LLM-based Reranking and Validation of Knowledge Graph Completion

Weihang Zhang[1,*], Ovidiu Serban[1]

[1]Data Science Institute, Imperial College London Exhibition Rd, South Kensington, London, SW7 2BX, United Kingdom

Abstract

In the past decade, Knowledge Graph Completion (KGC), a task aimed at discovering missing knowledge within a knowledge graph by predicting missing links between entities, has garnered significant attention from researchers. Despite significant algorithmic advances, even state-of-the-art KGC models produce predictions with unavoidable inaccuracies. The crucial yet underexplored task of validating these predictions before integration remains predominantly manual, creating a substantial bottleneck in knowledge graph curation pipelines. Unvalidated KGC predictions can propagate errors to downstream knowledge-intensive applications, such as fact-checking, potentially compromising reliability. To address this challenge, we propose a novel workflow leveraging Large Language Models (LLMs) that mimics human validation processes by systematically evaluating KGC predictions made by existing systems through dual verification: analyzing and reranking plausible KGC predictions via internal graph evidence and external knowledge sources. The proposed method leverages token probabilities from LLMs to quantify model confidences and, therefore, provides reranking results of KGC predictions made by existing methods. Our work presents the first large-scale empirical evaluation of using LLMs to post-process KGC predictions on standard benchmarks. Experimental results demonstrate that our approach can significantly reduce human validation efforts while maintaining high factual accuracy standards in knowledge graph curation.

Keywords

Knowledge Graph Completion, Large Language Model, Knowledge Graph Curation

1. Introduction

Knowledge graphs (KGs) have become essential for organizing structured information across numerous applications, from search engines and recommendation systems to complex reasoning tasks like factual verification. As KGs scale, one of the most critical challenges to the KG curation process is to improve the inherent incompleteness. Knowledge Graph Completion (KGC) has emerged as a promising approach to address this challenge by systematically predicting missing facts. However, despite significant theoretical advances [1, 2, 3, 4], in KGC research, a substantial gap remains between these algorithmic improvements and their practical implementation in real-world knowledge graphs: while substantial research efforts have focused on developing increasingly sophisticated algorithms to improve KGC prediction accuracy, comparatively little attention has been given to what happens after these predictions are generated and how can they be integrated into KG curation process. This oversight is problematic because even state-of-the-art KGC models produce imperfect predictions that, if directly integrated into knowledge graphs, can propagate errors to downstream applications. For instance, when knowledge graphs serve as evidence sources for fact-checking systems, erroneous KGC predictions can lead to incorrect verification outcomes, undermining the reliability of the entire pipeline. The validation and selection of KGC predictions is a crucial yet underexplored step in the knowledge graph curation lifecycle. This process relies heavily on manual human validation — a resource-intensive approach that can become increasingly unsustainable as KGs scale in size and complexity [5]. The fundamental challenge stems from the tension between two competing evaluation paradigms: the closed-world assumption traditionally used in KGC and linked prediction model evaluation, which assumes all missing facts to be incorrect, and the open-world assumption that better reflects reality by acknowledging

*Corresponding author.

w.zhang21@imperial.ac.uk (W. Zhang); o.serban@imperial.ac.uk (O. Serban)

0000-0002-6244-5748 (W. Zhang); 0000-0001-5359-3661 (O. Serban)

that knowledge graphs are inherently incomplete and that missing facts should not automatically be treated as incorrect. Although the open-world assumption aligns more closely with real-world scenarios, the closed-world assumption is far more commonly used in knowledge graph completion evaluation frameworks, primarily because it simplifies the automation of the evaluation process without requiring human effort. The extensive human efforts required to correctly evaluate the KGC predictions under open-world assumptions against real-world facts are discussed in several recent works [6, 5].

As knowledge graphs continue to expand, there is an urgent need for approaches that can automate or semi-automate the validation process while maintaining high standards of factual accuracy. The emergence of Large Language Models (LLMs), with their remarkable capabilities for reasoning, inference, and natural language understanding, presents a promising direction for addressing this challenge. LLMs have demonstrated the ability to reason about factual information and access parametric knowledge and external resources through techniques like Retrieval-Augmented Generation (RAG). This paper proposes a novel framework that leverages LLMs as automated agents for validating and ranking KGC predictions. Our approach mimics the workflow of human annotators by evaluating predicted triplets based on internal graph evidence (leveraging existing knowledge within the KG) and external references (utilizing broader knowledge sources via search engines). This dual-validation mechanism helps ensure that only high-confidence, factually accurate predictions are integrated into the knowledge graph, thus maintaining its integrity while reducing the human effort required for curation. Additionally, the proposed approach leverages next-token probability distributions from LLMs to quantify validation confidence, enabling systematic comparison across multiple candidates for the same KGC task.

