Synergizing Large Language Models and Tree-based Algorithms for Author Name Disambiguation

The 3rd Place Solution for WhoIsWho-IND

ShiEn He
School of Statics
East China Normal University
China
1162805600@qq.com

Qiang Yan
CAS Key Laboratory of Network Data Science and Technology
Institute of Computing Technology, Chinese Academy of Sciences
China
yanqiang@ict.ac.cn

ABSTRACT

The ultimate goal of academic data mining is to deepen our understanding of the development, nature, and trends of science. It offers the potential to uncover significant scientific, technological, and educational value. For instance, deep mining of academic data can assist governments in formulating science policies, support companies in talent discovery, and help researchers access new knowledge more effectively. Academic data mining encompasses many applications centered around academic entities, such as paper retrieval, expert discovery, and journal recommendation. However, the lack of data benchmarks related to academic knowledge graph mining has severely limited the development of this field. At KDD Cup 2024, we introduced the OAG-Challenge, consisting of three realistic and challenging academic tasks aimed at advancing the field of academic knowledge graph mining.

One of these tasks, WhoIsWho-IND, focuses on the increasingly complex problem of author name disambiguation due to the rapid increase in online publications. Inaccuracies in existing disambiguation systems have led to incorrect author rankings and award fraud. This competition challenges participants to develop a model that detects misassigned papers for a given author.

In this work, we approached the WhoIsWho-IND task by framing it as a binary classification problem, determining whether a paper belongs to an author. We employed two strategies: (1) extracting fundamental information from papers and authors and deriving their textual representations, followed by utilizing LightGBM for classification, and (2) fine-tuning a large model to assess the relevance of a list of papers to the author's historical publications. Both methods output a probability indicating the likelihood of correct paper assignment to the author. Our approach achieved significant results, earning us third place in the competition. Our code is published on github.

KEYWORDS

Large Language Models, Instruction Tuning, Feature Engineering, LightGBM, Model Fusion

1 Introduction

Academic data mining holds significant importance in understanding the development, nature, and trends of science. By deeply mining academic data, it is possible to uncover substantial scientific, technological, and educational value. For instance, such data mining efforts can assist governments in formulating effective science policies, support companies in discovering talent, and help researchers access new knowledge more efficiently. Academic data mining applications span various domains, including paper retrieval, expert discovery, and journal recommendations. Despite its potential, the lack of data benchmarks related to academic knowledge graph mining has severely restricted the advancement of this field. Addressing these limitations is crucial for furthering our comprehension of scientific progress and fostering innovation.

1.1 Background

The KDD Cup 2024 introduces the OAG-Challenge, aimed at promoting advancements in academic knowledge graph mining. This challenge comprises three realistic and challenging academic tasks designed to push the boundaries of this field. Among these tasks is the WhoIsWho-IND, which addresses the increasingly complex problem of author name disambiguation. With the rapid increase in online publications, distinguishing between authors with the same name has become more challenging. Existing disambiguation systems often fail, leading to incorrect author rankings and instances of award fraud. The WhoIsWho-IND task challenges participants to develop a model capable of detecting misassigned papers for a given author, thereby improving the accuracy and reliability of academic databases. This task involves analyzing author profiles and their published papers, utilizing detailed attributes such as titles, abstracts, authors, keywords, venues, and publication years to identify incorrectly assigned works.

1.2 Dataset Description

The dataset utilized in this competition originates from AMiner.cn and is meticulously organized into several key files, each serving a distinct purpose. The primary files included in the dataset are detailed below:

(1) train_author.json

This file is organized as a dictionary where the keys represent author IDs. Each entry within the dictionary includes:

- name: The author's name.

https://github.com/yanqiangmiffy/KDD2024-WhoIsWho-Top3

- normal_data: A list of paper IDs correctly assigned to the author.
- outliers: A list of paper IDs incorrectly assigned to the author.

(2) pid_to_info_all.json

This file encompasses detailed information about all papers used in the competition. It is structured as a dictionary, with paper IDs serving as keys and various attributes as values. The attributes include:

- ID (string): The unique identifier for the paper (e.g., "53e9ab9eb7602d970354a97e").
- title (string): The title of the paper (e.g., "Data mining: concepts and techniques").
- authors.name (string): The name of the author (e.g., "Jiawei Han").
- author.org (string): The organization of the author (e.g., "Department of Computer Science, University of Illinois at Urbana-Champaign").
- venue (string): The conference or journal where the paper was published (e.g., "Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial").
- year (int): The year of publication (e.g., 2000).
- keywords (list of strings): Keywords associated with the paper (e.g., ["data mining", "structured data", "world wide web", "social network", "relational data"]).
- abstract (string): The abstract of the paper, summarizing its content (e.g., "Our ability to generate…").

