Enhancing Name Disambiguation via Iterative Self-Refining with LLMs

Xiaocheng Zhang
MeiTuan
BeiJing, China
zhangxiaocheng@meituan.com

Yang Zhou
MeiTuan
ShangHai, China
zhouyang96@meituan.com

Haoru Chen
MeiTuan
BeiJing, China
chenhaoru02@meituan.com

Mengjiao Bao
MeiTuan
BeiJing, China
baomengjiao@meituan.com

Peng Yan*
MeiTuan
BeiJing, China
yanpeng04@meituan.com

Abstract

This paper presents the solution of our team BlackPearl in the WhoIsWho-IND Task of KDD Cup 2024 Open Academic Graph(OAG) Challenge.

The goal of the competition is to explore ways to discover paper assignment errors for given authors. In this paper, We present a LLM-based Name Disambiguator via Iterative Self-Refining. Our method transforms the clustering task into a comparison task, and improves the model's confidence that the current author belongs to the main class by iteratively improving the proportion of correct authors contained in the model input during reasoning. In addition, we employed Train-Time Difficulty Increase(TTDI), Test-Time Augmentation (TTA) techniques, and multi-source information model ensemble to maximizing the utilization of various information sources. Our method ranks 1st in the final leaderboard, code is publicly available at https://github.com/BlackPearl-Lab/KddCup-2024-OAG-Challenge-1st-Solutions.

CCS Concepts

• **Computing methodologies** → **Natural language processing**.

Keywords

Natural Language Processing, Entity Disambiguation, Entity Clustering, Large Language Model

ACM Reference Format:
Xiaocheng Zhang, Yang Zhou, Haoru Chen, Mengjiao Bao, and Peng Yan. 2024. Enhancing Name Disambiguation via Iterative Self-Refining with LLMs. In *Proceedings of KDD 2024 Workshop OAG-Challenge Cup (KDD-Cup'24).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author of this research.

KDDCup'24, Aug 25 2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

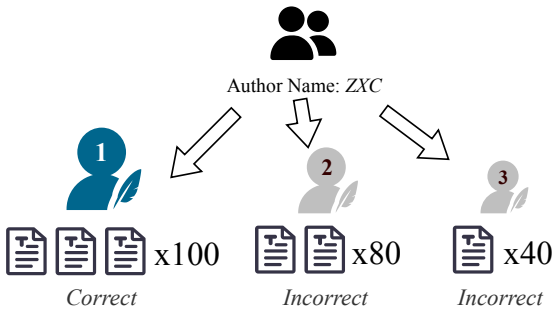

Figure 1: An example of name disambiguation.

1 Introduction

The increasing number of online publications has made the name ambiguity problem more complex. Moreover, the inaccurate disambiguation results have led to invalid author rankings and award cheating. To address the above challenges, KEG and Zhipu Ai proposed the WhoIsWho benchmark, leaderboard, and toolkit and hosted the KDD CUP 2024 OAG-Challenge[1, 9].

1.1 Task Description

Given each author's profile, including author name and published papers, participants were asked to develop a model to detect incorrect paper assignments across all papers without being allowed to use existing academic search systems. Paper attributes were provided, including title, abstract, author, keywords, venue, and year of publication. Figure 1 provides an example of name disambiguation.

1.2 Dataset Description

The data is organized into a dictionary. The key is the author ID; the value contains the name of the author (*name*). For the train dataset, the paper IDs owned by the concerned author(*normal_data*), and the paper IDs that are incorrectly assigned to the author(*outliers*). For the dev and test dataset, *papers* field of each author is all associated papers of this author. Paper attributes encompass *ID*, *title*, *author.name*, *author.org*, *venue*, *year*, *keywords* and *abstract*. The fundamental statistics of the datasets are summarized in Table 1.