Currently, there exist few studies that explore reducing or replacing human effort in the KGC task, with the notable exception of KGValidator [5], which utilises a RAG framework with LLMs to conduct validations of KGC predictions under the open-world setting. KGValidator approaches the validation of KGC predictions as a classification task, verifying each prediction against external sources such as web search results or reference knowledge graphs. Although this paper employs a similar setup to reduce human effort in post-processing KGC predictions, it differs from KGValidator in several key aspects. First, alongside external evidence, the proposed method incorporates relevant graphical evidence from within the KG, mimicking the human tendency to prioritize internal sources before using external knowledge sources to validate a predicted triplet. Second, rather than framing validation as a classification problem, the proposed method treats it as a reranking task. Here, the top-k candidates from KGC predictions are retrieved and reranked by LLMs. This reranking approach allows for broader-scale evaluation and directly compares model improvements with traditional KGC evaluation metrics. Lastly, LLMs in the proposed method are prompted to validate each prediction, with the next-token probabilities used to quantify the confidence scores for each judgment. These scores are then used to reorder the top-k candidates, providing a more intuitive view of intermediate results when used to aid human annotators.

Overall, this paper makes the following contributions to the field:

- We identify and address a critical gap in the real-world KGC pipeline by focusing on the underexplored post-prediction reranking and validation process.
- We propose a novel LLM-based framework for automating the validation and reranking of KGC predictions using internal and external knowledge sources, mimicking human workflow.
- Our work presents the first large-scale empirical evaluation of using LLMs to post-process KGC predictions on standard benchmarks, demonstrating that our approach can significantly reduce the required human validation effort.

The remainder of this paper is organized as follows: section 2 reviews related work in knowledge graph completion and recent advances in language models for the KGC task. Section 3 details our proposed methodology for LLM-augmented reranking of KGC predictions. Section 4 presents our experimental setup and results. Finally, section 5 discusses implications, limitations, and directions for future work.

2. Related Work

2.1. Knowledge Graph Completion

Over the past decades, the terms knowledge graph completion, link prediction, and sometimes knowledge graph reasoning have often been used interchangeably by researchers when referring to the task that aims to solve the sparsity and incompleteness problem of knowledge graphs. The past decades have also observed rapid development and iterations of KGC methods. The predominant approaches in KGC research rely on representation learning techniques, where entities and relations are encoded into low-dimensional embedding spaces. Translation-based models such as TransE [2] represent relations as translations in the embedding space, while semantic matching models like RESCAL [1] employ tensor factorization to capture multi-relational data. These two works have spawned numerous subsequent studies [7, 8, 9, 10, 11, 12, 13] addressing their inherent limitations. The emergence of graph neural networks catalyzed significant advancements in KGC approaches. Researchers have leveraged Graph Convolutional Networks (GCNs) [14] and their variants as sophisticated encoders to capture complex knowledge graph structures within representation spaces. Notable contributions such as R-GCN [3], CompGCN [4], and KGAT [15] have demonstrated superior performance on knowledge graph completion tasks by effectively modelling multi-hop neighbourhood information and relational patterns. More recently, researchers [16, 17] have also explored KGC, which utilizes the collective knowledge of multiple KGs to aid the curation process.

2.2. Language Model aided Knowledge Graph completion

Rapid developments in pre-trained and large language models have given researchers great tools for capturing semantic meaning and patterns in natural language. As a result, these advancements motivated the development of various methods for incorporating the language models to enhance the knowledge graph completion task. One of the earliest attempts was KG-Bert [18], in which triplets in KGs are treated as textual sequences. Several works further extended the concept of KG-Bert. For example, Wang et al. [19] proposed a structure-augmented text representation(StAR) model, which uses a Siamese-style textual encoder to learn two contextualized representations for a single triplet. Lovelace and Rosé [20] introduces a supervised embedding extraction method to best leverage entity embeddings for the KGC task. With the rapid development of LLMs, Zhu et al. [21] conducts comprehensive evaluations of LLMs for KG construction and reasoning, showing that LLMs perform close to state-of-the-art models across both tasks. While the research community has made remarkable progress in developing increasingly sophisticated KGC methods, the predominant focus has remained on improving prediction accuracy through algorithmic innovations. However, despite these advances, KGC predictions exhibit significant imperfections that limit their direct applicability in real-world knowledge graph curation. To our knowledge, KGValidator [5] is one of the notable exceptions that attempts to use LLMs to automate the validation process of the KGC predictions. A retrieval-augmented generation workflow against the open-source knowledge base and web search is proposed to reduce human efforts in the post-processing step of KGC predictions.

3. Method

As illustrated in Figure 1, the proposed method first generates predicted triplets and integrates internal knowledge graph evidence and external references to rerank the predicted triplets. To mimic the real-life situation, this study generates the predicted triplets using traditional embedding-based KGC methods. For each KGC task (e.g. head prediction or tail prediction), the top-k predicted entities are included for the proposed subsequent LLM reranking workflow. Including traditional KGC methods effectively reduces the search space for the LLM reranking workflow. However, this also imposes reliance on the quality of the conventional KGC method, as the reranking process can only be beneficial if the correct prediction exists in the initial top-k predicted entities for a KGC task.

