Enhancing Author Name Disambiguation: Combine Large Language Models and Machine Learning

4th Place Solution WhoIsWho-IND-KDD-2024

Lingze Xie
Z Lab
Guangzhou China
1458785076@qq.com

Guibin Lai
Z Lab
Guangzhou China
1711305687@qq.com

ABSTRACT

The rapid growth of online publications has increasingly complicated the problem of disambiguating authors with the same name. Existing disambiguation systems suffer from low accuracy, leading to errors in author rankings and instances of award fraud. This paper proposes a machine learning-based model to effectively detect incorrectly assigned papers within a given author's collection. The dataset includes personal profiles and detailed attributes of papers, such as title, abstract, authors, keywords, location, and year of publication. We developed four models: LightGBM, ChatGLM3-32k, Llama3, and GCN, and employed model fusion to leverage their differences. Through multiple rounds of experiments and validation on test sets, we achieved a fourth-place result. This model effectively detects misassigned papers for given authors, improving the accuracy of author disambiguation. Unlike existing academic search systems, our approach does not rely on pre-existing name disambiguation results. Consequently, our model more accurately identifies paper authorship, thereby preventing errors in author rankings and award fraud. This model has significant application value and research implications in the field of author name disambiguation.

KEYWORDS

Author Identification, Multi-model Fusion, Large Language Model, Graph Convolutional Network, Machine Learning, Natural Language Processing

1 INTRODUCTION

1.1 Background

In today's fast-paced academic environment, accurately identifying and distinguishing authorship has become an increasingly important challenge. With the increase in the number of researchers worldwide and the surge in academic publications, the problem of homonymous authors has become increasingly prominent, which has caused great trouble in areas such as document management, academic evaluation, and knowledge graph construction. The author identification task, also known as author disambiguation, aims to accurately distinguish different authors with the same or similar names and correctly attribute each academic paper to its true author.

The importance of this task is reflected in many aspects:

Document management: Accurate author identification helps librarians and researchers better organize and retrieve academic documents, and improves the efficiency and accuracy of document management.

Academic impact assessment: Accurate author identification is key when evaluating the academic impact of researchers. Incorrect author attribution may lead to incorrect assessments of researchers' contributions, affecting their career development and resource allocation.

Knowledge graph construction: Author identification is the basis for building a comprehensive and accurate academic knowledge graph. Accurate author information helps reveal the development trends, cooperation networks, and knowledge flows in the research field.

Academic integrity: Accurate author identification helps to detect academic misconduct, such as plagiarism or false author claims, thereby maintaining the integrity and fairness of the academic community.

Personal academic profile: For researchers, accurate author identification helps to build a complete personal academic profile, which is convenient for displaying their research results and professional development trajectory.

However, the author identification task faces many challenges. First, the problem of homonyms is prevalent, especially in some common names. Second, the author's information may change over time, such as changes in work units and research fields. Third, different publications may not represent author names in the same way, which increases the difficulty of identification. In addition, cross-language and cross-cultural author name representation also brings additional complexity.

In recent years, with the development of machine learning and natural language processing technology, researchers have begun to apply these advanced technologies to author identification tasks. From early rule-based methods to current deep learning and graph neural network methods, author identification technology has been

continuously updated. However, a single model is often difficult to fully capture multiple aspects of the author identification task. How to effectively combine the advantages of different models to improve the accuracy and robustness of identification has become a hot topic in current research.

In this context, this study proposes an author identification method based on multi-model fusion, aiming to improve the accuracy of author identification by integrating the advantages of different types of models. Our method achieved excellent results in the "Who is Who" competition, proving its effectiveness in practical applications.

1.2 Dataset Description

The dataset used in this study comes from the "Who is Who" competition. The dataset contains a large number of academic paper records covering multiple disciplines. The data includes specific paper information such as paper title, author name, author organization, publication year, journal name or conference name, abstract and keywords.The dataset contains a large number of authors with the same name to simulate the author disambiguation problem in the real world. At the same time, there are also cases where author information changes over time, such as changes in institutions and changes in research directions, which brings some challenges.

1.3 Task Description

The goal of this study is to develop a model that can effectively detect misassigned papers in a given author's collection of papers. Participants were not able to use disambiguation results of the same name from existing academic search systems. The evaluation index uses the AUC index commonly used in anomaly detection.

