Knowledge-Enhanced Academic Chatbot: Harnessing Large Language Models and Knowledge Graphs

Anonymous ACL submission

Abstract

In this paper, we introduce an academic chatbot designed to help identify relevant publications, authors, and affiliations in academia and applied innovation. Leveraging similarity thresholds and query transformations, the chatbot delivers answers that are informed by an academic knowledge base. We address the challenges of efficiency, reproducibility, and interpretability through a combination of rule-based solutions and large language models backed by knowledge-graph embeddings. We have also critically analyzed the output of our chatbot and discussed various future directions of improvement.

1 Introduction

In academia as well as in applied innovation-related work one is frequently faced with selection processes to identify relevant publications in academic outlets or patents; authors (e.g., peers of a similar age and research portfolio); affiliations/institutions (e.g., departments at universities or in companies with a similar specialization), etc.

Examples, where such search problems emerge are concerns about the originality of to-be- or already-published research of an author, with the identification of young job market candidates for open assistant professorships or researcher positions, with the evaluation of tenure candidates and the need of comparisons with peers of a similar age cohort and research orientation as regularly occurring for university departments or tenure committees, with the identification of the most relevant prior research on a topic, etc.

The current state of affairs with such search problems is at best a mix of machine-assisted procedures (mostly for checking originality and plagiarism of research) and selections by humans, where the latter may be subjective and limited in scope as well as accuracy.

While for many of the mentioned problems one might wish to not fully rely on machine-assisted procedures, having well-specified algorithms at hand which support the search is highly warranted. E.g., one problem that could be avoided with a broad machine-assisted search is that much larger data pools can be searched systematically than this is the case with human search alone. Also, unwarranted elements of subjectivity in the pre-selection and truncation of available data samples to be made available to relevant editorial, selection, and search committees or individuals can be avoided.

In this paper, we present a tool which can help identifying relevant samples from raw academic publication data as contained in academic graphs such as OpenAlex (Priem et al., 2022) and AIDA (Meloni et al., 2023a). The tool, which can be dubbed an "*academic chatbot*", provides an exhaustive list of relevant data based on a threshold value of similarity between user-defined inputs and the ones found in knowledge graphs (KGs).

For users to interact with this bot, the primary input are the anchor words in an academic graph, such as a publication, an author, an affiliation, or a topic. Then the chatbot performs query transformations following the steps visualized in Fig. 1. The output to the user is the query plan the chatbot follows, as well as natural language statements of answers to the user's question(s). The chatbot delivers these two types of output in that the former (query plan) assists the latter (answers). From the user's perspective, complexity in query transformation is hidden in and internal to the chatbot. We perform query transformations by identifying anchor words, converting user statements to SPARQL queries to a KG, creating the embedding of the KG that facilitates the reasoning in the KG, with which the chatbot can answer user queries.

Building such a chatbot has challenges of *efficiency*, *reproducibility*, and *interpretability*. We have designed solutions to mitigate them. To in-

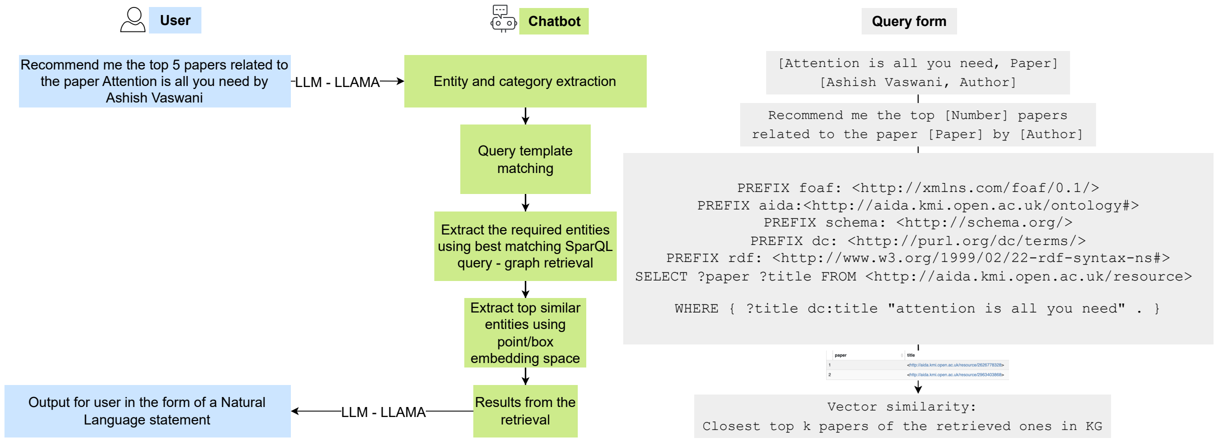

Figure 1: Our pipeline of chatbot design.

crease *efficiency* and *reproducibility*, we have incorporated solutions which are both rules-based and backended by Large Language Models (LLMs) as well as a KG embedding method that is easy to retrain and adapt. To enable *interpretability*, we provide users with a query plan output and conduct internal and external evaluations regarding our output and those of the academic LLMs and DBs.

The remainder of the paper is organized as follows. In Section 2 we discuss existing literature and tools related to building an academic chatbot. Section 3 is devoted to a description of the design of the proposed academic chatbot. We present several applications to underpin the usefulness of the tool in Section 4 and indicate how to critically evaluate its performance. We reflect on the chosen approach in Section 5 before concluding with a brief summary. We plan to open-source the chatbot through an anonymized github repository at `https://anonymous.4open.science/r/Academic_Chatbot-F9EB/`.

2 Related Work for Building Academic Chatbots

Text-to-SPARQL. Early work on translating natural language to SQL, such as SQLNet (Xu et al., 2017) and Seq2SQL (Zhong et al., 2017), laid the groundwork for database querying using natural language. Recent advancements, particularly with Large Language Models (LLMs), have significantly improved Text-to-SQL capabilities, with notable contributions by Gao et al. (2023) and Roberson et al. (2024), demonstrating the LLMs' flexibility and superior performance. Text-to-SPARQL extends these principles by incorporating ontologies, with methodologies evolving from rule-based approaches (Ochieng, 2020; Kondylakis et al., 2020) to leveraging pre-trained language models for enhanced semantic parsing (Banerjee et al., 2022). A notable application in the life sciences by Rangel et al. (2024) utilizes OpenLLaMA, fine-tuned with domain-specific KGs, to improve question-answering through SPARQL.