(3) ind_valid_author.json

This file follows a similar structure to train_author.json and provides validation data. Each entry for an author includes all papers associated with that author.

(4) ind_valid_author_submit.json

This file provides a sample submission format for the validation set.

(5) ind_test_author_filter_public.json

This file contains the public test set data, maintaining the same structure as ind_valid_author.json.

(6) ind_test_author_submit.json

This file provides a sample submission format for the test set.

The structured nature of this dataset, including comprehensive paper attributes and detailed author information, supports the task of author name disambiguation. It enables the development of sophisticated models capable of accurately detecting misassigned papers, thereby improving the reliability of academic databases.

1.3 Task Description

The WhoIsWho-IND task within the OAG-Challenge addresses the complex issue of author name disambiguation, which has become increasingly challenging due to the rapid growth of online publications. Inaccuracies in current disambiguation systems have led to erroneous author rankings and instances of award fraud. This competition aims to develop models that can accurately detect papers misassigned to a given author. Participants are provided with each author's profile, which includes the author's name and a list of their published papers. The task requires developing a model to identify papers that have been incorrectly attributed to the author. To assist with this, the dataset includes detailed attributes for each paper, such as the title, abstract, authors, keywords, venue, and publication year.

It is important to note that participants are not allowed to use existing disambiguation results from academic search systems. This restriction ensures that the solutions are developed independently and contribute to advancing the state of the art in author name disambiguation. The challenge involves creating a robust model that can analyze the provided data and accurately detect misassignments, thereby improving the reliability of academic databases and contributing to the broader goals of academic data mining.

2 Methodology

2.1 Data Processing

In this study, we leveraged multiple datasets to address the WhoIsWho-IND task in the OAG-Challenge. The data processing pipeline is designed to transform raw JSON files into structured CSV files, followed by further processing to facilitate model development. Below are the detailed steps involved in the data processing:

Data Loading: We loaded several key JSON files containing the necessary data for training and validation:

- train_author.json: This file contains information about authors, including their correctly assigned papers (normal_data) and misassigned papers (outliers).
- pid_to_info_all.json: This file includes detailed metadata for all papers used in the competition.
- ind_test_author_filter_public.json: This file provides validation data for authors.
- ind_test_author_submit.json: A sample submission file for the validation set.

Data Conversion: We converted the paper metadata from *pid_to_info_all.json* into a CSV file (papers_info.csv) to facilitate easier access and manipulation. Each row in this CSV file corresponds to a single paper with attributes such as title, authors, organization, venue, publication year, keywords, and abstract. We transformed the author data from *train_author.json* and *ind_test_author_filter_public.json* into separate CSV files (train_author.csv and valid_author.csv). Each row in these CSV files represents a paper assigned to an author, including attributes such as author ID, author name, paper ID, and a label indicating whether the paper is correctly assigned (1) or misassigned (0).

Data Enrichment: For each paper, we extracted and processed additional features such as author names and organizations, converted them to lowercase, and concatenated them into text fields (author_names_text and org_names_text). We also identified the top authors and organizations associated with each author based on frequency counts.

Data Integration: We combined the processed training and validation data into a single DataFrame, which was subsequently saved as a pickle file (step1_df.pkl) and a CSV file (step1_df.csv). This integrated dataset serves as the foundation for further analysis and model development.

