Bridging Wikipedia and Wikidata with Graph-LLMs and Retrieval-Augmented Inference to Identify Knowledge Gaps

Djellel Difallah
NYU Abu Dhabi, UAE

Abstract

The integration of structured knowledge bases like Wikidata and textual knowledge platforms such as Wikipedia represents a significant untapped opportunity for advancing knowledge completeness, cross-lingual coverage, and unbiased representation. While Wikipedia provides rich multilingual and multimodal content, Wikidata captures structured, language-agnostic information. This natural complementarity can not only surface critical knowledge gaps to the community but may also facilitate their resolution.

This research proposal aims to establish novel methodologies for tightly integrating Wikidata and Wikipedia through retrieval-enhanced inference mechanisms and graph representation to tackle knowledge gaps. Our research objective is twofold: (i) developing advanced retrieval-augmented models to automatically identify and propose new links within Wikidata by leveraging full textual evidence from Wikipedia; (ii) creating Graph-infused generative techniques using structured data from Wikidata to systematically pinpoint content gaps across multiple language editions of Wikipedia.

Introduction

The Wikimedia projects form a unique ecosystem aimed at capturing the sum of all knowledge across diverse facets and modalities. At the heart of this ecosystem, Wikipedia stands as a multilingual encyclopedia powered by volunteers who continuously create, enrich, and moderate its content. Another project is Wikidata, a collaborative knowledge graph that stores structured, machine-readable information for data-intensive applications. These two platforms are tightly related by design: every Wikipedia article connects to a language-agnostic Wikidata entity, linking textual content with fine-grained, traceable, entity-centric facts. This integration enables networked information dissemination through infoboxes, language links, and other functionalities.

The Challenge and Opportunity

While community efforts have spurred the growth of these projects, with millions of Wikipedia articles and over 120 million Wikidata entities, the rate of information creation outpaces the community's capacity to manage it. This often leads to knowledge gaps (e.g., missing links, outdated information), knowledge inequities (e.g., disparities across languages or regions), and the risk of misinformation spread. Although each project has tailored strategies to mitigate these issues, significant untapped potential exists in synergistic approaches that leverage their combined features.

Wikipedia relies on editors to ensure that its articles are up to date, accurate, well-cited and formatted, and broadly representative across many languages. Recently, many efforts have been made in order to streamline this process with the support of Machine Learning tools to optimize the editors workflow via so called structured tasks. Projects have sought to detect missing citations, links, images, content structure and other targeted tasks; thereby pinpointing critical actions that require community intervention. However, the majority of these efforts have been confined to reasoning at the single-article level, leaving broader and systemic issues largely undetected.

On the Wikidata front, effort to improve the graph completeness can draw directly from the extensive research work in knowledge graph completion, which mainly focuses on predicting missing links between entities. Existing research in knowledge graph completion has made notable progress by developing models that predict unseen edges using structural and textual information. Although current state-of-the-art models have demonstrated promising results, they often encounter scalability limitations when applied to real-world graphs, which is problematic given the size of Wikidata.

While there are multiple types of knowledge gaps, we focus in this proposal on the following definition, which provides a shared objective for both projects.

Definition. *A knowledge gap in Wikidata is a missing link between two entities (i.e., a triple), and in Wikipedia it is a missing passage that mentions a particular fact (i.e., the textual representation of a triple).*

The Proposed Solution

This research proposal aims to improve knowledge completeness across both Wikipedia and Wikidata by developing models that jointly encode text and structured data modalities. We seek to develop an approach for link prediction capable of leveraging deep textual content through retrieval-augmented generation mechanisms. As a byproduct, Wikidata entities can inform multiple enhancements to its corresponding articles in any language; for instance it could be used to improve the article organization, correlating groups of properties with corresponding sections, and identifying common citation sources for specific claims. Specifically, we propose to explore generative methods leveraging Wikidata's structure to identify multilingual knowledge gaps across Wikipedia editions. Lastly, given the large space of possibilities, we aim for a framework that can generate probabilistic knowledge gap indicators, providing a data-driven mechanism to guide community-driven prioritization of content creation/moderation efforts.

In summary, this research aims to address the following research questions:

- How can retrieval-augmented models leveraging Wikipedia content effectively inform new structured link prediction in Wikidata?
- What inference methods based on LLMs and Wikidata can accurately identify multilingual content gaps across different Wikipedia language editions?
- What knowledge gap indicators derived from these models guide community-driven prioritization and content creation efforts on Wikipedia?

Date: Start date September 1, 2025 (1 Year duration).

Related Work

In the following, we provide a broad literature review on knowledge graph completion methods and recent efforts to combine large language models with knowledge graphs, highlighting both our inspiration source and areas of development in the context of Wikipedia and Wikidata projects.