KG Embeddings and Reasoning. KGs such as DBpedia (Lehmann et al., 2015), SemOpenAlex (SemOpenAlex, 2024), and domain-specific KGs like MeSH are pivotal in AI for information retrieval and interaction, particularly chatbots. KGs combine symbolic information with the need for continuous representations in deep learning through embedding. Embeddings transform KG entities (h, r, t for head, relation, tail) into a vector space where related entities are positioned closely, facilitating queries in first-order logic.

The embedding learning objective is to optimize a function $\mathcal{L}$ that maximizes the proximity of true triples (h, r, t) in the KG and minimizes it for false ones, incorporating techniques like negative sampling. This approach enables the representation of KGs in a continuous space, with TransE (Bordes et al., 2013) noted for its computational efficiency.

For reasoning over KGs, especially across multi-hop relations, traditional point embeddings face scalability issues. Alternative approaches like Query2Box (Ren et al., 2020) suggest embedding queries as hyper-rectangles, or "boxes", which can more easily encompass answer nodes, mitigating the computational cost of dense KG traversal.[1]

Academic Knowledge Graphs and Chatbots.

[1] Note that there are also other box embedding methods beyond the aforemenetioned ones; see Abboud et al. (2020); Cao et al. (2022).

Despite the existing databases and tools related to our research agenda, we found no open-source conversational agent tailored to the specific queries on authors and publications our project addresses.

SemOpenAlex (Färber et al., 2023), succeeding Microsoft Academic Service, offers an academic graph overview with 1-hop connections (e.g., coauthors, author publications, affiliations) but lacks a recommendation engine for multi-hop queries. Although it paved the way for constructing an academic KG that supports querying and fact verification, its SPARQL endpoint is currently unavailable. The chatbot proposed here enhances this by offering a possible solution to multi-hop queries and offering a prototype for recommending papers, authors, fields, and affiliations, by integrating KGs.

Meloni et al. (2023a,b) developed AIDA-Bot 2.0, a proprietary chatbot that supports a range of predefined queries and open questions through transformer-based models. It uses an extensive database of scholarly publications and patents, but the underlying data are still much smaller than the respective raw data sources available (e.g., OpenAlex's 240 million publications and PATSTAT Global's 100 million patents). AIDA dashboard visualizes research metrics on venues but is not designed for recommendations. Our work introduces an open-source chatbot prototype with the possibility of extending to OpenAlex and PATSTAT for a more comprehensive academic inquiry.

3 Design of the Academic Chatbot

We illustrate the various stages in designing the chatbot in Fig. 1.

3.1 Query Templates

We predefine three templates for our academic chatbot to recommend articles, authors, and affiliations.

(1) Recommend the top papers similar to "Paper Name". Example: Recommend the top-5 papers related to the paper "Attention is All You Need" by Ashish Vaswani.

(2) Recommend the top authors similar to "Author Name". Example: Recommend the top-5 authors similar to Ashish Vaswani.

(3) Provide the top affiliations similar to "Affiliation Name". Example: Return the top-5 institutions similar to MIT.

3.2 Extracting Entity and Category from User Queries

In this paper, we primarily focus on three categories of entities related to academic papers: AUTHOR, AFFILIATION, and PAPER (the title of an academic paper).[2] We employ a two-phase mechanism to identify the entities in a user query.

In the first phase, we utilize the English transformer-based (Vaswani et al., 2017) pipeline en_core_web_trf[3] by spaCy (Honnibal et al., 2020) to perform the classical Named Entity Recognition (NER) task on user queries. SpaCy v3 provides efficient processing capabilities and a user-friendly API, making it highly suitable for precision-focused applications, even when dealing with smaller datasets, with accuracy up to the current state-of-the-art.[4] We constrain the model to recognize only three standard NER labels: PERSON, ORG, and WORK_OF_ART. Given the context of academic papers, we posit that these three labels align with the chosen entities of interest AUTHOR, AFFILIATION, and PAPER, respectively. In preliminary experiments, we observed that the recognition accuracies for PERSON and ORG are notably high. However, when treating an entire paper title as a single entity, the model, in most cases, did not mark this span as WORK_OF_ART. It often did not recognize it altogether or could only identify parts of the title. We speculate that this is due to the scarcity of academic papers in the training corpora, as WORK_OF_ART originally is used in the context of non-academic literature or music.

To address this issue, we introduce a second phase, where we specifically designate Llama-2-7b (Touvron et al., 2023) to recognize the PAPER entities. In our prompt template, we include several examples to facilitate in-context learning, as detailed in Fig. 2. This approach eliminates the need for model fine-tuning, thereby enhancing the versatility of our methodology. In a preliminary test involving 20 samples from various subfields of computer science, the model has accurately identified 16 paper titles. The output results are combined with WORK_OF_ART entities in

[2] The respective entities are the actual names of these categories.

[3] Model card is available at https://github.com/explosion/spacy-models/releases/tag/en_core_web_trf-3.7.3 (last accessed: Feb 1, 2024).

[4] See details under https://spacy.io/usage/facts-figures#benchmarks (last accessed: Feb 1, 2024).

```
You are an expert Named Entity Recognition (NER) system. Your task is to accept
Text as input and extract named entities. Entities must have one of the following labels:
{PAPER}. If a span is not an entity label it: ‘==NONE==‘.
Here is an example of the output format for a paragraph using different labels than this
task requires. Only use this output format but use the labels provided above instead of
the ones defined in the example below. Do not output anything besides entities in this
output format. Output entities in the order they occur in the input paragraph regardless
of label.
Paragraph: Who are the authors of the paper Chain-of-Note: Enhancing Robustness in
Retrieval-Augmented Language Models?
Answer:
1. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models |
True | PAPER | is an intact paper title
2. Chain-of-Note | False | ==NONE== | is only part of a paper title
3. Enhancing Robustness in Retrieval-Augmented Language Models | False | ==NONE==
| is only part of a paper title
......
Paragraph: {text}
Answer:
```

Figure 2: Few-shot learning prompt template for an LLM to recognize custom entities (PAPER). {text} should be filled with a user query.

the previous phase and deduplicated to form the final set of recognized PAPER entities.