2.2 Feature Engineering

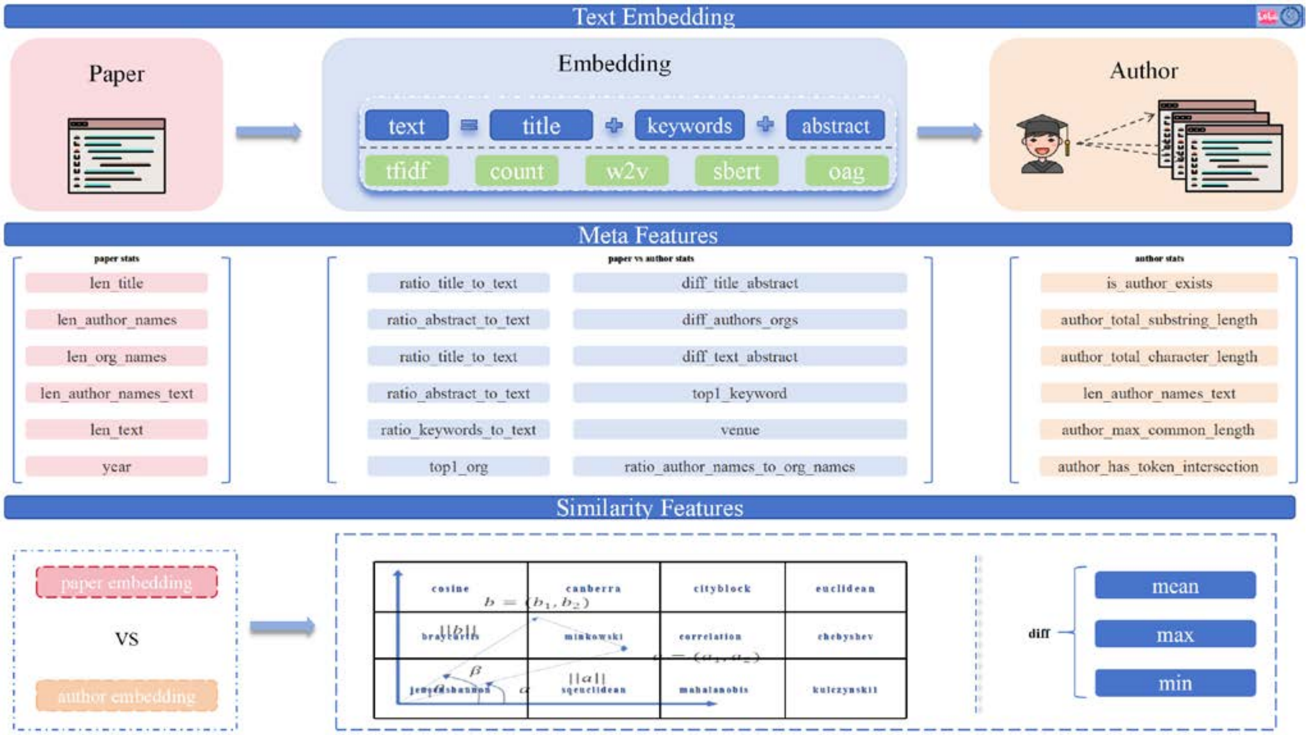


Figure 1 Feature Engineering

To enhance the performance of our model for the WhoIsWho-IND task, we meticulously engineered several features from the raw data. As shown in Figure 1,these features were derived through various text processing and statistical methods, which are described below:

(1) Text Processing:

Text Representation: We created text representations for multiple columns, including \`title\`, \`keywords\`, \`abstract\`, \`author_names_text\`, \`org_names_text\`, and \`venue\`. The text data was processed and concatenated to form comprehensive textual descriptions for each paper.

Text Embeddings: Using pre-trained text models, we converted the processed text data into numerical embeddings. These embeddings capture the semantic meaning of the text and were crucial for downstream tasks.

Distance Calculations: We computed distances between text embeddings of different papers to identify potential outliers. This involved calculating cosine similarity and other distance metrics.

(2) Statistical Features:

Author and Organization Statistics: For each author, we aggregated statistics based on their publication history. This included the count and uniqueness of venues, first and second authors, top keywords, and organizations.

Text Lengths: We calculated the lengths of various text fields (e.g., author names, organization names, abstracts) and derived statistical measures such as maximum, minimum, mean, standard deviation, and sum.

(3) Author-Specific Features:

Top Entities: For each author, we identified the top authors and organizations they frequently collaborated with. These were determined based on the frequency of co-authorship and affiliations.

Label Encoding: We applied label encoding to categorical features such as \`first_author\`, \`second_author\`, \`top1_author\`, \`top2_author\`, \`top1_org\`, \`top2_org\`, and \`venue\`. This transformation converted categorical data into numerical format, suitable for machine learning models.

(4) Data Merging and Integration:

We merged the processed features with the original dataset to create a comprehensive feature set for both training and validation data. This involved integrating text embeddings, statistical measures, and author-specific features.