Knowledge Graph Completion

Knowledge graph completion (KGC) research focuses on developing models that can accurately identify new connections (or edges) in a knowledge graph [1]. KGC methods can generally be divided into two categories: structure-based and text-based approaches.

Structure-based methods rely on the graph's topology, analyzing patterns and paths to infer new links. For example, knowledge graph embedding methods are effective at capturing structural patterns but often cannot handle inductive reasoning – such as predicting links for entities that were not seen during training [2]. Path-based methods aim to overcome this limitation by using graph neural networks to enable inductive reasoning [3], though they face challenges in scaling to large, real-world graphs with millions of entities [4].

Text-based approaches, on the other hand, use entity descriptions encoded by pretrained language models. This allows them to handle unseen entities during inference [5]. Recent studies have shown that combining both structural and textual information leads to the best performance [6].

LLMs, RAGs, and Graph RAGs

Large Language Models (LLMs) have significantly advanced natural language processing, thanks to their strong abilities in understanding, reasoning, and generating text. However, models like GPT-4 [7], Qwen2 [8], and LLaMa [9] still face challenges when handling domain-specific knowledge, constantly changing information, rare entities, and private data. These limitations can lead to problems like inconsistent conversations [10], poor explanations [11], and hallucinated (inaccurate or imaginary) content [12, 13]. To improve reasoning, researchers have introduced techniques such as Chain-of-Thought [14], which encourages step-by-step thinking, and Self-Consistency, which helps find the most logical reasoning paths [15].

To overcome the weaknesses of LLMs, researchers have been working on incorporating external information, including knowledge graphs, into language models. One popular method is Retrieval-Augmented Generation (RAG), which retrieves relevant information from targeted sources and incorporates it into the model prompt [16, 17]. This approach helps reduce hallucinations and improves the relevance and accuracy of generated content by grounding the LLM into a specific pool of desired information.

Building on this foundation, researchers have developed Graph-based RAG (GraphRAG), which uses structured knowledge from graphs to provide richer and more connected information [18, 19]. GraphRAG helps by linking related pieces of data, reducing repetition, and supporting focused summaries based on user queries [20].

Knowledge Graph Tasks with LLMs

LLMs have also been applied to improve a range of knowledge graph tasks, such as link prediction, entity alignment, and question answering based on knowledge graphs [21]. Early approaches treated KGC as a text generation task, using sequence-to-sequence models. Models like KGT5 [22] convert graph triples into natural language sentences and then predict missing triples using a pretrained language model such as T5 [23]. These approaches later evolved to include zero-shot learning and instruction tuning in larger language models [24]. However, they still struggle with the same issues as other LLM applications (particularly hallucinations) because the models often lack deep knowledge of specific domains and have difficulty understanding the structured, rule-based nature of knowledge graphs and the relationships between entities.

The PI's Relevant Work

In the context of this project, the PI brings an extensive experience working on Wikidata-related research project, from studying the editors' life-cycle and engagement on the platform by performing a longitudinal analysis of volunteer engagement and comparing activity patterns between power-users and regular users [28], to developing class completeness indicators to estimate if certain classes of entities are complete [29].

The PI worked on developing Wiki2Prop [30], a recommendation system that detects missing properties in Wikidata using a multimodal approach. This system fuses Wikipedia article embeddings and image representations to predict the missing properties that should be added to a given entity. This tool has been available for 4 years and it helps Wikidata editors sift through thousands of potential missing links for a given entity by ranking them based on relevance.[1]

Another project involves identifying relevant passages within Wikipedia that orphan entities (those without a Wikipedia page) could potentially link to [31]. We have released a curated dataset of such fine-grained entities to passage mappings [33]. Ongoing work involves developing realistic and continuous benchmarks for KGC based on Wikidata graph, improving the current state of available datasets which are largely either outdated or incomplete.

The PI has also made contributions to the Structured Tasks framework, developed by Wikimedia's Growth Team.[2] This approach focuses on building edit workflows designed to be easy and well suited for newcomers, and focused on a particular action, for example, adding image captions.[3] In particular, the PI contributed to the AddLink project [32], which uses machine learning and natural language processing tools to process a large set of articles and identify potential passages and mentions that would benefit from inserting a link.

Areas of Development

Given the aforementioned projects and the staggering progress made in LLMs and graph representation, extensions for the integration of these two modalities have emerged. Various Graph LLMs provide frameworks that integrate graph structural information into the semantic space of LLMs. For instance, GraphPrompter[25] transforms graph structures into embeddings that function as soft prompts, directing LLMs in graph-oriented tasks. Similarly, LLaGA [26] converts nodes from graphs into sequences that maintain structural awareness, then projects them into the LLM embedding space through trainable mechanisms. In recent developments, GraphICL [27] has introduced innovative prompt templates enabling general-purpose LLMs to execute graph reasoning tasks without requiring additional training phases.