3.3 Mapping User Queries to SPARQL templates in a KG

In addressing the challenge of mapping user queries to SPARQL templates in a KG, we developed a rule-based step designed to interpret and convert natural language queries into structured SPARQL queries. This step leverages a combination of linguistic patterns and entity recognition to understand the intent behind a user's query and to map this intent to one of several predefined SPARQL query templates.

Linguistic Pattern Recognition. The first component of this step involves the identification of key phrases and structures within the user's query. E.g., queries containing phrases such as "similar to" or "related to" are interpreted as requests for works similar to a given entity; phrases such as "authored by" or "written by" indicate a search for papers by a specific author; terms such as "cited by" suggest a request for papers that cite a particular work. Recognizing these patterns, our chatbot can determine the type of information the user is seeking.

Entity and Category Mapping. Upon identifying the intent of a query, the chatbot next identifies the relevant entities within the query, categorized as either AUTHOR, AFFILIATION, or PAPER. This categorization is achieved through the integration of an NER step as in Section 3.2, which tags entities in the query according to their recognized category.

SPARQL Query Generation. With the intent and entities identified, the chatbot then maps the query to a corresponding SPARQL query template. This mapping is facilitated by a set of predefined templates that are designed to cover the range of query intents recognized by the chatbot. Each template is structured to retrieve data from the KG based on the category of entity identified and the specific intent of the query. E.g., a query seeking papers similar to a given work (PAPER entity) would be mapped to a template designed to search the KG for papers with titles or topics closely related to that of the specified work. Similarly, a query inquiring about publications by a certain author (AUTHOR entity) would be mapped to a different template, specifically crafted to retrieve works authored by the identified individual.

Rule Based vs. LLMs. We have discussed in Section 2 the various approaches to convert natural language statements to SPARQL queries. As the first iteration of our academic chatbot, we have chosen rule-based heuristics that allow us to retrieve entities in a KG more precisely. Although LLMs could be a more efficient way in the future to directly perform the conversion from natural statement to SPARQL query, we plan to build this component step by step: (1) to use an LLM to map user queries into our pre-defined templates and then to use our rule-based heuristics to generate SPARQL queries, (2) to use a fully automatic LLM to generate SPARQL queries, such as Retrieval Augmented Generation (RAG) powered LLMs, similar to Rangel et al. (2024). In our tests (see Appendix B), the generated SPARQL queries by ChatGPT-3.5 were close to the reference. However, since the proposed chatbot will be open-sourced, we will not use ChatGPT-3.5 for RAG development.

3.4 Extracting KG Subgraphs using SPARQL

After expressing the user query in SPARQL format, it is conducted through the AIDA public SPARQL endpoint.[5] As we see in the query transformation in Fig. 1, the SPARQL query generated based on user input will be sent to the AIDA KG to locate the entities in the KG based on which we perform the embedding search.

3.5 Reasoning on KG in Vector Space

Enhancing the backend pipeline of AI chatbots, particularly for academic recommendations, significantly relies on vector-space reasoning. Embedding the AIDA KG into a vector space is crucial for enabling sophisticated and effective reasoning.

[5] https://aida.kmi.open.ac.uk/sparqlendpoint (last accessed: 24 January, 2024).

TransE. The TransE model is formally defined as $h + r \approx t$, where h and t are the vector embeddings of the head and tail entities, respectively, and r is the embedding of the relation linking them. TransE (Bordes et al., 2013) is a popular technique for KG embeddings, which optimizes a contrastive loss function. This function evaluates the embeddings by comparing the sum of the head entity and relation vectors against the tail entity vector, aiming to minimize the distance between them for positive triples and maximize it for negative triples.

Generating AIDA TransE Embedding Space. The following procedure is used to generate TransE embedding space for the academic chatbot. We obtain the collection of the entire AIDA KG[6] in the form of a set of turtle files (*.ttl), which contain all AIDA triples. Due to the large size of the entire graph, we select a subset of the graph and create a separate train, test, and validation split based on this subset. Specifically, the whole AIDA KG features around 26 million nodes and 19 different relation types,[7] from which we randomly selected 98,303 triples. The selected triples were then divided into 80-10-10 percentage bins for training, validation, and testing, respectively.

Following the documentation of the `Pytorch Geometric` implementation on TransE embeddings of FB15k, we use the `KGE` library to train the TransE model. The trained model that is used to train the embeddings for the selected subgraph of the AIDA KG features 4,916,100 trainable parameters. The number of trainable parameters necessary to train TransE embeddings for the full graph is quite large, which is why we are experimenting with other promising approaches for future projects. The current approach featuring the previously described subgraph takes around an hour to train the TransE model and create the desired embedding space. The hardware infrastructure used for training is 1x RTX Nvidia A5000 (24GB RAM), 1TB of system RAM, 8 units of AMD EPYC 7313 16-Core processors. Additionally, we used the following training hyperparameters: 100 epochs, a learning rate of 0.01, and a weight decay of 0.

To allow for KG reasoning over several entity types at the same time, we decided to embed the whole KG into the same embedding space. However, it would also be possible to create an embedding space for each entity type separately if

[6] "Research Papers v3.0" from `https://aida.kmi.open.ac.uk/downloads.php` (last accessed: Jan 10, 2024).

[7] See the different relation types in Appendix A.