We also handled missing values and replaced infinite values with NaN to ensure data quality.

By combining text embeddings, statistical features, and author-specific attributes, we constructed a robust feature set that captures the essential characteristics of the data. These features served as the foundation for our subsequent model training and evaluation.

2.3 LightGBM Setup

LightGBM is an open-source, distributed, high-performance gradient boosting framework developed by Microsoft. It is designed for efficiency, scalability, and accuracy. It is based on decision trees designed to improve model efficiency and reduce memory usage. It incorporates several novel techniques, including

Gradient-based One-Side Sampling (GOSS), which selectively retains instances with large gradients during training to optimize memory usage and training time. Additionally, LightGBM employs histogram-based algorithms for efficient tree construction. These techniques, along with optimizations like leaf-wise tree growth and efficient data storage formats, contribute to LightGBM's efficiency and give it a competitive edge over other gradient boosting frameworks.

We compiled a comprehensive list of features based on various attributes such as title, authors, abstract, keywords, author names, organization names, and more. An importance-based feature selection was performed using a pre-existing feature importance file (lgb_importance_df.csv). We filtered out features that were not among the top 500 most important features. Features containing certain keywords (_label_, jaccard, sorensen) were excluded from the final feature set to reduce noise and improve model performance.

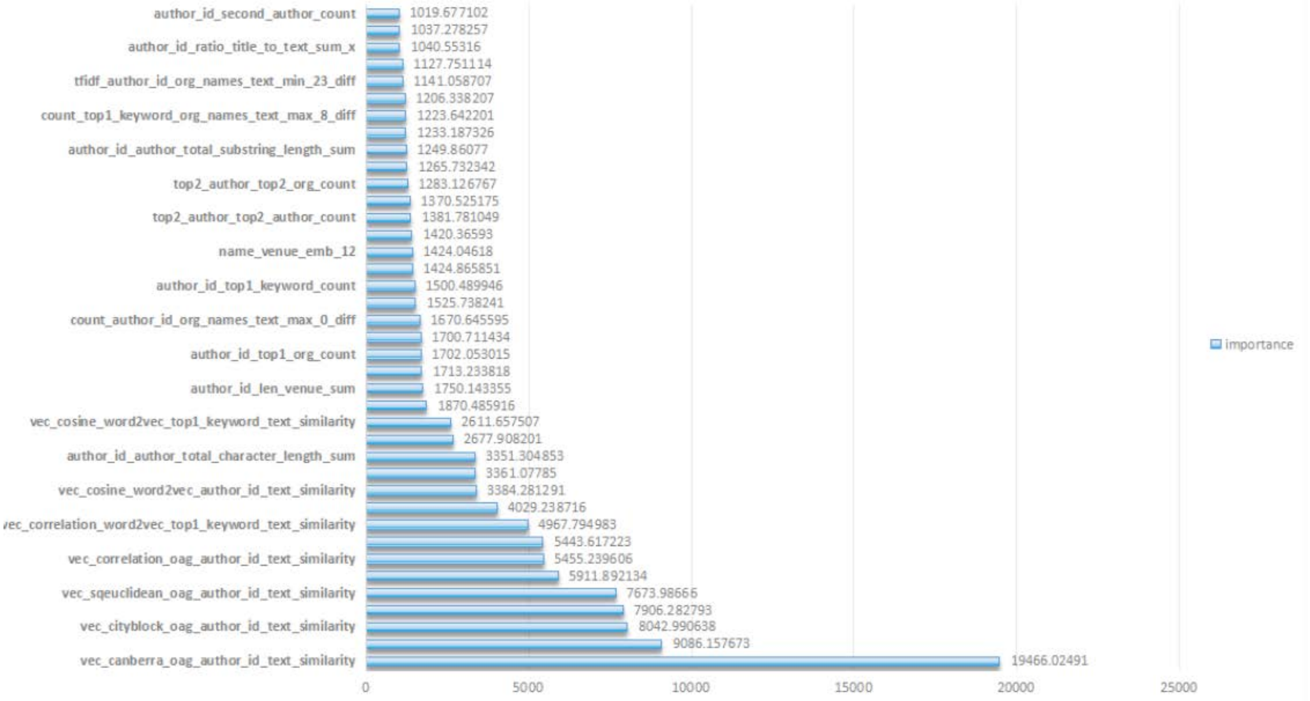
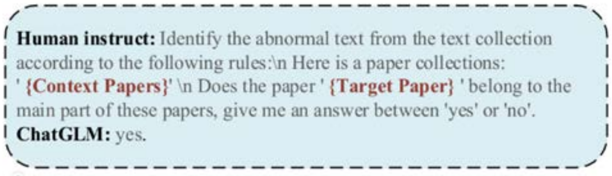


