Use Large Language Models for Named Entity Disambiguation in Academic Knowledge Graphs

Shaojun Liu[1,2*], Yanfeng Fang[1,2]

[1]Fujian Institute of Scientific and Technological Information, Fuzhou, 350001, China
[2]Fujian Provincial Key Laboratory of Information and Network, Fuzhou, 350001, China

Corresponding author. Email: liusj@fjinfo.org.cn; afang@fjinfo.org.cn

Abstract. This study investigates the application of large language models (LLMs) in disambiguating homonymous named entities in academic knowledge graphs. Current state-of-the-art methods rely on supervised learning techniques that often necessitate extensive annotated datasets, which may be scarce in specialized domains. For further exploration, we constructed an academic knowledge graph in the science and technology domain using publicly available data and extracted contrasting homonymous named entities from different projects to create a test dataset. We evaluated the performance of the ChatGPT model on this dataset using zero-shot, in-context, and chain-of-thought prompting strategies. The experimental results reveal that while LLMs achieve limited success in a zero-shot setting, chain-of-thought prompting can enhance their reasoning abilities. However, a performance gap persists when compared to supervised learning methods specifically trained on the dataset. These findings suggest that LLMs, such as ChatGPT, present a promising direction for assisting in knowledge graph construction for named entity disambiguation, particularly when labeled data is scarce. The utilization of LLMs could be especially beneficial for domains lacking extensive annotated datasets, offering a competitive alternative for disambiguating homonymous named entities.

Keywords: Large language models; Named entity disambiguation; Academic knowledge graphs; ChatGPT; Chain-of-thought

1 Introduction

Disambiguating named entities with the same name is a critical challenge in constructing and organizing knowledge graphs, particularly in the domain of academic knowledge graphs. Named entities, such as companies, organizations, and individuals, often share common names, leading to ambiguity when distinguishing between distinct entities within a knowledge graph [1]. This issue is especially prevalent in academic knowledge graphs, where multiple researchers, institutions, or publications may bear the same name but represent distinct entities. Accurate disambiguation is

G. Guan et al. (eds.), *Proceedings of the 2023 3rd International Conference on Education, Information Management and Service Science (EIMSS 2023)*, Atlantis Highlights in Computer Sciences 16,
https://doi.org/10.2991/978-94-6463-264-4_79

crucial for ensuring the reliability of information retrieval, data integration, and network analysis in various applications [2].

Current state-of-the-art methods for disambiguating named entities primarily involve measuring the similarity between entity nodes to distinguish them. These approaches often rely on supervised learning techniques using neural networks to model similarity. For example, Basile et al. proposed a deep recurrent network approach to resolve company name ambiguity by employing a Siamese Long Short-Term Memory (LSTM) network. This method extracts embeddings of company name strings in a relatively low-dimensional vector space through supervised learning. These embeddings can then be utilized to identify pairs of company names that represent the same entity [3]. Despite advancements in current methods for disambiguating named entities, the complexity of real-world heterogeneous data often results in performance limitations. Consequently, researchers have proposed leveraging human intelligence to enhance the disambiguation process. For instance, Ferreira et al. introduced "AuthCrowd," a crowdsourcing system designed to tackle author name disambiguation and entity matching by decomposing tasks for crowd workers. Experimental results on a real-world dataset of publicly available papers published in peer-reviewed venues demonstrate the potential of this approach to improve author name disambiguation [4].

Given the recent success of large-scale Language Models (LLMs), which have demonstrated human-like capabilities in various tasks [5], their potential application in constructing knowledge graphs has generated significant interest. LLMs can efficiently perform tasks such as entity recognition, relation extraction, and fact verification, thereby contributing to the automatic generation and enrichment of knowledge graphs [6]. In their roadmap for unifying LLMs and knowledge graphs, Pan et al. propose three general frameworks: KG-enhanced LLMs, LLM-augmented KGs, and synergized LLMs + KGs. These frameworks aim to leverage the strengths of both LLMs, like ChatGPT and GPT-4, and structured knowledge models like knowledge graphs, to enhance their capabilities and address their limitations. Recently, Peeters and Bizer (2023) investigated the use of ChatGPT for entity matching, demonstrating its competitiveness with traditional Transformer models. Their study showed that ChatGPT achieved a zero-shot performance of 82.35% F1 on a challenging matching task while also benefiting from in-context demonstrations and higher-level matching knowledge. These findings suggest that ChatGPT can significantly contribute to the efficiency and effectiveness of entity matching in knowledge graph construction [7].