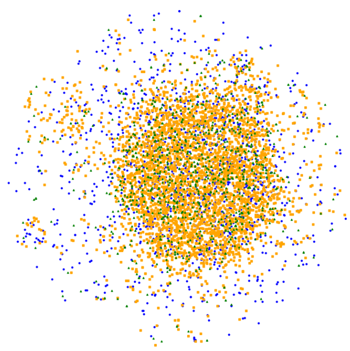

Figure 3: TransE embedding space trained for a subgraph of the AIDA KG visualized using t-SNE dimensionality reduction, displaying the entity types used for reasoning, i.e., `AUTHOR` (blue circles), `AFFILIATION` (green triangles), `PAPER` (orange rectangles).

"cross-type" reasoning is not relevant for the task at hand. Through the application of TransE to create an embedding space, we aim at increasing the distance for embeddings that are dissimilar and decreasing the distance for embeddings that are similar, as outlined in Section 2. The resulting KG embedding, by means of dimensionality reduction using t-SNE (van der Maaten and Hinton, 2008), is visualized in Fig. 3 for the target embeddings we used for KG reasoning in this project (papers, authors, and affiliations). The TransE embeddings result in a space where entities of different types are distributed according to their similarity in the graph structure and not their types, and therefore they are comparable by any vector distance metrics. We also include a t-SNE projection of all types of AIDA entity in Fig. 4.

3.6 Returning Results to the User in Natural Language Statement

A crucial aspect of the KG reasoning process is the translation of complex, structured data into natural language statements. These statements, forming the core of the chatbot's response, are designed to be concise yet comprehensive, providing either direct answers or detailed explanations tailored to the user's query. Additionally, the chatbot's response mechanism may include an overview of the query plan, highlighting the logical steps and reasoning employed by the LLM. This approach not only enhances user understanding but also fosters trust in the system's capabilities. Furthermore, for queries involving interactions with KGs, the chatbot can present details such as the SPARQL query used for data retrieval and the relevant nodes and their interconnections within the KG. This level of detail is particularly beneficial for users seeking deeper insights into the query's processing and the underlying data structure.

More specifically, we utilize the `Llama-2-7b` again to map from the resulting output of reasoning on the AIDA KG embeddings and the initial query to generate a natural language statement that is more user-friendly and human-like than simply returning an array of entity indices. Additionally, the chatbot features an explainability aspect, whereby the user is informed about the specific query plan or, in the current implementation, query template that was used to generate the results that are presented. This serves the purpose of gaining the trust of users in the recommendations provided by the chatbot and, in future work, could be improved by incorporating reinforcement learning with human feedback as suggested by Griffith et al. (2013).

4 Exemplary User Queries and Evaluations

We perform both internal (E1) and external (E2) evaluations of our chatbot performance: (E1) we have manually evaluated the top-k answers using various distance measures; (E2) we compare recommendations from external academic databases, such as SemOpenAlex (SemOpenAlex, 2024), Google Scholar, ChatGPT4 (OpenAI, 2023), Perplexity.ai (Perplexity.ai, 2024), ResearchGPT (Patnaik, 2023), SciSpace (SciSpace.com, 2024), Consensus (Consensus, 2024).

4.1 Internal Evaluation: Distance Measures

We first define the four distance measures we evaluate as follows using two points $\mathbf{u}, \mathbf{v}$ in the embedding space of KG, where u_i and v_i are the i-specific components of $\mathbf{u}, \mathbf{v}$, respectively. These four measures are commonly used in the literature to measure the proximity of points: $\text{dot product}(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i \cdot v_i$, $\text{cosine similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$, $\text{Manhattan distance}(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} |u_i - v_i|$, $\text{Euclidean distance}(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}$.

In Tab. 1 we have listed the top-ranked papers for the paper titled "A Non Revisiting Particle Swarm Optimization" (Chow and Yuen, 2008) in terms of similarity to the user query, measured by distance measures. As discussed in Section 3.5, the recommendations of the chatbot are based on the relations in the KG, associated with the node, edge, and neighborhood information in the KG. The paper in row (1) was ranked by two measures as top-1 and it was published in the same year as the queried paper and at a pertaining IEEE venue. Interestingly, human annotators might filter the recommended results by looking at the semantic "clues" (n-grams). As the paper in the user query and the paper in row (2) share the n-gram "Particle Swarm Optimization", naturally, a human would consider the paper in row (2) a similar paper. Therefore, with this preliminary study on paper recommendation we can conclude that the measure of Euclidean and Manhattan distance agree largely with human annotation relative to others.

We continue the same analysis in terms of queries on similar authors. In Tab. 2 we see that the overlap of authors returned by the various distance measures is substantially smaller than that of papers. The author ("Stefan Kraxberger") in the user query publishes in the domains of IT security, Network Security, Internet of Things, Identity Management, and Mobile Computing, according to Google Scholar. The author in row (1), Kishore Prahallad, specializes in Mobile Computing, which is closely related to the author in the user query. Elias L. Anagnostou, the author in row (4), is a researcher in Structural and Computational Mechanics, which is also closely related to the domains of the queried author. Jinbao Hu, the author in row (8), specializes on Natural Computation, which does not relate to the author domain in the user query. The other authors ranked first, Chen Wei in column (4) and Licai Yang in column (8), work in the domains of Battery Energy and Wireless Computing. The latter is closer to the research domain in the user query. Interestingly, the distance measures in author similarity do not follow the pattern we observe for paper similarity – in general, all distance measures return highly relevant recommendations, as the similar authors are closely connected by venues in the KG, which increases the likelihood that they work in similar domains. Looking at the examples in Tab. 2, dot product and cosine measures deliver more precisely similar candidates.

The affiliation similarity in Tab. 3 has shown patterns similar to the author similarity. The affiliation in the user query, "Suzhou University of Science and Technology", specializes in the domain of Materials Science, Environmental Science, Information Technology, and Medical and Biotechnology.

Table 1: Paper similarity - distance measure scores and ranks. Paper: a non revisiting particle swarm optimization. Underline words are the semantic clue for a human to find the most similar papers. Gray background indicates the largest intersections among the distance measures. Top 1 ranks are marked in **bold**.