2 METHEDOLOGY

This study uses a variety of advanced machine learning and deep learning techniques to solve the author identity disambiguation problem. Our approach combines the advantages of graph neural networks, pre-trained language models, and traditional machine learning algorithms to form a powerful and flexible solution. The following is a detailed description of the main models and techniques we used:

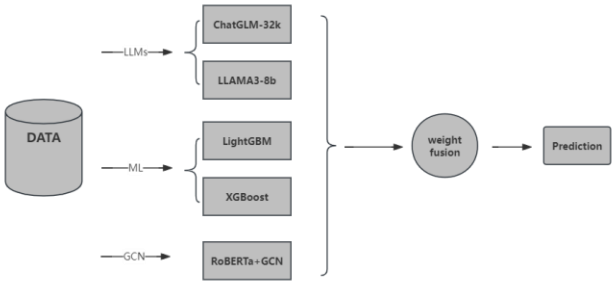

Figure1: Overview of our methodology pipeline

2.1 Graph Convolutional Network Model

Graph convolutional networks (GCN) are the core component of our approach, which can effectively process and learn graph-structured data. We use GCNs to capture complex relationships and structural information in the author-paper network.

The GCN model we designed contains the following key features:

2.1.1 Two-layer GCN structure: The model uses two layers of graph convolution layers. The first layer converts the input features into hidden representations, and the second layer further refines these representations. This two-layer structure allows information to propagate in the graph, enabling the model to capture node relationships at longer distances.

2.1.2 Non-linear activation: After each graph convolution layer, we apply the ReLU (Rectified Linear Unit) activation function. This introduces non-linearity and significantly enhances the expressiveness of the model, enabling it to learn more complex patterns.

2.1.3 Dropout regularization: To prevent overfitting, we apply dropout after each GCN layer. This technique randomly "drops" a certain proportion of neurons, forcing the network to learn more robust features and improving the generalization ability of the model.

2.1.4 Final linear layer: After the GCN layer, we add a fully connected layer. The role of this layer is to map the learned node representations to a binary classification space in preparation for the final classification decision.

2.1.5 Sigmoid Activation: In the output layer, we use the Sigmoid activation function. This compresses the output of the model to between 0 and 1, which can be interpreted as the probability that the node belongs to a specific category.

With this architecture, our GCN model is able to effectively learn the latent representation of the nodes while taking into account the overall structural information of the graph.

In order to make full use of the semantic information in the paper title, we use the pre-trained RoBERTa[5] model to encode the title. This method can capture the contextual information and potential semantics in the title, greatly improving the expressiveness of the features.

2.2 Gradient Boosting Decision Tree

XGBoost[2] and LightGBM[3] models are both classic SOTA (state of the art) Boosting algorithms, and can be classified into the gradient boosting decision tree algorithm series. They are both integrated learning frameworks based on decision trees, and they often combine powerful feature engineering to bring very good results. The following are our feature engineering and key steps.

2.2.1 Feature Engineering: We built hundreds of features for the classification model. The following list outlines some of the summary features included in the best single model:

- **Feature based on the author**
  - Number of papers: the total number of papers published by the author.

  - Citation statistics: including total citations, average citations per paper, and h-index.
  - Publication year statistics: first publication year, most recent publication year, and publication span.
  - Number of collaborators: the total number of different authors who have collaborated with the author.
  - Average number of co-authors: the average number of collaborators per paper.
  - Repeated collaboration rate: the proportion of authors who have collaborated multiple times to the total number of collaborators.
- **Feature based on the paper**
  - Title length: the average number of words in a paper title.
  - Abstract length: the average number of words in a paper abstract.
  - Number of keywords: the average number of keywords per paper.
- **Feature based on the Institution**
  - Number of institutions: the number of different institutions the author is associated with.
  - Frequency of institution changes: the number of times the author changes institutions divided by the total number of papers.
- **Feature based on the all text**
  - TF-IDF vector: TF-IDF vectorization of paper titles and abstracts.
  - Word frequency characteristics: the frequency of occurrence of high-frequency professional terms in the author's papers.

2.2.2 Training strategy: In order to maximize the effect of model training while reducing the risk of overfitting, we set the training hyperparameters as shown in Table 1 and Table 2

Table1: Hyperparameters setting of LightGBM Model

Hyperparameter	Value
boosting_type	gbdt
objective	binary
metric	auc
learning_rate	0.05
colsample_bytree	0.9
colsample_bynode	0.9
max_depth	12
reg_alpha	0.1
reg_lambda	10
max_bin	255
extra_trees	TRUE

Table2: Hyperparameters setting of Xgboost Model

Hyperparameter	Value
objective	binary:logistic
metric	auc
learning_rate	0.05
colsample_bytree	0.9
colsample_bynode	0.9
max_depth	12
reg_lambda	10
reg_alpha	0.1

Use 5-fold cross validation to evaluate model performance and adjust hyperparameters, and use an early stopping strategy: stop training when the performance on the validation set does not improve for 50 consecutive rounds.