Given the significance of disambiguating homonymous named entities in constructing academic knowledge graphs and the frequent absence of accurate annotation data, we delve into the possibilities offered by Large Language Models (LLMs) to tackle this issue. Our inquiry focuses on determining if the extensive textual knowledge inherent in LLMs, combined with their contextual understanding and information extraction capabilities, can efficiently resolve named entity ambiguity without depending on copious labeled data.

2 Dataset Construction and Preprocessing

In our study, we aim to construct an academic knowledge graph using publicly available research data. The data sources for the graph include research projects and talent information, with the primary data consisting of research project records. A significant portion of the nodes in the constructed graph exhibit the challenge of homonymous named entity disambiguation, particularly for nodes representing researcher names. Each researcher's name in the research projects is accompanied by a unique identifier, allowing us to differentiate between individuals sharing the same name. We exploit this data to create a named entity disambiguation dataset to assess ChatGPT's performance on this specific task.

The research project dataset comprises a total of 636,324 projects, involving 50,997 unique PIs, encompassing 48,386 unique names. Out of these, 40,227 PIs share their names with at least one other PI, and the most frequently occurring PI name is shared by 262 distinct researchers.

To evaluate the disambiguation capability of LLMs, we consider the scenario where the same name appears in different projects and assess whether it represents the same individual. The number of pairwise comparisons involving the same name appearing in distinct projects can be calculated using the combination formula, resulting in a substantial figure of 17,347,659 total comparisons for all names and projects. In practical applications, not all pairs require comparison; a project only needs to be compared with a well-matched project from a pool of same-named individuals. To assess LLMs' disambiguation ability, we randomly select 10,000 pairwise comparisons from the extensive pool to compose our test dataset. Among these 10,000 comparisons, 9,003 pairs involve the same named participant, while 997 pairs concern different individuals sharing the same name. To establish a performance baseline, we employ a supervised learning method, necessitating an additional 1,400 pairwise comparisons for training purposes. The data samples are presented in Table 1, while the data distribution can be observed in Table 2.

Table 1. Sample of Named Entity Disambiguation Dataset for Research Projects

Name	*excuOrganName*	*plan Year*	*…*	*Research Attribute*	*research-Field*	*re-search-Type*	*planA mt*	*isSame-Person*
Guo Bocheng	Institute of Materials Science and Engineering	2003	…	Applied research	Material technology	Academic grant	1,422,000	1
	Institute of Materials Science and Engineering	2002	…	Applied research	Material technology	Academic grant	1,048,000	
Chen Zhifeng	Department of Animal Science, National Chung Hsing University	2018	…	Basic research	Animal husbandry and veterinary medicine	Cooperative research	900,000	0
	Department of Chinese Literature, Shixin University	2019	…	Basic research	Chinese	Academic subsidy	517,000	

Table 2. Distribution of Named Entity Disambiguation Dataset for Research Projects

Dataset Type	# *Pairs*	# *Pos*	# *Neg*
test	10000	9003	997
training	1400	1262	138

3 Methodology

The previously constructed dataset consists of rows containing various pieces of information about each research project. As ChatGPT and similar language models primarily operate on textual data, it is essential to transform the tabular data into a human-readable format. For example, the column name 'excuOrganName' is adapted to 'Executing Organization Name,' and 'researchField' is converted to 'Research Field.' We concatenate the column names and values using the term 'is,' and separate individual columns within the same row with commas. An example of a transformed row appears as follows: "Executing Organization Name is Department of Information Engineering, University of Science and Technology, Plan English Name is The Study and Implementation of Agent-Based Hybrid Cloud Environment with QoS Dynamic Resource Allocation, Plan Year is 2010, Plan Start Date is February 1, 2010, Plan End Date is March 1, 2011, Plan Chinese Name is Implementation and Research of Agent-Based Dynamic Resource Allocation with QoS on Hybrid Cloud, Research Category is Technology Development, Research Field is Information Engineering-- Hardware Engineering, Research Type is Academic Grant, Plan Title is Implementation and Research of Agent-Based Dynamic Resource Allocation with QoS on Hybrid Cloud, Plan Amount is 663,000."