	Paper Title	Dot Product		Cosine Similarity		Euclidean		Manhattan	
		Score (1)	Rank (2)	Score (3)	Rank (4)	Score (5)	Rank (6)	Score (7)	Rank (8)
(1)	relation between joint optimizations for multiuser mimo uplink and downlink with imperfect csi			0.473	5	5.518	**1**	26.606	**1**
(2)	a modified membrane-inspired algorithm based on particle swarm optimization for mobile robot path planning					5.918	4	27.270	2
(3)	integrating schemas of heterogeneous relational databases through schema matching							27.321	3
(4)	image processor and image display	17.353	3	0.481	4			27.485	4
(5)	simple garbage bag bracket	17.383	2	0.494	2			27.547	5
(6)	interference cancellation in multiuser acoustic underwater networks using probabilistic sdma					5.828	2		
(7)	on the reversibility of live equal conflict petri nets	16.453	4	0.495	**1**	5.854	3		
(8)	towards an sdn-based architecture for analyzing network traffic in cloud computing infrastructures					5.957	5		
(9)	mergers and acquisitions management new directions editorial	18.831	**1**	0.492	3				
(10)	an efficient algorithm of performing range analysis for fixed point arithmetic circuits based on sat checking	16.138	5						

Table 2: Author similarity - top distance measure scores and ranks. Author: stefan kraxberger.

	Author Name	Dot Product		Cosine Similarity		Euclidean		Manhattan	
		Score (1)	Rank (2)	Score (3)	Rank (4)	Score (5)	Rank (6)	Score (7)	Rank (8)
(1)	kishore prahallad	17.085	**1**	0.503	4				
(2)	tobias j oechtering	16.938	2						
(3)	janusz pawliszyn	16.684	3						
(4)	elias l anagnostou	16.469	4	0.550	2				
(5)	niu xuemin	16.159	5						
(6)	chen wei			0.552	**1**				
(7)	wei jingrong			0.504	3				
(8)	jinbao hu			0.480	5	5.839	**1**		
(9)	vedran sabol					5.865	2		
(10)	sun guanyi					5.870	3		
(11)	darius hedgebeth					5.875	4		
(12)	kim jae hoon					5.876	5		
(13)	licai yang							26.818	**1**
(14)	zhibing zhang							26.821	2
(15)	li gang							26.829	3
(16)	gong yumeng							26.859	4
(17)	wei wei							27.016	5

Table 3: Affiliation similarity - top distance measure scores and ranks. Affiliation: Suzhou University of science and technology.

	Affiliation Name	Dot Product		Cosine Similarity		Euclidean		Manhattan	
		Score (1)	Rank (2)	Score (3)	Rank (4)	Score (5)	Rank (6)	Score (7)	Rank (8)
(1)	daido university	9.386	**1**	0.370	**1**				
(2)	university medical center freiburg	6.627	2	0.242	2				
(3)	university of kansas hospital	6.395	3	0.228	5				
(4)	benedictine university	5.568	4	0.234	4				
(5)	henkel (germany)	4.856	5						
(6)	university of tehran			0.240	3				
(7)	national autonomous university of mexico					6.866	**1**		
(8)	university of canterbury					6.917	2		
(9)	royal military academy					6.962	3		
(10)	czech education and scientific net work					6.987	4		
(11)	dalian maritime university					7.002	5		
(12)	university of indianapolis							31.932	**1**
(13)	bahçeşehir university							32.012	2
(14)	htw berlin - university of applied sciences							32.175	3
(15)	new zealand college of chiropractic							32.513	4
(16)	institut des nanotechnologies de lyon							32.608	5

The top-ranked similar affiliations in rows (1)-(4), are specialized in Health, Medical and Bioengineering Sciences, which correspond to the domains embedded in the user query. The closest points in rows (7) and (12) in the embedding spaces measured by Euclidean and Manhattan distances are more distant with respect to the top-1 affiliations generated by dot product and cosine similarity. Although Tab. 3 shows patterns similar to those in Tab. 2, measures such as the dot product and cosine similarity generate better recommendations than the Euclidean and Manhattan distances.

Based on our evaluations on various entity types, we return the top-ranked entities in the majority of distance measures to the user. It is very interesting to observe how distance measures across various entity types can vary using the identical user query template. As we discuss in Section 5, it is important to design distinctive query strategies for authors – with box embeddings, and then textual similarities can help to find similar author profiles.

4.2 External Evaluation: Academic Databases and LLMs

There have been many existing academic databases where researchers and practitioners frequently consult, and recently LLMs with functionality to support research-related prompts have been developed. In this section, we conduct external evaluations of the output of the chatbot with respect to the output generated by these DBs and specialized LLMs.

Tab. 5 presents a comprehensive comparison of research paper recommendations across a spectrum of LLMs and DBs, including SciSpace, ChatGPT4, Consensus, Perplexity, Research GPT, Google Scholar, and SemOpenAlex. This unique compilation highlights the diversity of perspectives within the AI and optimization community.

One notable observation from this analysis is the repeated recommendation of the paper "A Non-revisiting Quantum-behaved Particle Swarm Optimization Based Multilevel Thresholding for Image Segmentation" by both SciSpace and Perplexity, indicating its potential as a key contribution to the field of Image Segmentation Using Particle Swarm Optimization techniques. Similarly, "Non-revisiting Genetic Algorithm with Adaptive Mutation Using Constant Memory" has been recognized by SciSpace and Perplexity, which could point towards growing interest in efficient genetic algorithms that minimize memory use while optimizing performance. Although the recommendations of various products vary largely, it is interesting to see the commonalities of the recommended publications, which is mainly based on n-grams or textual similarity of title and/or abstract. This again indicates a promising future research direction in building a chatbot that can leverage both textual similarity as well as KG based similarity (see Section 5). It is interesting to back-engineer the recommendation protocols of the LLMs and DBs based on their output. However, at this stage, it is unclear whether the recommendation is based on citation, text similarity, or KG-based information.

