Graph-based Authorship Identification and Portrait Sketching

Jingze Xue, Richard Ma, Yang Su[*]
Cornell Tech, Cornell University
jx288, ym428, ys724@cornell.edu

December 11, 2022

Abstract

Our paper proposes a new method for authorship identification that incorporates graph structures and contrastive learning techniques. Authorship identification (AID) is the process of identifying the author of a given text using the structure of the text and the author's writing style. It is usually treated as a text classification problem, which exhibits limitations when encountering real-world datasets with too many authors to classify. To overcome this issue, we used a method similar to contrastive learning, where the positive and negative pairs would be an article with its correct and incorrect author features, respectively. We further improve the model's performance by utilizing graph machine learning, which could capture the inherent structure and relationships of authors and articles. In the end, we increased the model's AUC value from 73% to 79% on a sampled subset from the Citation Network of DBLP.

1 Introduction

1.1 Motivation

Authorship identification is often categorized under text classification, which is not a best practice when faced with many authors and unknown authors. Hence we want to introduce our method by incorporating graph structures and the techniques of contrastive learning to better tackle the above obstacles.

1.2 Problem Setup

We want to contribute to the problem of authorship identification based on articles. We want to draw a rough portrait of the unknown author by the entity's properties (i.e., nationality, age range, domain specialization) without knowing the exact name/identity of the authors. So far, we have implemented the author's affiliation (universities, majors, research fields, etc.) into our model, which can tell whether a given author writes a given paper.

2 Background

2.1 Authorship Identification

Authorship identification (AID) [1] is the process of identifying the author of a text using the structure of the text and the words used. Typical applications include anti-fraud, intellectual property protection, forensics, plagiarism detection, etc.

AID is usually treated by a text classification task. In our work, however, we will show that a naive approach to treating it as text classification does not fulfill our task objective because the number of authors in the real-world data set is

[*]author names are listed in alphabetical order: Richard writes about the motivation, problem setup, and the main methods; Jingze writes about the experimental results and future work; while Yang did most of the code implementation.

more than ideal for the model to classify. Instead, we use contrastive learning to turn the task into a binary decision task, as we will show later in more detail.

2.2 Pre-trained Language Model

Bidirectional Encoder Representations from Transformers (BERT) [2] is an NLP model developed by Google. BERT is trained to understand the context of a word in a sentence rather than just processing words individually, like traditional NLP models.

BERT has been a popular framework since 2017 to perform large-scale language understanding. It can extract key information from the given texts. In our case, we use BERT as a summarizing tool for the articles, and a features extractor for authors' features (affiliation, age, gender, etc.). Then we use these features to do portrait matching and sketching, which we will discuss later in more detail.

2.3 Graph Machine Learning

Graph machine learning (GML) [3] uses graphical data structures to represent and process data. In GML, the data is represented as a network of nodes and edges, where the nodes represent entities and the edges represent the relationships between those entities. This allows the model to capture and use the inherent structure and relationships in the data. Graph machine learning has been used in various applications including recommendation systems, social network analysis, and fraud detection.

In our project, the nodes are the authors' and articles' features, and the edges are the connection between them. The model can better match the relationship between authors and their features by studying this connection.

2.4 Contrastive Learning

Contrastive learning [4] is a type of machine learning that involves training a model to learn the differences between two or more sets of data. This is typically done by presenting the model with pairs of data, where one pair member is the "positive" example, and the other is the "negative" example. Contrastive learning has been used in various applications, including image recognition and natural language processing.

In our project, we use contrastive learning to learn the difference between different articles and their authors' features (affiliation, age, etc.). A positive pair, in this case, would be an article with its correct author features, and a negative pair would be an article with incorrect author features. Ideally, the ratio of PN (positive pair vs. negative pair) in this case would be 1:f(n), where n is the sample number, and f(.) is a function involving n. However, due to both time and hardware limitations, the ratio we used was 1:k, where k is a relatively small number we chose for free. Furthermore, we did not use the contrastive loss as proposed in the original paper. Instead, we utilized a lighter version by creating a PN pair beforehand and the sampled ground truth as the label (see the PyG implementation). The limitations are discussed later in section 6.

3 Method

We plan to use three methods to gradually improve the authorship identification result.