[1] https://wiki2prop.toolforge.org/

[2] https://www.mediawiki.org/wiki/Growth/Personalized_first_day/Structured_tasks

[3] https://www.mediawiki.org/wiki/Growth/Personalized_first_day/Structured_tasks/Add_an_image

Despite these breakthroughs, we are still lacking models tailored to the needs of Wikipedia development, particularly from the perspective of facilitating the workflow of editors through new interfaces such as conversational structured tasks.

Methods

To tackle our research questions, we will follow a systematic approach where we first assemble the necessary data and prepare it, then build our retrieval and graph-inference models, develop new means of estimating knowledge incompleteness, and finally assess their effectiveness both quantitatively and qualitatively.

GRAPH-TEXT Data collection and preprocessing

We will start by downloading the latest monthly data dumps from Wikidata and at least 5 Wikipedia language editions from different families. After that, we will collect data according to specific criteria.

- We focus on entities with at least 1 Wikipedia page across the selected set of languages. This choice excludes orphan entities, but strikes a balance between structured graph representations and rich textual context.
- We limit our dataset to a curated subset of Wikidata properties deemed most relevant to knowledge representation, excluding properties intended for editorial purposes. These exclusions include maintenance tags, external identifiers, and other metadata that do not directly influence the core relational structure of the graph. By focusing on core attributes, we enable more accurate and meaningful evaluations of knowledge graph completion models. This set has approximately 800 properties, which presents a significant challenge for most models handling multi-relational graphs.
- In addition, we integrate supplementary content from Wikipedia by mapping Wikidata entities to their corresponding articles across multiple languages through the use of *sitelinks*. This integration enriches the dataset with full textual information.

We note that Wikipedia dumps do not readily provide a one-to-one mapping between Wikidata entities and Wikipedia articles; hence, a join operation between the two datasets is necessary using sitelinks, hence requiring development of a new tool to construct the GRAPH-TEXT dataset.

Retrieval-augmented link-prediction

To enable Retrieval-Augmented Generation (RAG) for link prediction, we will build a dense passage retrieval system over multilingual Wikipedia. The corpus will be split into fixed-length spans and embedded with state-of-the-art semantic models into a dense vector space, then indexed in a vector database for efficient lookup. As a baseline we will fine-tune a dual-encoder retriever on (head, relation, tail) triples, using relevant passages as positives and strategically sampled negatives for contrastive learning. At inference time, the top-k passages retrieved for each triple will feed into a sequence-to-sequence generator that proposes candidate tail entities. A lightweight scorer will then rerank those candidates.

To benchmark our RAG pipeline, we will also compare two alternative strategies. First, we will assess scalable embedding-only models to isolate the performance from retrieval. Second, we will implement an LLM reader-only approach: framing link prediction as a binary classification task where an agent LLM, given a candidate triple and its supporting passages, decides whether to add the link to Wikidata. This evaluation will compare the strengths and complementarity of generative, retrieval-based, and classification-based methods.

Graph-infused LLM Assistant

Next, we will develop a hybrid model that integrates a graph encoder for Wikidata with a large language model, enabling native reasoning alongside graph-structured data. Specifically, we will explore conventional knowledge-graph encoding techniques to produce entity and relation embeddings, then map those embeddings into the LLM's token-embedding space via a lightweight projection layer. The entire architecture will be fine-tuned end-to-end on graph tasks (e.g., link prediction and question answering) so that the LLM learns to incorporate and reason over the injected graph information.

This design enables the model to leverage three components: Wikipedia's textual context (the input), Wikidata's structured facts (the graph), and the LLM's language and reasoning capabilities. As a pilot application, we will use this framework to detect missing mentions of specific Wikidata facts in a given Wikipedia language edition. In principle, the same hybrid approach can be adapted for other objectives, such as predicting article structure or recommending section groupings, by simply changing the fine-tuning task or prompt templates.

Knowledge Gap Indicators

The space of potential knowledge gaps in both Wikidata and Wikipedia is very large, with N entities and R relations, every possible triple is a potential knowledge gap, hence exhaustive evaluation is impractical. This research will explore high-level indicators to surface a top-k list of true positive knowledge gaps. Structural signals may include link-prediction confidence scores from hybrid graph-LLM models, under the hypothesis that higher scores correlate with truly missing links. For textual gaps, we will examine LLM-generated logits, where a higher probability reflects how a fact fits into a target-language article's context. We will also investigate auxiliary metrics, such as retrieval frequency from dense indices and disparity measures comparing property sets across languages to define robust indicators. Finally, calibrating the usage of such indicators on a held-out validation set will help building structured tasks balancing precision and recall for the use of the community.