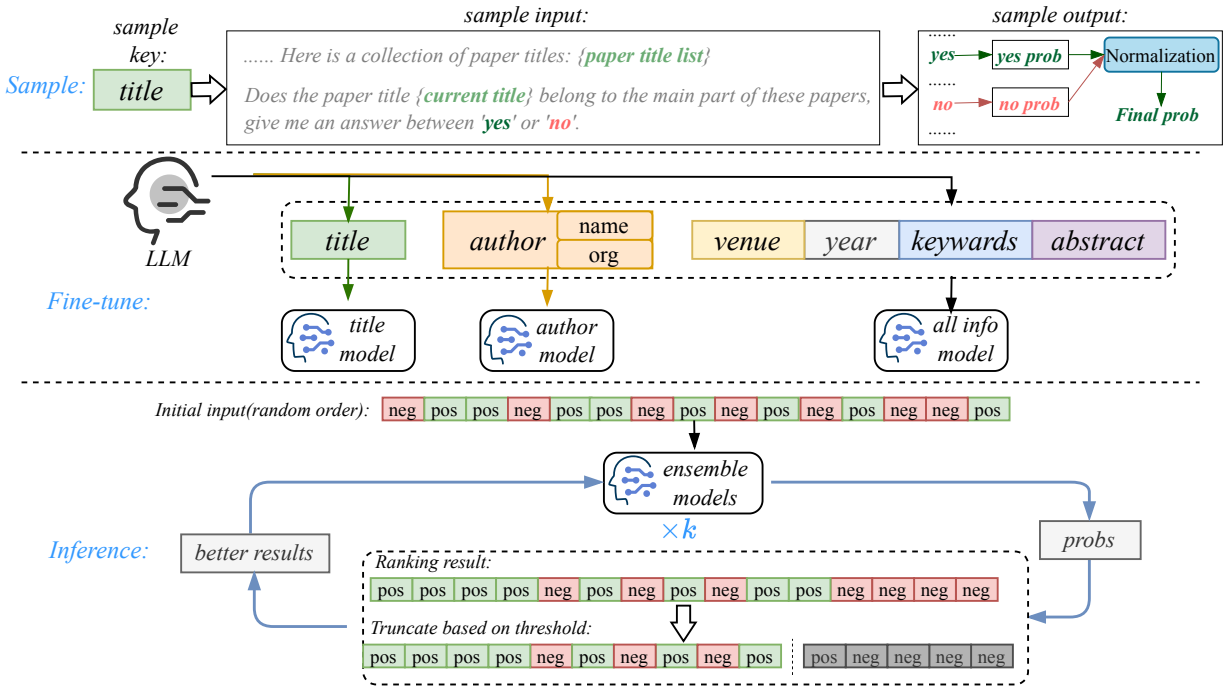

Figure 2: The framework we used in this competition. The uppermost part of the figure is a sample where we use the title as the information source. In the fine-tune phase, multiple models are fine-tuned for different information sources. Ensemble models means weighted averaging of the results inferred from multiple fine-tuned models. k represents the number of iterative self-refining rounds.

Split	Author	POS	NEG	Total
Train	779	131,024	17,285	148,309
dev	370	–	–	62,229
test	515	–	–	116,262

Table 1: Statistics of the datasets.

2 Methodology

In this section, we introduce the architecture of the proposed approach first and then present the details of the fine-tuning process and inference process. Finally, we introduce the loss function and some tricks to improve the score.

2.1 Overview

The Figure 2 shows our framework. In the fine-tuning phase, multiple models are fine-tuned to focus on different information sources. In the inference phase, an iterative self-refining method is used to reintegrate the output of the model as the input for the next round, thereby increasing the probability of identifying the correct paper.

2.2 Fine-tune

We continue the fine-tuning approach of the baseline, which transforms the clustering task into a comparison task, where we give a bunch of reference papers in the input and determine whether the current paper is the main class. Empirically, the more reference papers in the input, the more credible the current judgment result will be. However, under the limit of the maximum input length, the more reference papers are spliced into the input, the less information each paper contains.

To address this issue, we developed a strategy to split, fine-tune and finally integrate multiple information sources. Based on logical reasoning and experimental validation, we identified the title and author as the two most critical information sources for fine-tuning the model, employing Low-Rank Adaptation (LoRA)[5] as the fine-tuning technique. Subsequently, we fine-tuned a comprehensive model that utilizes all available information sources, using Quantized Low-Rank Adaptation (QLoRA)[3] as the fine-tuning method.

2.3 Iterative Self-Refining

Since we employ a comparison task to determine whether the current paper belongs to the main category, the inference length can be extended during the inference phase to incorporate more

reference papers, thereby enhancing the model's robustness in reasoning. It is natural to consider that the higher the proportion of correct papers among the reference papers, the more confident the model will be in judging the correctness of the current paper.

Based on this rationale, we propose the Iterative Self-Refining (IRF) method, which does not require additional model training and achieves better results by continuously refining the proportion of correct papers in the reference set. Specifically, we sort the papers by their probability of correctness, placing the correct papers at the forefront and truncating the papers at the back based on a predefined threshold. The initial input for the inference stage is randomly sorted. Detailed iterative benefits and threshold settings are provided in the experimental section.

2.4 Sample Augmentation for Fine-Tuning and Inference

2.4.1 Train-Time Difficulty Increase(TTDI). During the training process, we aim to increase the task difficulty by, for instance, reducing the maximum training length of the *authortask* to prevent the task from becoming overly simplistic. By appropriately increasing the proportion of incorrect papers in the training input, we can push the model out of its "comfort zone" and enable it to better handle challenging examples during inference.

2.4.2 Test-Time Augmentation(TTA). In the comparison task, the probability of the model output should not be influenced by the input order of the reference papers. In response to this issue, we make full use of TTA[8] by shuffling the order of the reference papers in each sample before feeding them into the model, and averaging multiple results to achieve more robust outcomes.

3 Experiments

In this section, we present our main results and ablation studies for some crucial components.