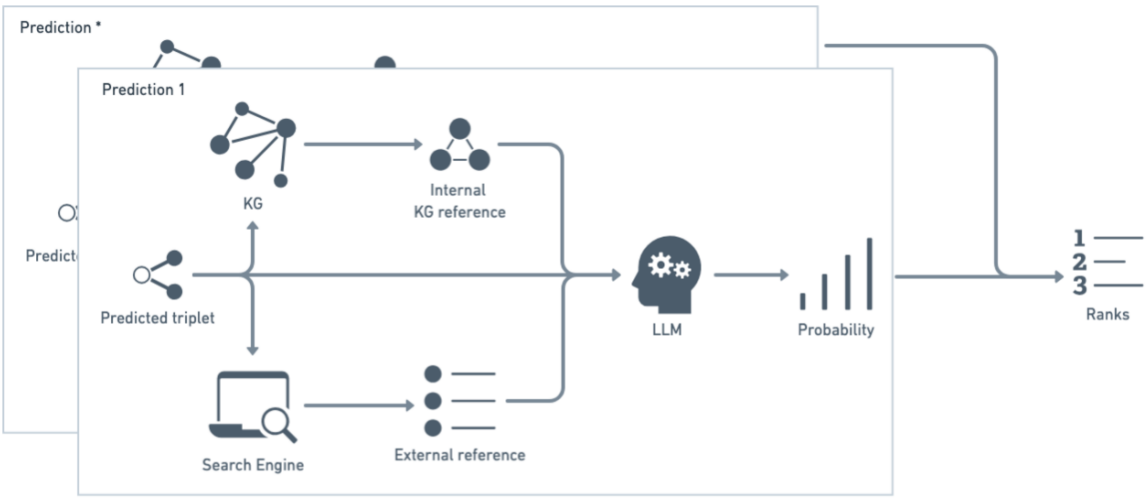

Figure 1: Workflow of the proposed method using LLM as a reranker for KGC predictions: for each KGC task (h, r, ?) or (?, r, t), each of the top-k predicted triplets is evaluated and reranked through the pipeline.

Overall, to rerank the potential candidates for a KGC task, the retrieval process consists of three main steps: 1) retrieval of internal graphical evidence, 2) retrieval of external references via the DuckDuckGo search engine, and 3) reranking KGC predictions based on LLM token probabilities.

3.1. Retrieval of internal graphical evidence

Drawing inspiration from how humans validate predicted triplets, the first step involves retrieving internal evidence directly from the knowledge graph. For each predicted triple, three types of retrieval are performed:

- First, for each triplet of form (h, r, t), graph walks are initiated from the head entity h to the tail entity t, or vice versa, with a maximum walk length of 3. These walks explore the intermediate nodes and edges that connect h and t within the knowledge graph. For example, for a triplet (Paris, locatedIn, France), a valid graph traversal path could be (Paris, partOf, Île-de-France), (Île-de-France, locatedIn, France). The resulting paths represent potential reasoning chains that may explain or support the predicted relationship. These paths capture indirect associations and contextual information, which can help validate the plausibility of the predicted triplet by highlighting logical or meaningful connections between the entities.
- Second, all triplets in the knowledge graph are encoded and stored in an embedding index. For each predicted triplet, dense retrieval is performed to identify and retrieve semantically similar triplets. This dense retrieval process is designed to uncover similar patterns not directly connected to the triplets, providing additional context or supporting evidence for the prediction.
- Lastly, for each predicted triplet, all triplets in the knowledge graph that share the same relation r are also retrieved. These examples offer additional context for the predicate, providing insights into its typical usage and helping to prevent misinterpretation, especially when the relation has multiple or ambiguous meanings. For example, in FB15K-237 [22], a widely-adopted KGC dataset, the relation “place” is sometimes used in a self-referential triplet, such as (’Glen Cove’, ’place’, ’Glen Cove’). The triplet indicates that Glen Cove is categorized as a place. Such self-referential examples may initially seem confusing for human annotators, but they are technically accurate. Depending on the definitions of relations in different knowledge graphs, similar relations can occasionally lead to ambiguity. The proposed method retrieves example triples corresponding to the target relation to address this issue, providing more contextual examples for the subsequent LLM prompting.

Table 1
Example of a prompt for verifying predicted triplet.