In the inquiry about recommending top authors similar to “Stefan Kraxberger", it was observed that none of the LLMs provided any author recommendations based on this author. The procedure either provides authors with similar names (“Stefan Berger", “Stefan Hergarten" by SciSpace, “Brennan M. Kraxberger" by Consensus) or it explains about the steps to build a co-author recommendation system. This outcome might reflect the specific challenges associated with accurately mapping individual authors' styles, contributions, or thematic similarities within the vast expanse of literary or academic landscapes. Such a task requires not only a deep understanding of the author's work but also an extensive knowledge base of similar works by other authors, which might not always be readily accessible or easily quantifiable by LLMs. On the other hand, the academic databases like Google Scholar cannot give extensive recommendations of similar authors, SemOpenAlex only lists co-authors and most cited/citing authors.

When asked to recommend affiliations similar to “Suzhou University of Science and Technology" (see Tab. 6), all LLMs except SciSpace offered various recommendations. This contrast underscores the difference in the data handling capabilities of LLMs when dealing with structured information like affiliations, which can be categorized based on factors like location, academic focus, and size, as opposed to the nuanced and subjective task of comparing authors. The absence of recommendations from SciSpace in the institutional query further highlights the variability in how different models or platforms are tuned to respond to specific types of queries, possibly influenced by their underlying datasets, algorithms, or research areas.

5 Discussions and Future Directions

Point vs. Box Embeddings. Point embeddings involve embedding user queries into the same vector space as the KG, creating a new point within this space. This method proves especially effective for single-hop queries due to its simplicity and the interpretability of the results. It operates by measuring distances to neighboring points using distance metrics, like e.g. Euclidean, cosine, or Manhattan, to identify the most suitable match for a query.

Box embeddings offer a more sophisticated approach compared to point embeddings. Rather than representing a user query as a single point, it is transformed into several “boxes" in the vector space. This transformation is based on a query plan derived from the user's input query in first-order logic. The Query2Box method (Ren et al., 2020), a prominent technique for box embeddings, is particularly effective for multi-hop queries. The intersections of these boxes naturally represent filtered results, which can be presented to users without the need for additional distance metrics.

More Query Templates. We plan to leverage techniques of box embeddings that can understand more complex queries with multi-hop projections and intersection, e.g., “Return the top cited papers by “Author Name"" or “Return the top publishing authors belonging to the “Institute Name"".

Unseen Entities in KG. We plan to conduct another internal evaluation using a hold-out set of unseen entities in our KG. The idea is that we map the unseen entities to the closest top-k candidates in the embedding spaces.

KG Based vs. Textual Based Recommendations. In Section 4.1, we analyze the importance of KG-based recommendations (node, edge, neighborhood) in the current implementation of the chatbot. However, we do see the need to incorporate textual based recommendations either through the textual embedding space through Pre-trained LM or n-grams (Muennighoff et al., 2022) or through the joint learning of graph structure and textual nodes as in Yan et al. (2023).

6 Conclusions

We have presented an academic chatbot which uses LLMs and reasons with KG embeddings to recommend authors, affiliations and articles similar to user queries. We have also evaluated internally and externally the recommendations by the chatbot with respect to the ones from the popular academic

LLMs and DBs. By doing so, we increase the interpretability of the results returned to the user. The current implementation of the chatbot can be extended to multi-hop queries in future work.

Limitations

As we have reflected in Section 5, there are various ways to modify and improve the proposed chatbot. One important direction is to enable recommendations for unseen entities in the existing KG. This will improve the robustness of our chatbot by searching for similar entities in the embedding space.

Another limitation of the current work is the backend KG. Due to the fact that the SPARQL public endpoint of SemOpenAlex is currently out of service, we build the chatbot development on the AIDA KG. In the long run, we intend to support the chatbot with SemOpenAlex as the backend KG. Internally, we have already started building an in-house KG of SemOpenAlex. However, we are also in contact with the SemOpenAlex team to have a public KG running for the common good.

Last but not least, we also plan to support multilingual user query in the future, on one hand we support cultural diversity in user query, one the other hand, we plan to incorporate multilingual academic KGs that better cover the global landscape of academic research.

Ethics Statement

All our KG and LLM products that have been or will be used are open source products. We will not disclose user queries as part of our question answering system. To increase the interpretability of our system, we have also returned the query transformation plan to keep users informed. In the future, we plan to use reinforcement learning to collect user comments to improve the learning process and enhance the user experience. We plan to protect user information by anonymizing comments and ensuring fair and unbiased treatment for all users. We plan to release our chatbot for user studies via a Telegram bot and will handle user privacy discreetly once we are that far with the implementation.

Broader Impact Statement

We plan to incorporate one of the largest academic graph, OpenAlex, and the patent graph, PATSTAT. This will support the decision making in researchers and industry practitioners in querying complex graph-structured data. In addition to graph-based information retrieval, the academic chatbot can also be used in pertinent domains such as teaching assistance and research evaluation.

References

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33:9649–9661.

Debayan Banerjee, Pranav Ajit Nair, Jivat Neet Kaur, Ricardo Usbeck, and Chris Biemann. 2022. Modern baselines for sparql semantic parsing. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2260–2265.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. 2022. Knowledge graph embedding: A survey from the perspective of representation spaces. *arXiv preprint arXiv:2211.03536*.

Chi Kin Chow and Shiu Yin Yuen. 2008. A non-revisiting particle swarm optimization. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 1879–1885. IEEE.

Consensus. 2024. Consensus. https://consensus.app/search/. Accessed on February 3, 2024.

Michael Färber, David Lamprecht, Johan Krause, Linn Aung, and Peter Haase. 2023. Semopenalex: The scientific landscape in 26 billion rdf triples. In *International Semantic Web Conference*, pages 94–112. Springer.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023. Text-to-sql empowered by large language models: A benchmark evaluation.