After transposing the projects into text, we submit the paired projects to the ChatGPT API for evaluation. The results are predominantly contingent upon the constructed prompt; consequently, we design a fundamental base prompt and incorporate In-Context Learning and Chain-of-thoughts for testing purposes. During the prompt testing phase, we observe that ChatGPT's results are highly susceptible to the influence of non-essential columns. As a result, we derive the most influential column rankings from the supervised learning baseline method and transmute the top three most pertinent columns into text for submission to ChatGPT as a comparison.

In the subsequent subsections, we expound on the specifics of the Supervised Learning Method, Base Prompt, In-Context Learning, Chain-of-thoughts, and Column Selection approaches.

3.1 Supervised Learning Method

The current state-of-the-art approaches for named entity disambiguation predominantly involve supervised learning methods that utilize neural network-based encodings. In the present study, we employ the Luotuo Embedding method as our encoding technique[8]. This generative text embedding model is distilled from the OpenAI API, offering a unique and powerful approach to capturing semantic information in textual

data. This model is meticulously trained by employing a combination of three distinct loss functions: (1) a distillation loss that harmonizes the model's embedding with that of OpenAI's, (2) a KL divergence loss that fosters coherence between the embeddings of interconnected textual data, and (3) a margin loss that mitigates the risk of the model mastering an excessively simplistic task. Experimental evidence demonstrates that their model achieves performance metrics that are on par with OpenAI's state-of-the-art embedding model text-embedding-ada-002, across a diverse range of downstream applications including text visualization, search, and dialogue. After outlining the Luotuo Embedding method as our primary encoding technique, we employed two distinct approaches for encoding the data. The first approach, akin to ChatGPT, involved converting the entire row of data into a single text segment for encoding. The second approach, on the other hand, focused on encoding each column of data individually. This latter method proved to be more effective in extracting the distinct influences of different columns on the results within the labeled datasets.

Upon vectorizing the data using these two encoding approaches, we proceeded to train a Random Forest model on the training set and evaluated its performance on the test set. The purpose of this step was to compare the results obtained from the Random Forest model with those achieved using ChatGPT. This comparison allowed us to assess the effectiveness and validity of our chosen encoding techniques in the context of named entity disambiguation tasks.

3.2 Base Prompt

Prompt engineering is a crucial aspect of fine-tuning large-scale language models for specific tasks. It involves designing effective input queries or statements that enable the model to generate desired outputs[9]. Crafting well-structured prompts can significantly improve the performance of models like GPT-3, allowing them to produce more accurate and coherent responses. The careful consideration of the phrasing, context, and format of the prompt can greatly influence the model's understanding and response generation, leading to better task performance.

In our task, we encountered several challenges due to the limited information provided to the large language model (LLM), which needed to return a definitive result. When directly providing project information and asking for a judgment, the LLM might indicate that more information is needed before making a decision. In cases where a clear result is required, the LLM is more likely to return an "uncertain" response. Moreover, considering that the test set consists of 10,000 records, employing manual evaluation would be too labor-intensive.

To address these issues, we explicitly requested the LLM to assess probabilities in the prompt, and then returned results based on the assessed probabilities. If the probability fell within the categories of "Very likely," "Highly probable," "Likely," "Probable," or "Possible," we returned a value of 1. If the probability was categorized as "Unlikely," "Improbable," "Highly improbable," "Very unlikely," or "Impossible," we returned a value of 0. The system content was phrased as follows: 'As a scientist, I need your expert opinion on two projects and a specific individual involved in them. '

The user content was formulated as: 'Given the following information about two research plans:

{delimiter}

Plan 1: {str(porject1)}

Plan 2: {str(project2)}

{delimiter}

Based on the provided information, can we conclude that the person named {name} in both plans is the same individual, or are they two different people with the same name? please consider the possibility of in both plans is the same individual, Respond with 1 or 0:

0 - If the possibility is "Unlikely" or "Improbable" or "Highly improbable" or "Very unlikely" or "Impossible".