Example prompt
You are given a target knowledge graph triplet in the form (head, relation, tail). Your task is to judge whether this triplet is factually correct based on the given information and common sense. To help you with the judgment, we provide some additional information that may be helpful to your judgment: 1) existing triplets in the knowledge graph have the same relation, so you can see if the target triplet follows the same pattern. 2) existing triplets in the knowledge graph that may be relevant to the target triplet. 3) additional context that may be relevant from a search engine to validate the target triplet. Remember your task is to judge the factuality of the target triplet based on the provided information and your internal knowledge. Use common sense where applicable. Always start your answer with correct, incorrect or NEI(not enough info) to indicate your judgment of the factuality of the target triplet and give a short one-sentence reason to justify your judgement. == Provided information == **KG triplets with same relation:** [('John Wood', 'nationality', 'England'), ('Mark Twain', 'nationality', 'United States of America'), ('Lars von Trier', 'nationality', 'Denmark-GB'), ('Sridevi', 'nationality', 'India'), ('Brad Grey', 'nationality', 'United States of America')] **Existing triplets:** [[('Stevie Wonder', 'currency', 'United States Dollar'), ('United Kingdom', 'currency', 'United States Dollar')], [('Stevie Wonder', 'currency', 'United States Dollar'), ('United Kingdom', 'adjustmentcurrency', 'United States Dollar')], [('Elton John', 'awardnominee', 'Stevie Wonder'), ('Elton John', 'nationality', 'United Kingdom')], ('Stevie Nicks', 'nationality', 'United States of America'), ('Stevie Ray Vaughan', 'nationality', 'United States of America'), ('Stevie Wonder', 'profession', 'Singer-songwriter-GB'), ('Stevie Wonder', 'profession', 'Actor-GB'), ('Stevie Wonder', 'origin', 'Detroit')] **Additional context:** Wonder told the BBC that gaining Ghanaian nationality on his birthday was an "amazing thing". The superstar was born and bred in the US state of Michigan but has long had an affinity for Ghana - a ... American music icon Stevie Wonder has been granted Ghanaian citizenship. In a visit to Accra, Ghana's capital, in which he was joined by his family, the artist heard a speech from the country's ... Ghana was the first Black African country south of the Sahara to achieve independence from colonial rule in 1957, britannica.com said. Blind from infancy, Wonder was born in Saginaw as Steveland ... **Target triplet:** ('Stevie Wonder', 'nationality', 'United Kingdom')

3.2. Retrieval of external reference

In addition to internal references derived from patterns and reasoning paths within the knowledge graph, external references play a crucial role in validating and ranking predicted triples, particularly when internal knowledge is insufficient. Previous studies [23, 6] have shown that predicted triples, while deemed incorrect under the closed-world assumption commonly used in automated KGC evaluations, may be correct under the open-world assumption. Incorporating external references from outside the knowledge graph can effectively address this discrepancy, providing additional validation during the reranking process and improving the accuracy of predicted triples. In the process, external evidence is retrieved using the DuckDuckGo search engine. The queries are constructed based on the simple concatenations of the predicted triplet's head, relation, and tail (e.g. 'Paris partOf Île-de-France'). DuckDuckGo is selected for its privacy-preserving search capabilities, which provide broad and unbiased retrieval from diverse external sources. The top results, including documents, web pages, and knowledge snippets, are collected and processed to supplement the validation process.

3.3. Reranking with LLMs

After retrieving internal graphical evidence and external references, the next step is constructing a structured prompt for the LLM. This prompt includes the original KGC prediction alongside the

supporting or contradicting evidence from internal and external sources. The prompt aims to refine the ranking of predictions for each KGC query, prioritising the most accurate results. While LLMs have already shown impressive performance with various reasoning tasks, even under the zero-shot settings, there has been limited success for the important text ranking problem using off-the-shelf LLMs [24]. Several existing studies [24, 25, 26] have focused on improving the document ranking problems by formulating the task with different prompting strategies. The two main formulations of the ranking task are list-wise ranking and point-wise ranking. In list-wise ranking, the ranking candidates are evaluated together, requiring the inclusion of all candidates' contextual information within the prompt for the LLM. While this method has shown effectiveness in previous studies [24, 25], it often leads to conflicting or irrelevant outputs, particularly when using moderate-sized LLMs. For ranking KGC predictions, list-wise ranking poses a significant challenge, as it necessitates an exceptionally long context to account for all candidates, potentially exceeding the input length limit of some models and causing confusion or hallucination for the LLMs. Pair-wise ranking with LLMs [26] addresses the long-context issue inherent in list-wise ranking by comparing candidates in pairs. However, it can be computationally expensive, as the model may need to evaluate numerous permutations of candidate pairs. Even with optimized sorting and selection strategies, pair-wise ranking can still incur significant computational costs, making it less efficient for tasks with large candidate sets. For ranking KGC predictions, pair-wise ranking becomes significantly less efficient due to the large number of predictions typically generated by algorithms, requiring an impractical number of pairwise comparisons.

This paper proposes a method for ranking KGC predictions by prompting the LLM with each predicted triplet while specifying the LLM agent to start the response with the judgement of "Correct", "Incorrect" or "NEI" (Not Enough Information). The token probability of the first generated token is then used to quantify and rank the candidates for each KGC query. Table 1 provides an illustrative example of the designed prompt. In the designed prompt, instructions are explicitly added for the LLM to begin by providing a natural language judgment on the correctness of the KGC predicted triplet, given the retrieved evidence. Instead of simply providing a ranked output, the LLM first assesses the prediction, either affirming it as correct or flagging it as incorrect, based on the evidence provided. This step allows the model to engage in reasoning and explain its decision, similar to how a human evaluator might approach the task. The next-token probability for the first token $P(Correct)$ is recorded as a quantification of the LLM's internal assessment and is used for ranking.