3.1 Portrait classification (PC)

The idea of portrait classification (PC) is to classify the correct author of the article given an article. This is not feasible when there are too many authors to classify. For example, our dataset has 500k articles and 400k authors. The number of classes to classify gets asymptotically close to the data set size, so overfitting or gradient vanishing will be a serious issue, which makes logistic regression unable to converge sometimes. Another issue is that when we have a new author that has not been seen, we can not predict this author because they have been trained and are not in our options of classes yet. Because of these two disadvantages, we progress to our second method, portrait matching.

3.2 Portrait Matching (PM)

The idea of portrait matching (PM) is that, given an article and many author portraits, the model will calculate the similarity between these two and decide whether the author writes our given article.

This essentially changes the multi-class classification problem into a binary classification problem. The following arguments can overcome the two disadvantages of method 1. Firstly, this method is easy to train and can be done on new authors if we have some basic information about them. The method works by learning a hidden representation of the author's portrait, for instance, a sentence that is genuinely unique and can roughly describe the author, as well as the hidden representation of the articles. Once we have the two hidden representations, we can calculate their cosine similarity and predict the matches.

Our empirical result suggests that adding relevant author information can significantly improve our algorithm, which causes our accuracy scores to change from 47.48% to 81.36%, where the former only uses information like the authors' names. In contrast, the latter also incorporated the author's affiliation, their research field information, etc. There are still some disadvantages of this method because our dataset's description of the author is limited—garbage in, garbage out. The method accuracy can be greatly improved if we have better data about the description of the authors.

Further Improvement

We have a paper-author pair, which tells if an author writes this paper. If the pair is true, we call it a positive pair, meaning that this author writes this paper; otherwise, if the pair is false, we call it a negative pair, meaning that the author does not write this paper. The data we trained on to get 81.36% uses a 1:1 positive-to-negative pair ratio. When we modify the data so that the positive and negative pair ratio decreases to 1:3, we get a 90.26% accuracy.

This is because the model may assign a higher weight to negative examples since they are more in the dataset. To compare the model's performance more fairly, we use ROC Curve and the Area Under Curve (AUC) metric to illustrate our result better (**Figure 1**). The details of these models' setups are summarized in section 4.

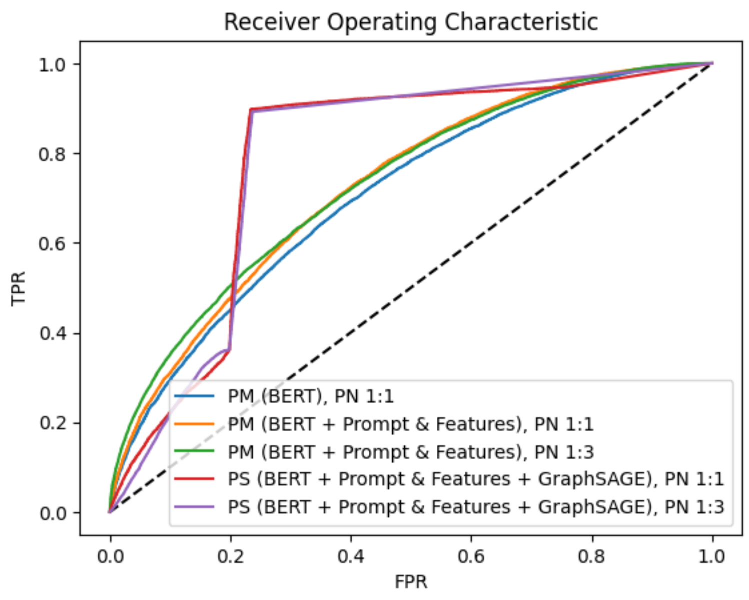

Figure 1: ROC Curve

3.3 Portrait Sketching (PS)

The idea of portrait sketching (PS) is that we do not need manual input of the author's information, and the model can predict whether an author writes this article by their properties. The dataset needs to be information-rich, meaning a clear category of the author's properties is explicitly encoded. With Graph Neural Network, we can predict the features of a certain author by learning the connection between their papers with other similar papers and their corresponding authors. We assume that if two authors write similar papers, their features are more likely to be similar (affiliation, age, field of study, etc.), which is what GNN is good at capturing.