1 - If the possibility is "Very likely" or "Highly probable" or "Likely" or "Probable" or "Possible".

Output a single character.'

Using the aforementioned prompt structure, we were able to ensure that the LLM consistently generated output in the form of either 1 or 0 as the final result. This design effectively streamlined the response generation process, enabling the model to produce clear and definitive answers based on the assessed probabilities.

3.3 In-Context Learning

In-Context Learning is an essential technique employed by large-scale language models to adapt their understanding and responses to the given context. This learning paradigm allows models, such as GPT-3, to leverage the contextual information embedded in a sequence of tokens and make accurate predictions based on the surrounding text[5]. In-Context Learning plays a vital role in enhancing the performance of language models for various tasks, including question-answering, sentiment analysis, and summarization, by providing them with the necessary context for generating more coherent and contextually relevant responses.

In our task, we provide judgment examples in the prompt. However, due to the maximum token length of GPT-3.5 being 4097 and the average token count of our project information being around 400, we cannot provide too many examples. Therefore, we have randomly selected two positive examples and two negative examples from our training set to serve as prompts. The format is to add the following before the user content:

'For example：

{Two plan information as in Base Prompt}

Result: The names of the persons in charge of the two plans are both ***, they are the same person.

{Two plan information as in Base Prompt}

Result: The names of the persons in charge of the two plans are both ***, but they are different people . '

3.4 Chain-of-thought

In recent research, Wei et al. proposed an innovative method called Chain-of-Thought Prompting to enhance the reasoning capabilities of large language models. This approach revolves around generating a chain of thought, which consists of intermediate natural language reasoning steps that ultimately lead to the final output. By incorporating a few chain of thought demonstrations as exemplars in the prompting process, the authors demonstrated that this technique significantly improves the performance of language models on various arithmetic, commonsense, and symbolic reasoning tasks. Notably, Chain-of-Thought Prompting outperforms standard prompting methods and achieves state-of-the-art accuracy on benchmarks such as the GSM8K math word problems. This method offers a promising avenue for unlocking the full potential of large language models in reasoning tasks, while also providing an interpretable window into their behavior[10].

In our task, considering that we do not have a precise understanding of the impact of each column on the results, we aim to avoid potential biases from flawed thought processes. To achieve this, we simply add a sentence after the question: "Let's do it step by step." Experimental results reveal that adding this sentence may lead to more concise reasoning steps when we only require the final response to be 1 or 0. Consequently, we have removed the requirement for a specific response format from the prompt and shifted to using multi-turn dialogues. After receiving the reasoning result, we request ChatGPT to summarize the outcome as 1 or 0, akin to the basic prompt. Although this approach significantly increases token consumption, empirical evidence indicates that it effectively utilizes the Chain-of-Thought reasoning process.

3.5 Column Selection

Considering that when only returning 1 and 0 as results, it is evident that ChatGPT can be easily influenced by variations in columns that have minimal impact on the actual outcome. Without the effect of the Chain-of-Thought, the model may lack the complex reasoning ability to accurately distinguish between the effects of important columns and those of less significant ones. Therefore, we decided to leverage the results of supervised learning to obtain the importance ranking of the columns and then experiment with providing only the important columns to ChatGPT.

We used code to visualize the feature importances of a trained Random Forest classifier. The process involved extracting the feature importances from the classifier and sorting them in descending order. This visualization is useful for understanding the relative importance of each feature in the model, which can aid in feature selection and model interpretation in the context of the research. The results are shown in Fig. 1. It is clear that the impact of the first four columns is far greater than that of the subsequent columns. Considering that the influence of 'Plan Chinese Name' and 'Plan English Name' might overlap, we selected 'Executing Organization Name', 'Plan Chinese Name', and 'Research Field' as the columns. We then conducted experiments using the Base Prompt, In-Context Learning, and Chain-of-Thought approaches, following the methods previously described.