For KGC tasks like head prediction $(h, r, ?)$ and tail prediction $(?, r, t)$, traditional deep-learning-based KGC methods are used to generate a ranking of candidates. The proposed LLM-based reranking method takes the top 10 candidates for each query, retrieves relevant reference information for each candidate, and applies the prompting strategy to rerank them based on the retrieved evidence. This effectively reduces the search space for the LLM reranking process. Unlike prior work [5] that focuses on individually validating each predicted triplet, this paper formulates the task as a reranking problem. This approach has two key reasons: First, this paper positions the LLM as an efficient tool to aid human decision-making rather than a full replacement for human expertise, which we argue is more appropriate for processes where accuracy is highly sensitive and critical. Human evaluators can concentrate on the most promising predictions by leveraging the LLM to quickly organize and rerank the top candidates generated by existing systems. Second, maintaining the ranking framework for KGC predictions allows for a more seamless evaluation of the LLM's impact, as standard ranking metrics can still be applied to assess the effectiveness of the reranking process.

4. Experiments

4.1. Basic Settings

The experiments in this paper are designed to evaluate the effectiveness of the proposed LLM reranking process. Standard ranking metrics commonly applied in KGC tasks are used for comparison. Traditional embedding-based KGC methods are first employed in the experiments to generate baseline results. From these, the top 10 candidate predictions for each KGC query are selected and subsequently reranked using

Table 2
Dataset statistics of KGC prediction rerank task.

Dataset	Source	Entities	Relations	Triples
FB15k-237	Freebase	14,541	237	310,116
InferWiki64k	Wikidata	64,718	239	797,737

Table 3
Reranking results on the FB15K-237 and InferWiki64k datasets, with **TransE** being the original KGC method.

	FB15K-237			InferWiki64k		
	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
Original	12.22%	21.75%	30.34%	8.08%	21.76%	35.54%
GPT-3.5 W/ External	11.26%	19.84%	-	20.10%	29.49%	-
GPT-3.5 W/ Internal	18.27%	25.92%	-	23.70%	31.59%	-
GPT-3.5 W/ Full reference	18.82%	26.52%	-	25.43%	32.86%	-
GPT-4o-mini W/ External	17.77%	25.93%	-	26.94%	32.61%	-
GPT-4o-mini W/ Internal	21.35%	27.37%	-	26.67%	32.20%	-
GPT-4o-mini W/ Full reference	**21.83%**	**28.01%**	-	**28.42%**	**33.54%**	-

the proposed LLM reranking process. The performance before and after reranking is then reported, demonstrating the impact of the LLM reranking method. Since only the top 10 predicted candidates from the KGC tasks are selected for reranking, metrics such as Mean Rank (MR) and Mean Reciprocal Rank (MRR) are no longer applicable for evaluation. Instead, Hit@1 and Hit@3 are reported in the results table to reflect performance.

The experiments are conducted on two datasets commonly used for knowledge graph completion tasks. The first dataset, FB15K-237 [22], is a subset of the Freebase knowledge graph, containing 237 different relations. It is widely adopted as a standard benchmarking dataset for evaluating various KGC methods under the closed-world assumption. The second dataset, InferWiki [27], was designed to enhance the evaluation of KGC methods under the open-world assumption. It includes human-annotated test triplets that indicate whether the triples can be verified through inferential patterns within the KG or by referencing external open-world sources. Positively labelled test triplets are selected as the test set for the experiments, as these annotations are thoroughly verified using available open-world evidence. In automatic evaluations, biases can occur when unknown facts in the knowledge graphs are assumed to be incorrect. The annotations in the InferWiki dataset help minimize this bias, simulating an evaluation in a real open-world setting.

The TransE and RotatE models are chosen as the baseline KGC methods. For each dataset used in the experiments, pipelines from the Pykeen library [28] are used to train and evaluate the traditional embedding-based KGC approaches. The traditional KGC models are trained for 10 epochs with an embedding size of 100, and their performances on the tail prediction task serve as the baselines. The top ten candidates for each tail prediction query are preserved for further reranking using an LLM. GPT-3.5 and GPT-4o-mini are selected as the LLMs for this evaluation. In the LLM reranking process, reasoning paths of up to length 2 are used as internal KG references, while the top three results from the DuckDuckGo search API are selected as external references.

4.2. Results

Table 3 and table 4 present the performance of LLM reranking when different types of evidence are provided, with TransE and RotatE serving as the original KGC methods, respectively. In the tables, full reference refers to when internal graphical reference and external reference from search engines are available for the LLM. In both tables, the LLM reranking process achieves the highest performance across all datasets, significantly improving upon the original predictions made by existing KGC methods.

Table 4
Reranking results on the FB15K-237 and InferWiki64k datasets, with **RotatE** being the original KGC method.