4 Experiments

4.1 Baselines (BERT)

So far, we have done two experiments on portrait sketching: one without data scrapping and the other with data scrapping. The first is our baseline, where we do not incorporate author information into the model. The accuracy was around 47%, which is expected since we assumed author information was necessary. In the second one, we put author information into our model, including the text of the author's affiliation into the model. The result improved to 81%. We then further tested the necessity of a negative pair, where the author and the information do not match. With the presence of negative pairs, our accuracy was 81%; with double the amount of negative pairs, our accuracy went up to 90%. The details are summarized in **Table 1**.

Table 1: Test Result

General Setup for the Accuracy metric: 2e-5 learning rate, 2 epochs, 8 batch size
General Setup for the AUC metric: 2e-5 learning rate, 3 epochs, 128 batch size

Model	PN Ratio	Accuracy	AUC
PC (BERT)	1:1	N/A	N/A
PM (BERT)	1:1	47.48%	70.66%
PM (BERT + Prompt & Feature)	1:1	81.36%	72.26%
PM (BERT + Prompt & Feature)	1:3	90.26%	73.23%
PS (BERT + Prompt & Feature + GraphSAGE)	1:1	92.71%	77.49%
PS (BERT + Prompt & Feature + GraphSAGE)	1:3	93.24%	78.66%
PS (BERT + Prompt & Feature + GraphSAGE) Fine-tuned	1:3	**93.31%**	**79.08%**

4.2 BERT + N

The examples produced by BERT are illustrated in **Table 2**, where we showcase how the **prompt** and **feature** are used. Since the feature data are very limited in the original dataset, we web-scrapped the authors' affiliation through the DBLP API.

Table 2: Input Examples for Language Model

Model	Article Input	Author Input
BERT	Strategic Budgeting is a computer program for strategic budgeting and performance assessment...	J. Souza
BERT + Prompt	Feature selection methods for hidden Markov model-based speech recognition include...	The author's name is Donald P. Brown
BERT + Feature	Now fully updated for version 1.5, this comprehensive book-DVD combo starts with...	Fundación Universidad de las Américas, Puebla, México; Javier Vázquez-Salceda
BERT + Prompt & Feature	Applying model reference adaptive search to American-style option pricing can...	The author's affiliation is Wroclaw University of Science and Technology, Poland; The author's name is Deborah Timmons

4.3 BERT + Graph ML

To encode the powerful graph machine learning technique into our code, we can apply GNN layers after we compute the text embeddings of both the article and authors' features. We used the GraphSAGE model described by Hamilton et al. [5] to do this.

The following graphs show some **real** samples taken from our dataset.

The top one is drawn by sampling 500 author-paper pairs three times, where the papers are labeled with blue dots, and the authors are labeled with green, yellow, and orange dots, corresponding to the three samplings' results.

The bottom left one is drawn from the top graph but with authors with a paper count of less than 5 removed, and the bottom right one removes papers with an author count of less than 10. Our graph is pretty sparse, and most authors' research areas only concern a small group of people, and vice versa for papers.

The detailed results are summarized in **Table 1**. As we can see, when applying graphs, a much larger batch size is needed because GNN mini-batch training would sample a subgraph with their connected neighborhoods, and it would be better to use a larger batch size, given the sparsity of our citation network graph. We used 20 and 10 neighbors for the first and second GNN layers across all GraphSAGE model setups.