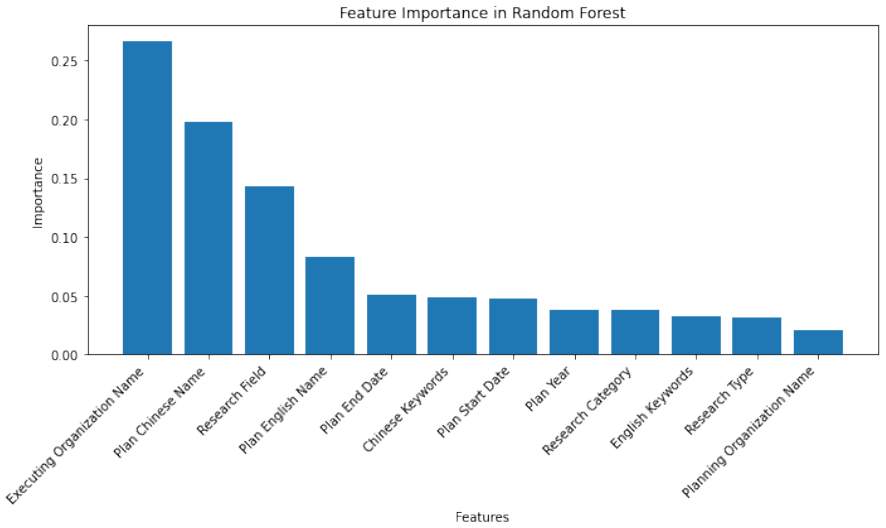

Fig. 1. Feature Importance Ranking

4 Experimental Results

We conducted experiments using OpenAI's gpt-3.5-turbo-0613 model, setting the temperature parameter to 0 to obtain deterministic responses. The API rate limits for RPM were 3,500 and TPM were 90,000. During the experiment, the RPM limit was not reached, but the TPM limit was occasionally hit. The call duration was affected by the number of tokens, with Base Prompt and In-Context Learning experiments completed within a day, while Chain-of-Thought experiments required more than three days due to longer responses and two-turn dialogues. We will now discuss the experimental results and subsequent analysis.

4.1 Results

The disambiguation problem of named entities with the same name differs slightly from other issues like entity matching, as it requires considering not only the precision of positive examples but also that of negative ones. We used macro precision, macro recall, and macro F1 to evaluate the performance of the methods, along with the average cost of calling the API. Costs were calculated based on the number of tokens: input pricing was $0.0015 per 1K tokens, and output pricing was $0.002 per 1K tokens. The results are shown in Table 3.

Table 3. Experimental results and associated costs

Method	Results						
	Macro P	***Macro R***	***Macro F1***	***ΔF1***	***Cost(¢) Per pair***	***Cost Increase***	***cost increase per ΔF1***
ChatGPT-zeroshot	54.91	62.87	52.59	-	0.13	-	-
ChatGPT-in-content	56.28	67.45	53.15	0.56	0.51	304%	542%
ChatGPT-think-chain	66.35	82.36	69.84	17.25	0.33	163%	9.46%
Select-zeroshot	85.23	58.75	62.53	9.94	0.06	-54%	-5.52%
Select- in-content	64.06	79.18	66.79	14.20	0.22	73%	5.14%
Select- think-chain	65.41	76.81	68.52	15.94	0.21	68%	4.28%
Embeddings-rf	69.12	68.44	68.77	16.18	-	-	-
Multi-embeddings-rf	87.49	79.90	83.16	30.57	-	-	-

4.2 Discussion

From the experimental results, it can be seen that the performance of ChatGPT using zero-shot and requiring direct return of results is very limited, clearly affected by the interference of columns with little impact on the results. Even when using In-Context Learning to increase example prompts, the performance only shows a slight improvement. The significant performance improvement after using only important columns indicates that providing a large amount of information in the case of direct return does not help the model make more accurate judgments.