	FB15K-237			**InferWiki64k**		
	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
Original	22.23%	32.66%	39.95%	37.07%	55.83%	67.44%
GPT-3.5 W/ External	15.81%	27.20%	-	35.59%	55.11%	-
GPT-3.5 W/ Internal	21.78%	32.95%	-	36.48%	55.63%	-
GPT-3.5 W/ Full reference	22.89%	34.02%	-	43.49%	60.14%	-
GPT-4o-mini W/ External	23.02%	33.90%	-	49.70%	61.92%	-
GPT-4o-mini W/ Internal	24.84%	34.62%	-	43.63%	57.92%	-
GPT-4o-mini W/ Full reference	**26.34%**	**35.75%**	-	**51.45%**	**63.19%**	-

The more recent LLM at the time of the experiment, GPT-4o-mini, also outperforms its predecessor, GPT-3.5, across all tasks, highlighting how the ongoing evolution of LLMs can lead to continuous improvements in this reranking task. Furthermore, on both datasets, the original KGC method RotatE offers stronger baselines than TransE. The reranking results for RotatE are also significantly higher, primarily due to the superior quality of its initial pool of candidate predictions. Notably, since the LLM reranking process only has access to the top-10 candidates from the existing predictions, the **theoretical upper bound** for Hit@1 and Hit@3 metrics after reranking is exactly the Hit@10 performance of the original KGC method. With this in mind, the LLM reranking with full evidence, especially GPT-4o-mini, comes reasonably close to the theoretical best for Hit@3 when both internal and external references are provided. This observation demonstrates the potential of the proposed reranking process to reduce human effort significantly.

Additionally, it can be observed from table 3 and table 4 that LLMs reranking generally performs better when only internal graphical references are available compared to when relying solely on external references from a search engine. The only exception is GPT-4o-mini on Inferwiki64k, with RotatE as the original KGC method. While a more comprehensive external reference search might improve results, the comparison still supports our initial intuition that internal graphical evidence plays a crucial role in validating and ranking KGC predictions. Interestingly, when relatively earlier LLMs like GPT-3.5 are used, reranking using only external references results in worse performance than the original predictions. This can be attributed to two main factors. First, the quality of search results from the DuckDuckGo search API is not always reliable due to the relatively basic search strategy. When LLMs are provided solely with external references, they may struggle to make accurate judgments without clear or relevant evidence, leading to confusion. Second, the automatic ranking evaluation operates under the closed-world assumption, which treats unobserved facts as incorrect. External references from web searches can introduce open-world knowledge, causing LLMs to rank specific triplets higher based on external evidence. While this behaviour aligns with how LLMs handle open-world information, it leads to predictions being considered incorrect due to the closed-world evaluation criterion. Interestingly, similar effects are not observed on the InferWiki64K dataset. This may be because the test set of InferWiki64K has been validated by human annotators against open-world knowledge. As a result, for each query pattern of (h, r, ?) and (?, r, t), most of the correct triplets are already included in the dataset, creating a scenario that approximates open-world conditions even when evaluated under closed-world assumptions.

4.3. Reranking case studies

Table 5 presents several examples of LLM reranking outputs. These include the original KGC queries, the top 10 candidate predictions from existing methods, and the LLM-reranked outputs and their respective probabilities. For the first query (Parkersburg, timezones, ?), the correct test triplet is (Parkersburg, timezones, Eastern Time Zone), where Parkersburg refers to the town in West Virginia, which is in the Eastern time zone. However, the original KGC embedding method incorrectly ranked

Table 5
Case study examples on *FB15K-237* showing the candidates and their rankings after LLM reranking with internal and external references, correct candidate in bold.

Query	Original candidates	Reranked Candidates with probs
(Parkersburg, timezones, ?)	Greenwich Mean Time Zone Eastern Time Zone Central Time Zone Parkersburg Ken Ralston Librarian Jack Lemmon George Stevens Central European Time Zone Frank Tashlin	Central Time Zone [0.99] **Eastern Time Zone** [0.98] Parkersburg [0.83] Central European Time Zone [0.71] Greenwich Mean Time Zone [0.33] Ken Ralston [0.30] Jack Lemmon [0.20] Librarian [0.17] Frank Tashlin [0.14] George Stevens [0.14]
(Joe Shuster, placeofdeath, ?)	Malibu Manhattan Hawaii The Bronx Hollywood New Jersey Los Angeles Palo Alto New York City Teaneck	**Los Angeles** [1.00] Malibu [0.92] New York City [0.89] Palo Alto [0.85] New Jersey [0.82] Manhattan [0.70] Hollywood [0.67] Hawaii [0.56] The Bronx [0.53] Teaneck [0.48]
(A.I. Artificial Intelligence, genre, ?)	Crime Fiction Romance Film Drama Adventure Film Film adaptation Science Fiction Mystery Animation Melodrama Thriller	Drama [0.98] **Science Fiction** [0.98] Adventure Film [0.93] Animation [0.91] Mystery [0.88] Thriller [0.87] Film adaptation [0.81] Crime Fiction [0.73] Melodrama [0.59] Romance Film [0.59]
(University of Ottawa, majorfieldofstudy, ?)	Mathematics Biology Electrical engineering Chemical Engineering Science Geography Economics Politics Engineering-GB Civil Engineering	Geography [0.99] Civil Engineering [0.99] Mathematics [0.98] Politics [0.97] Biology [0.97] Economics [0.94] **Chemical Engineering** [0.93] Electrical engineering [0.92] Science [0.52] Engineering-GB [0.41]