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Haridimos Kondylakis, Dimitrios Tsirigotakis, Giorgos Fragkiadakis, Emmanouela Panteri, Alexandros Papadakis, Alexandros Fragkakis, Eleytherios Tzagkarakis, Ioannis Rallis, Zacharias Saridakis, Apostolos Trampas, et al. 2020. R2d2: A dbpedia chatbot using triple-pattern like queries. *Algorithms*, 13(9):217.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Antonello Meloni, Simone Angioni, Angelo Salatino, Francesco Osborne, Aliaksandr Birukou, Diego Reforgiato Recupero, and Enrico Motta. 2023a. Aidabot 2.0: Enhancing conversational agents with knowledge graphs for analysing the research landscape. In *International Semantic Web Conference*, pages 400–418. Springer.

Antonello Meloni, Simone Angioni, Angelo Salatino, Francesco Osborne, Diego Reforgiato Recupero, and Enrico Motta. 2023b. Integrating conversational agents and knowledge graphs within the scholarly domain. *IEEE Access*, 11:22468–22489.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Peter Ochieng. 2020. Parot: Translating natural language to sparql. *Expert Systems with Applications: X*, 5:100024.

OpenAI. 2022. GPT-3.5. https://platform.openai.com/docs/models/gpt-3-5.

OpenAI. 2023. Chatgpt-4. Online Resource.

Mukul Patnaik. 2023. Researchgpt. https://github.com/mukulpatnaik/researchgpt.

Perplexity.ai. 2024. Perplexity.ai.

Jason Priem, Heather Piwowar, and Richard Orr. 2022. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833*.

Julio C Rangel, Tarcisio Mendes de Farias, Ana Claudia Sima, and Norio Kobayashi. 2024. Sparql generation: an analysis on fine-tuning openllama for question answering over a life science knowledge graph. *arXiv preprint arXiv:2402.04627*.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. *arXiv preprint arXiv:2002.05969*.

Richard Roberson, Gowtham Kaki, and Ashutosh Trivedi. 2024. Analyzing the effectiveness of large language models on text-to-sql synthesis.

SciSpace.com. 2024. Scispace. https://www.scispace.com. Accessed on February 3, 2024.

SemOpenAlex. 2024. Semopenalex.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Computing Research Repository*, arXiv:2307.09288.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xiaojun Xu, Chang Liu, and Dawn Song. 2017. Sqlnet: Generating structured queries from natural language without reinforcement learning.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning.

A AIDA KG Relation Types

The following relation types are contained in the AIDA KG. We include the nodes and relations used in our TransE embeddings. You can refer to the official AIDA schema under this URL (https://aida.kmi.open.ac.uk/#aidaschema).

1. Member of: Describes the membership relation of an entity in another entity or group.
2. Has Syntactic Topic: Indicates the syntactic topic associated with an entity.
3. Has Industrial Sector: Relates an entity to a specific industrial sector.
4. Type: Specifies the type or category of an entity.
5. Title: Represents the title of a document or a work.
6. DOI: Stands for Digital Object Identifier, uniquely identifying digital documents.
7. Name: Indicates the name of an entity.
8. Has Percentage Of Industry: Specifies the percentage share an entity has in an industry.
9. Has Affiliation: Indicates the affiliation of a person or an entity to an organization.
10. Has Grid Type: Relates to the type of grid system in an industrial context.
11. Has Affiliation Type: Specifies the type of affiliation of an entity.
12. Has Semantic Topic: Relates an entity to its semantic topic or theme.
13. Publication Date: Indicates the date of publication of a document or work.
14. Same As: Indicates that two entities are identical or equivalent.
15. Label: Provides a human-readable label or name for an entity.
16. Creator: Indicates the creator or author of a work.
17. Has DBpedia Category: Links an entity to its corresponding category in DBpedia.
18. Has Percentage Of Academia: Specifies the percentage share an entity has in the academic sector.
19. Has Topic: Relates an entity to a general topic or subject area.

The representations of these relation types for the AIDA KG are visualized in Fig. 4.

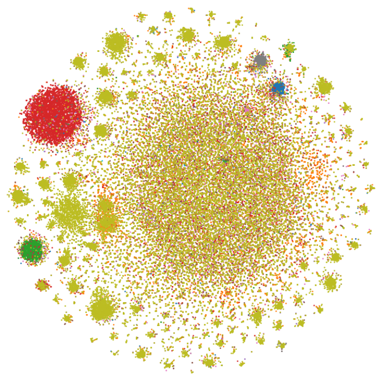

Figure 4: TransE embedding space trained for a subgraph of the AIDA KG visualized using t-SNE dimensionality reduction, displaying the different entity types as colors, e.g., `AUTHOR`, `AFFILIATION`, `PAPER`.

B Generating SPARQL Query using LLM

We employ few-shot learning to evaluate the quality of SPARQL queries automatically generated by LLMs. Taking Fig. 5 as an example, after providing four examples,[8] we ask LLMs to generate a SPARQL query for a new task. Tab. 4 displays the reference answer for this task, alongside the outputs generated by `Llama-2-7b-chat` (Touvron et al., 2023) and `ChatGPT-3.5` (OpenAI, 2022). It is evident that the result from `ChatGPT-3.5` closely approximates the reference answer, whereas the query produced by `Llama-2-7b-chat` fails to meet the requirements.