2.3 ChatGLM-32k

We use ChatGLM-32K to process long texts and generate additional semantic features. The specific implementation includes:

2.3.1 Input processing: Concatenate all the author's paper titles and abstracts to form a long text input. Use sliding window technology to handle inputs with more than 32K tokens.

2.3.2 Prompt engineering: Design a specific prompt template, such as: "Analyze the following academic text and extract key topics and research areas:"

2.3.3 Feature generation: Use the model to generate a summary of each author's research topic (limited to 100 tokens). Extract the top-5 research keywords for each author.

2.3.4 Output processing: Use regular expressions to clean and standardize the model output. Convert the generated text features to numerical features (e.g., using LDA topic model).

2.3.5 Batch processing: Implement batch processing logic to process 100 authors at a time to optimize computational efficiency.

2.4 LLAMA model

We use the LLaMA[4] model to enhance feature representation and perform few-shot learning. The specific implementation is as follows:

2.4.1 Model selection: Use the 7B parameter version of the LLaMA model to strike a balance between performance and efficiency.

2.4.2 Fine-tuning process: Use LoRA (Low-Rank Adaptation) technology for efficient fine-tuning. The learning rate is set to 3e-4 and trained for 5 epochs.

2.4.3 Input formatting: Design a specific input template containing information such as author name, institution, and paper title. Example: "Does the paper {paper title} published by {author name} belong to the same person? Answer: Yes/No"

2.4.4 Few-shot learning: Select 10 representative samples for each author category. Construct a context containing these samples as the input prefix of the model.

2.4.5 Output processing: Use regular expressions to extract the yes/no judgment of the model. Convert the output to a probability score (e.g., based on the confidence of the generated token).

2.4.6 Integration strategy: Add the output of LLaMA as an additional feature to the final integrated model.

2.5 Model Ensemble Strategy

To take full advantage of the strengths of each model, we adopted a carefully designed model ensemble strategy:

2.5.1 Feature-level fusion: We combine the node representations learned by GCN[6], the title encodings generated by RoBERTa, and the hand-crafted features to form a comprehensive feature set. These features are fed into the LightGBM and XGBoost models simultaneously.

2.5.2 Prediction-level fusion: We fuse the predictions of GCN, LightGBM, and XGBoost using a weighted average. The weights are determined by grid search on the validation set to obtain the best combination.

2.5.3 Cross-validation: During training, we use k-fold cross-validation to ensure the stability and generalization ability of the model. This also helps us more accurately estimate the performance of the model on unseen data.

2.5.4 Stacking: We also tried a more advanced stacking ensemble method, using a meta-learner (usually logistic regression) to learn how to best combine the predictions of the base models.

2.5.5 Difference Preservation: During the ensemble process, we pay attention to maintaining the differences between the models. For example, for tree-based models, we use different feature subsets and parameter settings to ensure that they can provide complementary predictions.

Through this multi-model ensemble strategy, we are able to fully leverage the strengths of each model while making up for their respective shortcomings, resulting in a more robust and accurate authorship disambiguation system. This comprehensive approach not only improves the overall performance of the model, but also enhances its adaptability and reliability in handling various complex scenarios.

3 CONCLUSION

In this paper, we present our solution to WhoIsWho-IND-KDD-2024 competition. Our approach employs a multi-model fusion strategy aimed at addressing the complex problem of academic author name disambiguation. By integrating various machine learning models including Graph Convolutional Networks (GCN), RoBERTa encoder, XGBoost, and LightGBM, along with leveraging large language models such as ChatGLM-32K and LLaMA, we constructed a robust and flexible system. We meticulously designed multi-dimensional feature engineering, encompassing author features, paper features, institutional features, domain features, and co-authorship features, to comprehensively capture authors' academic characteristics. This integrated approach enabled us to effectively handle various challenges in author name disambiguation, such as same-name different-person cases and cross-domain publications. Ultimately, our model achieved an AUC score of 0.8089 in the competition, ranking us 4th among all participating teams, demonstrating the effectiveness and competitiveness of our method.

REFERENCES

[1] WhoIsWho-IND-KDD-2024. https://www.biendata.xyz/competition/ind_kdd_2024

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785-794. https://doi.org/10.1145/2939672.2939785

[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3146–3154.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[5] Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, Yanzheng Cai, Guoyang Zeng, Zhixing Tan, Zhiyuan Liu, Minlie Huang, Wentao Han, Yang Liu, Xiaoyan Zhu, and Maosong Sun. 2022. CPM: A Large-scale Generative Chinese Pre-trained LanguageModel. AI Open 3, 100027. https://doi.org/10.1016/j.aiopen.2022.100027

[6] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In International Conference on Learning Representations (ICLR).