the Greenwich Mean Time Zone as the top candidate. During the reranking process, the LLM assigned a confidence score of 0.98 to the correct Eastern Time Zone, placing it second behind the Central Time Zone, which received a 0.99 confidence score. While the correct candidate's rank did not improve, it can be observed that under the open-world assumption, the Central Time Zone may also be considered valid, as Parkersburg could refer to the town in Illinois, which is in the Central Time Zone. At the same time, the incorrect prediction of the Greenwich Mean Time Zone is ranked much lower by the reranking process.

For the second query (Joe Shuster, placeofdeath, ?), the correct test triplet is (Joe Shuster, placeofdeath, Los Angeles). The LLM reranking process successfully ranks Los Angeles as the top candidate, whereas the original embedding method struggles to distinguish between various U.S. locations. Interestingly, Malibu and New York City are also assigned relatively high probabilities. While Malibu is reasonable given its proximity to Los Angeles, New York City's high ranking is likely due to it being a notable place where Joe Shuster lived during his lifetime. The observation highlights the importance of treating

the post-processing of KGC predictions as a ranking task instead of individual validation tasks, as binary validation may lead to incorrect conclusions when dealing with partially correct predictions. For the third query (A.I. Artificial Intelligence, genre, ?), the correct genre is Science Fiction in the test set. However, beyond the context of the knowledge graph, Drama is also a valid genre as the movie is often classified as a Science Fiction Drama film. The LLM reranking process correctly ranks the two as top candidates with identical probabilities. The final query (University of Ottawa, major-fieldofstudy, ?) represents a classic one-to-many relationship in the knowledge graph, resulting in multiple valid candidates. In the FB15k-237 dataset, only Chemical Engineering, Computer Science, and Mathematics are listed. However, the search results suggest that the University of Ottawa offers relevant majors across all top-ranked candidates. The LLM ranks all majors with high probabilities when external references are provided. Interestingly, the LLM assigns lower probabilities to broader terms like Science and Engineering, likely because they refer to fields of study rather than specific majors. This observation suggests that the LLM's decision-making process closely aligns with that of human annotators in this context.

5. Limitation & Conclusion

This paper demonstrates the effectiveness of using large language models as a post-processing step to rerank knowledge graph completion predictions. By leveraging LLMs' advanced contextual understanding and their next-token probabilities, the proposed method can refine the initial predictions generated by traditional KGC models. Experimental results on existing KGC benchmarking datasets have shown that LLMs substantially improve the rankings of potential candidates. Using the LLM in the post-processing phase means that future knowledge discovery systems could achieve higher accuracy and completeness without requiring significant manual intervention. This has broad implications for automating knowledge graph curation and completion, as it reduces the reliance on human oversight to vet predictions, making the system more scalable. While the results presented in this paper show promise, several limitations must be acknowledged. The success of the LLM-based reranking process is still dependent on the initial quality of KGC predictions, and while the LLM enhances accuracy, it is not infallible. This approach is analogous to existing information retrieval systems, where an initial dense retrieval phase is typically followed by a reranking stage with cross-encoders. In these systems, the initial retrieval aims to capture a broad set of relevant results, which is crucial for the overall success of the process. A limited number of failure cases can also be observed, especially when the LLMs are presented with ambiguous references or triplets. The imperfect results also demonstrate that at the current stage, using the LLM as a reranking tool to help reduce human efforts presents a more promising path than directly using the LLM to replace human validation of KGC predictions.

Overall, using LLMs to rerank and validate KGC predictions is an important step forward in developing robust and accurate knowledge mining in the KGs. The experimental results demonstrate that the LLM reranking system can reduce human effort. As LLMs continue to advance, they will likely play an increasingly critical role in KG curation pipelines, further minimizing human intervention and enhancing the quality and scalability of knowledge graphs for their downstream applications.

Acknowledgments

This research is funded by the Royal Bank of Canada's Wealth Management Strategy, Products & Digital Investing team.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-3.5 for: Grammar and spelling checks. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] M. Nickel, V. Tresp, H.-P. Kriegel, A Three-Way Model for Collective Learning on Multi-Relational Data (2011).

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 26, Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[3] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling Relational Data with Graph Convolutional Networks, in: A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, M. Alam (Eds.), The Semantic Web, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 593–607. doi:10.1007/978-3-319-93417-4_38.

[4] S. Vashishth, S. Sanyal, V. Nitin, P. Talukdar, COMPOSITION-BASED MULTI-RELATIONAL GRAPH CONVOLUTIONAL NETWORKS (2020) 16.