[8]Examples of queries are retrieved from "Help" section of https://aida.kmi.open.ac.uk/sparql/.

```
The following query returns all papers written by authors from the industrial sector computing
and it associated with the topic robotics:
PREFIX aida-ont:<http://aida.kmi.open.ac.uk/ontology#>
PREFIX aida:<http://aida.kmi.open.ac.uk/resource/>
PREFIX aidaDB: <http://aida.kmi.open.ac.uk/resource/DBpedia/>
PREFIX cso: <http://cso.kmi.open.ac.uk/topics/>
SELECT ?paperId
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?paperId aida-ont:hasIndustrialSector aida:computing_and_it .
    ?paperId aida-ont:hasTopic cso:robotics .
}
LIMIT 20

The following query counts how many papers have been written by authors from an industrial
affiliation.
PREFIX aida:<http://aida.kmi.open.ac.uk/ontology#>
SELECT (COUNT(?sub) as ?count)
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?sub aida:hasAffiliationType "industry"
}

The next query counts how many authors are affiliated with The Open University.
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX schema:<http://schema.org/>
SELECT (COUNT(DISTINCT(?sub)) as ?count)
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?sub schema:memberOf ?aff .
    ?aff foaf:name "the_open_university"
}

The following query returns the industrial sectors of all the papers having Semantic Web as a topic.
PREFIX aida:<http://aida.kmi.open.ac.uk/ontology#>
PREFIX cso: <http://cso.kmi.open.ac.uk/topics/>
SELECT DISTINCT ?ind
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?sub aida:hasTopic cso:semantic_web .
    ?sub aida:hasIndustrialSector ?ind
}

Please output the SPARQL query that returns the topic distribution of a given affiliation
(in this case The Open University)
```

Figure 5: Few-shot learning prompt template for an LLM to recognize custom entities (PAPER). {text} should be filled with a user query.

Table 4: LLM-generated SPARQL queries vs. reference.

Gold Standard

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX aida:<http://aida.kmi.open.ac.uk/ontology#>
PREFIX schema: <http://schema.org/>
SELECT ?topic (count(distinct(?paper)) as ?count)
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?paper schema:creator ?author .
    ?author schema:memberOf ?aff .
    ?aff foaf:name "the_open_university" .
    ?paper aida:hasTopic ?topic .
} GROUP BY ?topic
ORDER BY DESC(?count)
```

ChatGPT-3.5

```
PREFIX aida: <http://aida.kmi.open.ac.uk/ontology#>
PREFIX schema: <http://schema.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX cso: <http://cso.kmi.open.ac.uk/topics/>

SELECT ?topic (COUNT(?paperId) as ?count)
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?paperId schema:memberOf ?aff .
    ?aff foaf:name "the_open_university" .
    ?paperId aida:hasTopic ?topic .
}
GROUP BY ?topic
ORDER BY DESC(?count)
```

Llama-2-7b-chat

```
PREFIX aida:<http://aida.kmi.open.ac.uk/ontology#>
SELECT ?paperId
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
    ?sub aida:hasAffiliationType "the_open_university"
}
LIMIT 20
```

Table 5: Comparative ranking of recommended research papers by academic LLMs and databases. Paper: a non revisiting particle swarm optimization.

	Paper Title	SciSpace	ChatGPT4	Consensus	Perplexity	Research GPT	Google Scholar	SemOpenAlex
(1)	A non-revisiting particle swarm optimization	1						
(2)	A non-revisiting quantum-behaved particle swarm optimization based multilevel thresholding for image segmentation	2			4			
(3)	Continuous non-revisiting genetic algorithm	3						
(4)	Non-revisiting genetic algorithm with adaptive mutation using constant memory	4			3			
(5)	Restarting Particle Swarm Optimisation for deceptive problems	5						
(6)	Quantum-behaved particle swarm optimization based on solitons		1					
(7)	A Quantum-Behaved Particle Swarm Optimization Algorithm on Riemannian Manifolds		2					
(8)	Non-revisiting Stochastic Search Revisited: Results, Perspectives, and Future Directions		3					
(9)	Advances in Particle Swarm Optimization for Antenna Designs			1				
(10)	A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems			2				
(11)	The fully informed particle swarm: simpler, maybe better			3				
(12)	Particle swarm optimization versus genetic algorithms for phased array synthesis			4				
(13)	A Survey of Particle Swarm Optimization Applications in Electric Power Systems			5				
(14)	An Improved Particle Swarm Optimization with Mutation Based on Similarity				1			
(15)	A non-revisiting genetic algorithm based on a novel binary space				2			
(16)	A Genetic Algorithm That Adaptively Mutates and Never Revisits				5			
(17)	Particle swarm optimization					1		
(18)	Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients					2		
(19)	A modified particle swarm optimizer					3		
(20)	The swarm and the ant: A particle swarm optimizer using ants as neighborhood guides					4		
(21)	On the convergence of the global best particle swarm optimizer					5		
(22)	A hybrid particle swarm algorithm with Cauchy mutation						1	
(23)	Opposition-based particle swarm algorithm with Cauchy mutation						2	
(24)	A novel intelligent particle optimizer for global optimization of multimodal functions						3	
(25)	A uniform searching particle swarm optimization algorithm						4	
(26)	Particle swarm optimization algorithms with novel learning strategies						5	
(27)	Guaranteed Convergence Particle Swarm Optimization using Personal Best							1
(28)	Particle Swarm Optimization with comprehensive learning & self-adaptive							2
(29)	Novel particle swarm optimization with heuristic mutation							3
(30)	Enhanced Particle Swarm Optimization and Its Application							4
(31)	An Improved Particle Swarm Optimization Based on Biological Chemotaxis							5

Table 6: Comparative ranking of recommended research papers by academic LLMs.
Affiliation: Suzhou University of Science and Technology.

	University Name	ChatGPT4	ResearchGPT	Consensus	Perplexity	SciSpace
(1)	Zhejiang University of Science and Technology (ZJUT)	1				
(2)	Nanjing University of Science and Technology (NUST)	2	1			
(3)	Harbin Institute of Technology (HIT)	3				
(4)	Huazhong University of Science and Technology (HUST)	4				
(5)	Shanghai University of Science and Technology (USST)	5				
(6)	Jiangsu University		2			
(7)	Hefei University of Technology		3			
(8)	China University of Mining and Technology		4			
(9)	Southeast University		5			
(10)	South University of Science and Technology of China (SUSTC)			1		
(11)	Shenyang University			2		
(12)	ShanghaiTech University and Westlake Institute for Advanced Study			3		
(13)	Singapore University of Technology and Design (SUTD)			4		
(14)	Xi'an Jiaotong–Liverpool University (XJTLU)			5	2	
(15)	Soochow University				1	
(16)	The Chinese University of Hong Kong, Shenzhen				3	
(17)	Suzhou Centennial College				4	
(18)	IE University				5	