32-core Intel Xeon Silver 4110 CPU for the experiment.

The count vector's feature size in the Telegram dataset's ensemble model was 4500 to 6000, with numbers in the upper of the interval with more authors. For the Techcrunch dataset, the count vector dimension was between 44000 and 55000. Furthermore, the count vector length for the Enron dataset was around 26000. The count vector size represents the training corpus's unique words without English stopwords. As we expected, the size of the count vector was larger with more extensive datasets and longer texts.

6 Results

We only evaluated the count vector ensemble model described in Section 4.1. The experiment results for Techcrunch, Telegram, and Enron are shown in

Table 3: Ensemble model experiment results for Telegram dataset

Telegram	$k = 5$	$k = 10$	$k = 25$
Ensemble	0.34	0.2062	0.0896
Random Forest	0.344	0.2062	0.0902
XGB classifier	0.308	0.1985	0.0885
MLP	0.31	0.2338	0.0766
Training time	52s	78s	190s

Table 4: Ensemble model experiment results for Enron dataset

Enron	$k = 5$	$k = 10$	$k = 25$
Ensemble	0.9675	0.9339	0.8417
Random Forest	0.9625	0.9228	0.8171
XGB classifier	0.9395	0.8961	0.8142
MLP	0.968	0.9234	0.8057
Training time	902s	1323s	1591s

Tables 2, 3, and 4, respectively. The tables show results for different numbers of authors.

The BertAA and Email Detective experiment was not finished by the time this paper was written. All measured results will be presented at the Raslan Conference 2023.

It can be seen the ensemble robustness shows itself when used on a larger number of authors. With enough features, the MLP model outperformed the ensemble model on the Techcrunch dataset with 5 and 10 authors and the Enron dataset with 5 authors. Furthermore, the random forest algorithm is probably better at classifying data with fewer features as it outperformed other models on the Telegram dataset, which has several times lower feature space than the other two datasets.

A possible explanation for the Telegram dataset's drop in accuracy is the text's nature. The texts are mutually very similar in topic and style, so it is difficult to distinguish the author by the words they have used.

7 Conclusion and Future Work

The aim of this paper was to examine recent approaches to authorship identification. We selected three of them and three datasets. We re-implemented each approach and evaluated it on all three datasets to see how reproducible the original paper is and how robust the approach is.

The results published in [2] were not fully reproducible since the authors did not publish details about MLP classifier architecture. In addition, the evaluation dataset was different from ours. Despite these conditions, the evaluation on the three datasets has shown that the approach is somewhat robust. Mainly, MLP

contributes the most to high accuracy in the case of a smaller number of potential authors. On the other hand, random forests are more accurate with a higher number of authors. Apparently, the text length does not matter much, but the number of features does.

In the near future, we plan to evaluate the re-implementations of the Email Detective and BertAA. In the case of Email Detective, we cannot expect the same results since the re-implementation does not take the email header into consideration. At least, we could estimate how much the email header classification contributes to the overall accuracy.

Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Abbasi, A., Chen, H.: Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace. ACM Trans. Inf. Syst. **26**(2) (apr 2008). https://doi.org/10.1145/1344411.1344413, https://doi.org/10.1145/1344411.1344413
2. Abbasi, A., Javed, A.R., Iqbal, F., Jalil, Z., Gadekallu, T.R., Kryvinska, N.: Authorship identification using ensemble learning. Scientific Reports **12**(1) (2022). https://doi.org/10.1038/s41598-022-13690-4
3. Anton: Crypto telegram groups (Feb 2021), https://www.kaggle.com/datasets/aagghh/crypto-telegram-groups
4. Balbo, T.: Techcrunch posts compilation (Oct 2016), https://www.kaggle.com/datasets/thibalbo/techcrunch-posts-compilation
5. Cohen, W.W.: Enron email dataset (2015), https://www.cs.cmu.edu/~./enron/
6. Fabien, M., Villatoro-Tello, E., Motlicek, P., Parida, S.: BertAA : BERT fine-tuning for authorship attribution. In: Bhattacharyya, P., Sharma, D.M., Sangal, R. (eds.) Proceedings of the 17th International Conference on Natural Language Processing (ICON). pp. 127–137. NLP Association of India (NLPAI), Indian Institute of Technology Patna, Patna, India (Dec 2020), https://aclanthology.org/2020.icon-main.16
7. Fang, Y., Yang, Y., Huang, C.: EmailDetective: An Email Authorship Identification And Verification Model. The Computer Journal **63**(11), 1775–1787 (07 2020). https://doi.org/10.1093/comjnl/bxaa059, https://doi.org/10.1093/comjnl/bxaa059
8. Nicholson, C.V.: A beginner's guide to word2vec and neural word embeddings, https://wiki.pathmind.com/word2vec
9. Nirkhi, S., Dharaskar, R.: Comparative Study of Authorship Identification Techniques for Cyber Forensics Analysis. International Journal of Advanced Computer Science and Applications **4** (12 2013). https://doi.org/10.14569/IJACSA.2013.040505
10. Nirkhi, S., Dharaskar, R., Thakare, V.: Authorship Verification of Online Messages for Forensic Investigation. Procedia Computer Science **78**, 640–645 (2016). https://doi.org/https://doi.org/10.1016/j.procs.2016.02.111, https://www.sciencedirect.com/science/article/pii/S1877050916001137, 1st International Conference on Information Security & Privacy 2015

11. Shetty, J., Adibi, J.: The Enron Email Dataset Database Schema and Brief Statistical Report. Tech. rep., University of Southern California (01 2004)