A substantial performance improvement was observed when using the Chain-of-Thought approach, suggesting that tasks requiring reasoning abilities indeed achieve better results with this method. Comparing the supervised learning results of random forests after row vectorization, the performance of the Chain-of-Thought approach is slightly better, indicating that it can indeed make accurate judgments based on textual information. However, there is still a gap compared to the performance of supervised learning using random forests with column vectorization. This difference may arise from various factors, including the data's inherent characteristics, the model's language understanding, and its ability to reason about specific domain knowledge. In the case of the data itself, the impact of each label on the result is specific and may exhibit complex relationships. For example, when using logistic regression for supervised learning, specific weights can be learned for each label, which heavily depend on the dataset's characteristics. This nuanced understanding of the data is challenging for ChatGPT to capture, as it is trained on general-purpose corpora and might not have exposure to the specific domain or dataset.

Furthermore, while ChatGPT has shown remarkable performance in various natural language understanding tasks, its reasoning capabilities in specialized domains might still be limited compared to supervised learning models explicitly trained on those domains. Despite this, considering the relatively low cost of each comparison using the Chain-of-Thought approach, it remains a competitive alternative in situations where labeled data is scarce.

5 Conclusion

In this study, we explored the potential of leveraging large language models (LLMs) to disambiguate homonymous named entities in academic knowledge graphs. The experimental results demonstrate that while models like ChatGPT show promise in this task, their performance is still limited compared to supervised learning methods. The Chain-of-Thought prompting approach was shown to improve ChatGPT's reasoning abilities, allowing it to surpass supervised learning methods that use entire lines of text. However, there still exists a performance gap compared to models trained specifically on the dataset using structured data. The main factors contributing to this performance gap include:

(1) The data's inherent complexity and specific characteristics which are difficult for general-purpose LLMs to capture.

(2) The LLMs' limited understanding of the domain and exposure to similar data.

(3) The LLMs' relatively restricted reasoning capabilities in specialized domains compared to models explicitly trained on those domains.

Overall, while LLMs exhibit some success in tasks requiring zero-shot or few-shot learning, their performance is still constrained for complex reasoning problems that benefit from explicit training on nuanced data. Further research into enhancing LLMs' domain-specific knowledge and training them jointly with specialized knowledge graphs may help bridge the current performance gap. Nevertheless, LLMs represent a promising direction for assisting in knowledge graph construction, particularly for tasks requiring common-sense or textual reasoning, or in situations where labeled data is scarce, making them a competitive choice.

Acknowledgment

This work was supported by the Fujian Provincial Department of Science and Technology's External Cooperation Project (Grant No. 2022I0025).

References

1. Navigli, R., and Ponzetto, S. P. (2012) BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, 193: 217-250.
2. Moro, A., Raganato, A., and Navigli, R. (2014). Entity linking meets word sense disambiguation: A unified approach. Transactions of the Association for Computational Linguistics, 2: 231-244.
3. Basile, A., Crupi, R., Grasso, M., Mercanti, A., Regoli, D., Scarsi, S., Yang, S., and Cosentini, A. (2023). Disambiguation of Company names via Deep Recurrent Networks. arXiv preprint arXiv:2303.05391. [https://doi.org/10.48550/arXiv.2303.05391]
4. Correia A, Guimarães D, Paulino D, et al. Authcrowd: Author name disambiguation and entity matching using crowdsourcing[C]//2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2021: 150-155.

5. Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
6. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2023). Unifying Large Language Models and Knowledge Graphs: A Roadmap. arXiv preprint arXiv:2306.08302v2. [https://doi.org/10.48550/arXiv.2306.08302]
7. Peeters, R., and Bizer, C. (2023). Using ChatGPT for Entity Matching. Proceedings of ADBIS 2023. arXiv preprint arXiv:2305.03423v2. [https://doi.org/10.48550/arXiv.2305.03423]
8. Liu, S., Leng, Z., Huang, H., Chen, S., Hu, J., Sun, A., Chen, Q., and Li, C. (2023). Luotuo Embedding: Generative Text Embedding Model distilled from OpenAI API. GitHub repository. Retrieved from https://github.com/LC1332/Luotuo-Text-Embedding
9. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2021). Improving Language Understanding by Generative Pre-Training. OpenAI. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
10. Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in Neural Information Processing Systems, 2022, 35: 24824-24837.