Figure 2 Feature Importance

Then we utilized the LGBMClassifier from the LightGBM library, which is optimized for speed and efficiency.The model was trained using a StratifiedGroupKFold cross-validation strategy with 5 folds, ensuring that the splits were stratified by the target label and grouped by the author ID. This approach helped in handling potential leakage and ensured robust evaluation.

During training, we employed evaluation metrics such as ROC-AUC to monitor the performance of the model on the validation set. We also implemented early stopping and logging callbacks to prevent overfitting and to track the training progress.

2.4 LLM Instruction Tuning



Human instruct: Identify the abnormal text from the text collection according to the following rules:\n Here is a paper collections: ' **{Context Papers}**' \n Does the paper ' **{Target Paper}** ' belong to the main part of these papers, give me an answer between 'yes' or 'no'.

ChatGLM: yes.

Figure 3 LLM Instruction Tuning

Instruction tuning of large language models (LLMs) is an essential step to enhance their performance on specific tasks. This process involves adapting a pre-trained model to follow human instructions more accurately, improving its ability to handle nuanced queries and generate more relevant responses. The objective is to fine-tune the model using a targeted dataset that reflects the desired task performance, thereby aligning the model's outputs with the expected results. Figure 2 is the prompt which we use.

In the context of the KDD Cup 2024's WhoIsWho-IND task, instruction tuning plays a critical role. The task requires participants to develop a model that can accurately detect papers erroneously assigned to a given author. The dataset provided includes detailed attributes of all involved papers, such as titles, abstracts, authors, keywords, locations, and publication years. Participants are expected to fine-tune their models on this dataset without leveraging existing disambiguation results from academic search systems. The large models we use include **glm4, mistral and chatglm3**.

Our fine-tuning code implements an instruction fine-tuning pipeline for large language models (LLMs). It supports both GLM and causal language model architectures, integrating advanced techniques like QLoRA for 4-bit quantization and LoRA for efficient adaptation. The script handles data preprocessing, model loading, and training setup, including options for gradient checkpointing and resuming from checkpoints. It uses custom dataset classes and collators to manage task-specific data formats. The training process is managed by a Trainer class, with options for using the Unsloth library for additional optimizations. Throughout the pipeline, various optimizations are applied to manage memory usage and improve training efficiency for large models. The code is designed to be flexible, allowing for different model types, quantization strategies, and adaptation techniques, making it suitable for a range of instruction tuning tasks on LLMs.

2.5 Ensemble Models

We grouped and normalized the probabilities of the LGB model and the LLM model, and then integrated the probabilities together by weighted summation. Model integration has greatly improved our scores.

$$\text{Results} = W_{lgb} \times Result_{lgb} + W_{llm} \times Result_{llm}$$

Where W represents the score weight after normalization.

3 Results and Discussion

As shown in Figure 3, the analysis of feature importance reveals that text similarity features play a dominant role in accurately identifying the textual context and similarities between author IDs and their respective papers, which is crucial for detecting misassigned papers. The use of various embeddings, such as Cadebert, GloVe, Word2Vec, and FastText, indicates a comprehensive approach to capturing different aspects of textual similarity. This diversity in embeddings enhances the robustness of the model.

On the other hand, count and length-based features, while still contributing to the model, have relatively lower importance, suggesting that they serve as supplementary features to the primary similarity-based ones. In conclusion, the feature importance analysis indicates that leveraging multiple embedding techniques to capture the textual context and similarities is paramount in solving the WhoIsWho-IND task, with count and length features playing a secondary role in the overall model performance.

In our study, we utilized large models to determine if a specific text, referred to as the "Target Paper," belongs to a given set of author texts, known as the "Paper Collection." The Context Papers represent the collection of papers attributed to the current author, while the Target Paper is the one being tested for correct attribution. Our findings indicate that the performance of large models significantly surpasses that of the LGB model, demonstrating remarkable inferential capabilities.