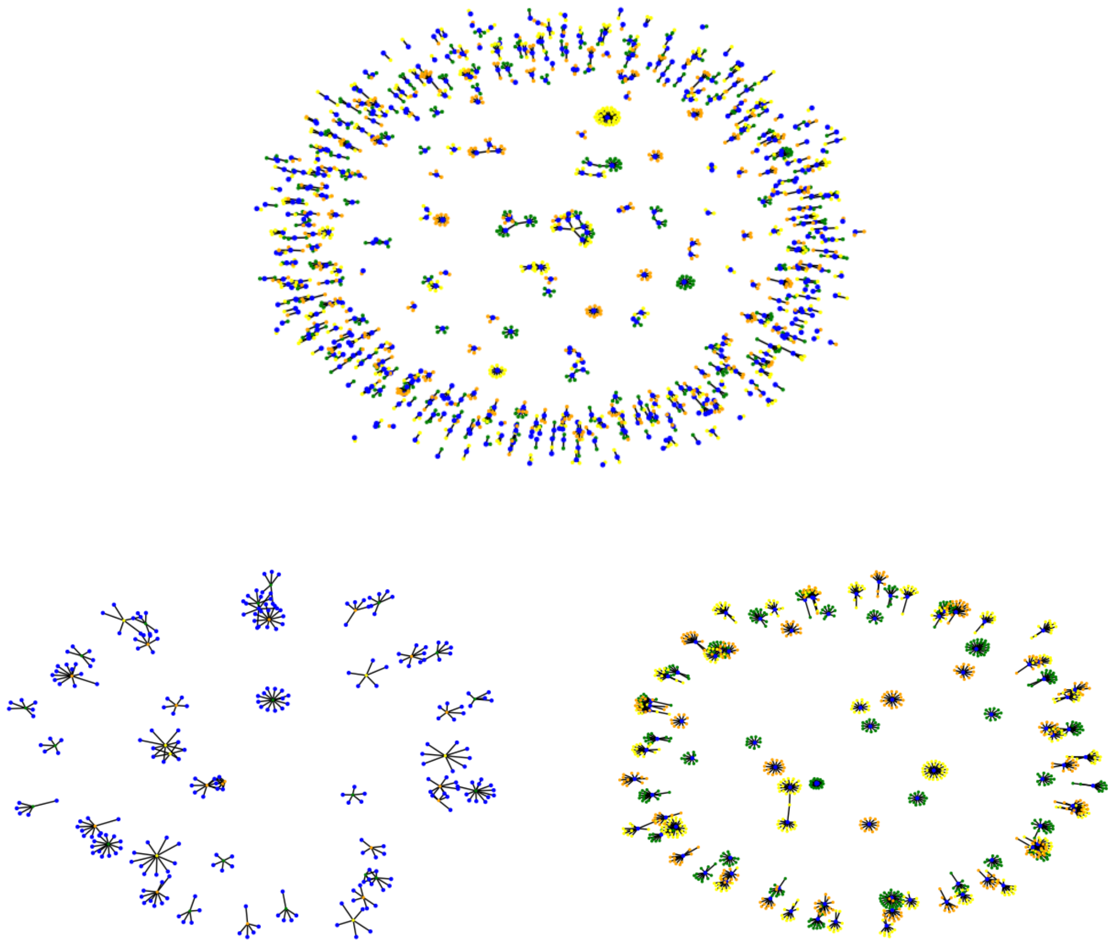

Figure 2: Graph Network Samples

4.4 Dataset and Code

We use a citation network dataset found at Aminer Citation. Currently, we only use around 20% subset of the DBLP V0 one due to GPU limitations (Google Colab one). The libraries we use are mainly PyTorch [6] and PyTorch Geometrics (PyG) [7].

The codes are available at https://github.com/yang-su2000/Authorship-Identification-with-NLP.

5 Discussion

5.1 Current Limitation

1. Limitation on how we use contrastive learning: the ratio between the positive and negative pair was ideal to be smaller. However, we could not reach this result due to limitations on computing units. Also, we do not implement the original contrastive loss because of limited data sizes.
2. Limitation on the dataset: we did not reach the goal of multi-feature portrait sketching. Due to the dataset's limitation, we only did one feature, affiliation.

5.2 Future Work

1. Dataset improvement. Right now, our results' accuracy and AUC performance are very promising. The next step of this project would be collecting more data for author features for multi-feature sketching.
2. Better PN paring ratio. With only a 1:1 positive vs. negative pair, our performance in AUC was already 79%; with better computing units and more time, we are confident that with a 1:f(n) ratio between positive pair and negative pair, we can further improve our performance.

6 Conclusion

To conclude, we proposed a new method for authorship identification that incorporates graph structures and contrastive learning techniques. This approach is better suited to real-world datasets with many authors and unknown authors, and we demonstrated that it outperforms traditional text classification methods. We hope to further improve the usability of our model by improving our data set to include more features of authors, as well as implementing a better method to comprise contrastive learning techniques.

References

[1] Abbasi, A., Javed, A.R., Iqbal, F., et al. "Authorship Identification Using Ensemble Learning." vol. 12, no. 9537, 2022, doi:10.1038/s41598-022-13690-4.

[2] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[3] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).

[4] Khosla, Prannay, et al. "Supervised contrastive learning." Advances in Neural Information Processing Systems 33 (2020): 18661-18673.

[5] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Advances in neural information processing systems 30 (2017).

[6] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." Advances in neural information processing systems 32 (2019).

[7] Fey, Matthias, and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric." arXiv preprint arXiv:1903.02428 (2019).