3.1 Experimental Setup

3.1.1 Metrics. This competition adopt Area Under ROC Curve (AUC), broadly adopted in anomaly detection as the evaluation metric.For each author:

$$Weight = \frac{\#ErrorsOfTheAuthor}{\#TotalErrors} \tag{1}$$

For all authors (M is the number of authors):

$$WeightedAUC = \sum_{i=1}^{M} AUC_i \times weight_i \tag{2}$$

3.1.2 Experiment Settings. Our implementations are based on Pytorch. The base model selected is ChatGLM-6B-32k[4]. The number of training epochs is set to 1. The maximum training length for the *authormodel* is 15k, while for other models it is set to 25k. The rank, alpha, and dropout parameters of LoRA are set to 128, 256, and 0.05, respectively. The warmup ratio is set to 0.03. During the inference phase, the maximum input length is set to 30k. The number of iterations k for ISR is set to 3. For the inference of the *author* and *title* models, the truncation threshold is set to 0.6, whereas for the all info model, the truncation threshold is set to 0.5. Both training and inference are performed on $8 \times A100$ GPUs.

Methods	Dev	Test
GCN[7]	0.586	–
GCCAD[2]	0.634	–
LGB[6] Ensemble	–	0.799
LGB + ChatGLM	–	0.813
Title model	0.757	0.767
Author model	0.715	–
All info model	0.758	–
Ours	**0.794**	**0.834**

Table 2: Overall performance on dev and test dataset.

Methods	Dev	Test
Title model	0.757	0.767
+TTA	0.761	0.772
Author model	0.715	–
+TTDI	0.727	0.788
Ensemble model	0.772	0.808
+ISR, k = 1	0.786	0.827
+ISR, k = 2	0.791	0.831
+ISR, k = 3	**0.794**	**0.834**

Table 3: Ablation study on Dev and Test datasets. It shows the results of adding TTA, TTDI and ISR.

3.2 Overall Performance

The overall performance is shown in Table 2. Based on the experimental results, it can be concluded that graph neural network methods perform poorly on the competition dataset, indicating significant room for improvement. Tree-based models achieve commendable performance, with a score of 0.799 on the Test dataset. Combining features output by large models with tree-based models yields superior results, achieving a score of 0.813 on the Test dataset. Individual Title, Author, and All info models only achieve mediocre scores, as they do not fully exploit the available information sources, and their inputs are not refined. Our method, through multiple iterations and the integration of diverse information sources, attained optimal performance on both the Dev and Test datasets, with scores of 0.794 and 0.834 respectively.

3.3 Ablation Study

To answer how TTA, TTDI, and ISR contribute to the performance of our method, as well as to quantify the benefits of iterative refinement in ISR, we construct an ablation study by TTA, TTDI and ISR respectively. The results are shown in Table 3. TTA enhance the performance of the title model by 0.004 on the Dev dataset and by 0.005 on the Test dataset. This validates that TTA can mitigate the impact of the reference paper order on the results. TTDI

improve the author model by 0.012 on the Dev dataset, which is a significant improvement. This suggests that we can explore a paradigm of "making training harder to make inference easier". ISR demonstrated significant score improvements after just one iteration, with the ensemble model achieving an increase of 0.014 on the Dev dataset and 0.019 on the Test dataset. After three iterations, the improvements reached 0.022 and 0.026, respectively. This demonstrates the effectiveness of our iteration.

4 Conclusion

In this paper, we propose a novel approach to address the Name Disambiguation task, achieving first place in the WhoIsWho-IND Task of the KDD Cup 2024 Open Academic Graph (OAG) Challenge. In particular, during the fine-tuning phase, we employed LoRA/QLoRA methods to fine-tune multiple models based on different information sources. In the inference phase, we innovatively proposed an Iterative Self-Refining (ISR) method, which enhances the model's confidence in its judgments by continuously refining the proportion of correct papers among the reference papers. We also utilized Train-Time Difficulty Increase (TTDI) and Test-Time Augmentation (TTA) during the fine-tuning and inference phases respectively, to further improve the scores. Comparative and ablation experimental results demonstrate the effectiveness of our approach. We hope that our proposed solution will inspire other researchers and practitioners in the field. In the future, we will improve this method and apply it to datasets in other domains to achieve a broader impact.

References

[1] Bo Chen, Jing Zhang, Fanjin Zhang, Tianyi Han, Yuqing Cheng, Xiaoyan Li, Yuxiao Dong, and Jie Tang. 2023. Web-scale academic name disambiguation: the WhoIsWho benchmark, leaderboard, and toolkit. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3817–3828.

[2] Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang. 2022. Gccad: Graph contrastive coding for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8037–8051.

[3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems* 36 (2024).

[4] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 320–335.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[6] 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. *Advances in neural information processing systems* 30 (2017).

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

[8] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. 2019. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*. Springer, 61–72.

[9] Fanjin Zhang, Shijie Shi, Yifan Zhu, Bo Chen, Yukuo Cen, Jifan Yu, Yelin Chen, Lulu Wang, Qingfei Zhao, Yuqing Cheng, Tianyi Han, Yuwei An, Dan Zhang, Weng Lam Tam, Kun Cao, Yunhe Pang, Xinyu Guan, Huihui Yuan, Jian Song, Xiaoyan Li, Yuxiao Dong, and Jie Tang. 2024. OAG-Bench: A Human-Curated Benchmark for Academic Graph Mining. *arXiv preprint arXiv:2402.15810* (2024).