[5] J. Boylan, S. Mangla, D. Thorn, D. G. Ghalandari, P. Ghaffari, C. Hokamp, Kgvalidator: A framework for automatic validation of knowledge graph construction, ArXiv abs/2404.15923 (2024). URL: https://api.semanticscholar.org/CorpusID:269362826.

[6] E. Huaman, E. Kärle, D. Fensel, Knowledge graph validation, CoRR abs/2005.01389 (2020). URL: https://arxiv.org/abs/2005.01389. arXiv:2005.01389.

[7] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning Entity and Relation Embeddings for Knowledge Graph Completion, Proceedings of the AAAI Conference on Artificial Intelligence 29 (2015). URL: https://ojs.aaai.org/index.php/AAAI/article/view/9491. doi:10.1609/aaai.v29i1.9491, number: 1.

[8] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, Proceedings of the AAAI Conference on Artificial Intelligence 28 (2014). URL: https://ojs.aaai.org/index.php/AAAI/article/view/8870. doi:10.1609/aaai.v28i1.8870, number: 1.

[9] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: C. Zong, M. Strube (Eds.), Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 687–696. URL: https://aclanthology.org/P15-1067. doi:10.3115/v1/P15-1067.

[10] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, in: L. Màrquez, C. Callison-Burch, J. Su (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 705–714. URL: https://aclanthology.org/D15-1082. doi:10.18653/v1/D15-1082.

[11] B. Yang, W. tau Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: International Conference on Learning Representations, 2014. URL: https://api.semanticscholar.org/CorpusID:2768038.

[12] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of The 33rd International Conference on Machine Learning, PMLR, 2016, pp. 2071–2080. URL: https://proceedings.mlr.press/v48/trouillon16.html, ISSN: 1938-7228.

[13] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, AAAI Press, 2016, p. 1955–1961.

[14] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, arXiv:1609.02907 [cs, stat] (2017). ArXiv: 1609.02907.

[15] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 950–958. URL: https://doi.org/10.1145/3292500.3330989. doi:10.1145/3292500.3330989.

[16] V. Tong, D. Q. Nguyen, H. T. Trung, T. T. Nguyen, Q. V. H. Nguyen, N. Mathias, Joint Multilingual Knowledge Graph Completion and Alignment, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022.

[17] W. Zhang, O. Şerban, J. Sun, Y. Guo, Conflict-aware multilingual knowledge graph completion, Knowledge-Based Systems 281 (2023) 111070. URL: https://www.sciencedirect.com/science/article/pii/S0950705123008201. doi:https://doi.org/10.1016/j.knosys.2023.111070.

[18] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, ArXiv abs/1909.03193 (2019). URL: https://api.semanticscholar.org/CorpusID:202539519.

[19] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference 2021, WWW '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1737–1748. URL: https://doi.org/10.1145/3442381.3450043. doi:10.1145/3442381.3450043.

[20] J. Lovelace, C. Rosé, A framework for adapting pre-trained language models to knowledge graph completion, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 5937–5955. URL: https://aclanthology.org/2022.emnlp-main.398. doi:10.18653/v1/2022.emnlp-main.398.

[21] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities, World Wide Web 27 (2024) 58. URL: https://doi.org/10.1007/s11280-024-01297-w. doi:10.1007/s11280-024-01297-w.

[22] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: A. Allauzen, E. Grefenstette, K. M. Hermann, H. Larochelle, S. W.-t. Yih (Eds.), Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Association for Computational Linguistics, Beijing, China, 2015, pp. 57–66. URL: https://aclanthology.org/W15-4007. doi:10.18653/v1/W15-4007.

[23] H. Yang, Z. Lin, M. Zhang, Rethinking knowledge graph evaluation under the open-world assumption, ArXiv abs/2209.08858 (2022). URL: https://api.semanticscholar.org/CorpusID:252367474.

[24] X. Ma, X. Zhang, R. Pradeep, J. Lin, Zero-shot listwise document reranking with a large language model, 2023. URL: http://arxiv.org/abs/2305.02156. doi:10.48550/arXiv.2305.02156. arXiv:2305.02156 [cs].

[25] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, Is ChatGPT good at search? investigating large language models as re-ranking agents, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 14918–14937. URL: https://aclanthology.org/2023.emnlp-main.923. doi:10.18653/v1/2023.emnlp-main.923.

[26] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang, M. Bendersky, Large language models are effective text rankers with pairwise ranking prompting, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 1504–1518. URL: https://aclanthology.org/2024.findings-naacl.97. doi:10.18653/v1/2024.findings-naacl.97.

[27] Y. Cao, X. Ji, X. Lv, J. Li, Y. Wen, H. Zhang, Are missing links predictable? an inferential benchmark for knowledge graph completion, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 6855–6865. URL: https://aclanthology.org/2021.acl-long.534. doi:10.18653/v1/2021.acl-long.534.

[28] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, J. Lehmann, PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings, Journal of Machine Learning Research 22 (2021) 1–6. URL: http://jmlr.org/papers/v22/20-825.html.