The large language models' ability to capture complex textual nuances and contextual relationships allowed for more accurate identification of misassigned papers. This superiority is attributed to their advanced architectures and extensive training on diverse datasets, which enable them to generalize better and understand intricate textual patterns. Consequently, the use of large models not only enhances the accuracy of author disambiguation tasks but also showcases their potential in improving the overall robustness and reliability of academic data mining processes.

From the rankings, we can see that our lb score is different from other players. The main reason is that our lgb model does not use the vector model based on the large model. In addition, our large model fine-tuning strategy is lacking, and the prediction is relatively simple.

Table 1 Ranking top 3 scores

Team Name	LB Score	Parameters
BlackPearl	0.83454	6,000,000,000
LoveFishO	0.82487	2,000,000,000
AGreat	0.81349	9,000,000,000

4 Conclusion

This study demonstrates the significant potential of combining traditional machine learning techniques with large language models for the task of author-paper matching and incorrect assignment detection. Our approach, which secured third place in the KDD Cup 2024 OAG-Challenge WhoIsWho-IND task, leverages the strengths of both paradigms to achieve robust performance.

The machine learning component, particularly the LightGBM model, proved highly effective in extracting and utilizing nuanced features from the textual data. By engineering a diverse set of features, including text embeddings, statistical attributes, and cross-paper comparisons, we were able to capture complex patterns that differentiate an author's genuine works from incorrectly assigned papers. The importance of these engineered features, especially the distance metrics between author and paper representations, underscores the value of domain-specific feature crafting in academic text mining tasks.

Complementing this, the application of large language models (LLMs) through instruction fine-tuning showcased their remarkable capability in understanding and reasoning about academic text. By fine-tuning ChatGLM3, GLM4-Chat, and Mistral-7B models, we harnessed their pre-trained knowledge and adapted it specifically to the nuances of author-paper relationships. This approach allowed for a more contextual and nuanced analysis of the content, capturing subtle aspects of authorship that may be challenging to encode explicitly in feature-based models.

The fusion of these two approaches – traditional machine learning and fine-tuned LLMs – proved to be greater than the sum of its parts. By normalizing and merging the predictions from both paradigms, we achieved a more robust and accurate system for detecting incorrectly assigned papers. This synergy highlights the potential for hybrid approaches in tackling complex text mining tasks in the academic domain.

Our work contributes to the growing body of evidence supporting the efficacy of instruction-tuned large language models in specialized domains. It also reinforces the continued relevance of traditional machine learning techniques, especially when combined with cutting-edge NLP models. As the volume and complexity of academic publications continue to grow, such hybrid approaches offer promising avenues for improving the accuracy of author disambiguation systems, ultimately contributing to the integrity and efficiency of scientific knowledge management.

ACKNOWLEDGMENTS

We would like to extend our deepest gratitude to the organizers of KDD Cup 2024 for their outstanding efforts in making this competition a success. Their dedication and meticulous planning have resulted in a high-quality, challenging event that has greatly enriched our experience. We are particularly grateful to the data providers for supplying the invaluable datasets that were essential for our participation. Additionally, we appreciate the contributions of fellow competitors who shared their baselines and insights, fostering a collaborative and competitive environment. Thank you to everyone involved for making this an unforgettable experience.

REFERENCES

[1] Chen, Bo, et al. "Web-scale academic name disambiguation: the WhoIsWho benchmark, leaderboard, and toolkit." Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023.

[2] Zhang, Fanjin, et al. "OAG-Bench: A Human-Curated Benchmark for Academic Graph Mining." arXiv preprint arXiv:2402.15810 (2024).

[3] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[4] Liu X, Yin D, Zheng J, et al. Oag-bert: Towards a unified backbone language model for academic knowledge services[C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022: 3418-3428.

[5] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[J]. Advances in neural information processing systems, 2017, 30.

[6] Team G L M, Zeng A, Xu B, et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools[J]. arXiv e-prints, 2024: arXiv: 2406.12793.

[7] Jiang A Q, Sablayrolles A, Mensch A, et al. Mistral 7B[J]. arXiv preprint arXiv:2310.06825, 2023.

[8] Longpre S, Hou L, Vu T, et al. The flan collection: Designing data and methods for effective instruction tuning[C]//International Conference on Machine Learning. PMLR, 2023: 22631-22648.