Evaluation

We measure link-prediction performance using standard ranking metrics, including mean reciprocal rank (MRR) and Hits@{1,3,10} on our held-out triples. For textual knowledge-gap detection, we will compare model predictions against a manually annotated dataset, reporting precision, recall, and F1-score. To enable larger-scale evaluation, we will also generate an auxiliary test set via distant supervision by extracting entity pairs that are explicitly linked within the same paragraph.

To validate our indicators in a real-world setting, we will develop a Minimum Viable Product for knowledge-gap identification and deploy it to a small group of community editors or domain experts. Their feedback and editing actions will serve as a direct benchmark, providing both qualitative insights and quantitative measures of each step in the pipeline, allowing us to refine our methods.

In summary, we organize the research methodology into the following work packages:

WP1 [Datasets]: GRAPH-TEXT Dataset Construction: Build an aligned dataset by extracting triples over a select set of 800 Wikidata relation types and joining them with sitelink-mapped multilingual Wikipedia passages.

WP2 [Infrastructure]: Retrieval Infrastructure: Implement text parsing, passage splitting, semantic encoding, and low-latency vector indexing for efficient multilingual retrieval.

WP3 [Research]: RAG-based Link Prediction: Fine-tune and benchmark a dual-encoder retriever, seq2seq generator, and two baselines for link prediction using an LLM Reader.

WP4 [Research]: Hybrid Graph-LLM Modeling: Develop and fine-tune a projection-based hybrid graph-LLM model to reason jointly over text and relational graph embeddings.

WP5 [Metrics]: Evaluation, Knowledge Gap Indicators: Calibrate gap-detection scores, benchmark link-prediction and textual gap metrics, and evaluate the interface with Wikidata and Wikipedia editors.

Expected output

Scientific publications: We expect at least two scientific papers, (i) on the hybrid graph-LLM architecture for Wikipedia, targeting NLP venues such as ACL/EMNLP, (ii) will be focusing on link prediction, targeting venues such as SIGIR/KDD.

- The targeted audience: Researchers working on knowledge completion and hybrid language models.

Datasets: We plan on distributing the datasets output of WP1 (Graph-TEXT), which would be a welcome addition to the community working on KGC problems. The Wikipedia vector database from WP2 can also be expensive to build and maintain continuously (we expect roughly 100 Million documents), for this reason we will release the code base to create the entire index from dumps.

- The targeted audience: Researchers, tool-builders, and data scientists can directly benefit from this data to create visualization, explore and analyze the data, or develop new models.

Minimum Viable Product. We will create a demo for knowledge gap detection for both Wikipedia and Wikidata. See figure 1 for a sketch of the Wikidata completion tool.

- The targeted audience: The editor community at large can utilize the tool to identify potential actions for content of improvement.

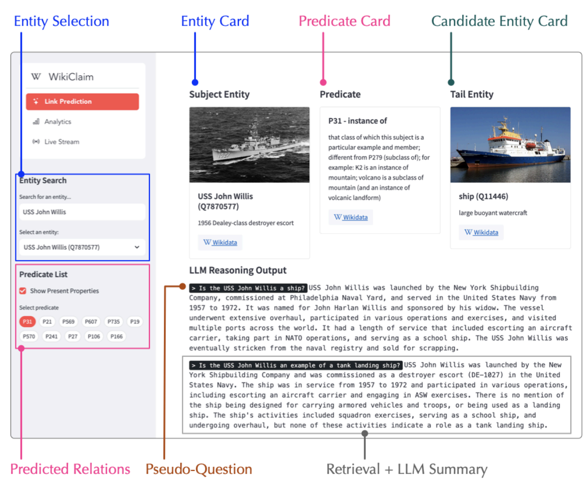

Fig1. Screenshot of the Wikidata Link recommendation prototype.

Risks

Data quality and coverage: Wikipedia content quality varies widely across languages; this can hamper the development of robust models. Similarly, Wikidata suffers from noise, duplicate items, and a power-law node-degree distribution – where most entities have, on average, only two properties. To mitigate these issues, we can implement preference-based pipelines: (i) prioritize Wikidata link prediction on well-documented but poorly connected entities, and (ii) focus on Wikipedia knowledge gap detection for articles tied to well-connected Wikidata entries.

Model hallucination and bias: Fine-tuning LLM models can be notoriously difficult; the proposed hybrid graph-LLM may poorly generalize, overfit, or be biased toward rich entities. Here, we will utilize standard approaches for improving training and consider several optimization strategies and objectives.

Scalability: From our prior experience working on RAG-based pipelines involving millions of documents, the sheer size of the database can be daunting if not equipped with the right infrastructure. We will leverage efficient vector-search libraries (e.g., FAISS, Milvus) and find an optimal trade-off when selecting the embedding size. The planned MVP will run in the cloud.

Community alignment: Our ultimate goal is to surface knowledge gaps; however, these may not match editor priorities (even if they are true positives), or may not be actively adopted. As the project evolves into an MVP, we will gather qualitative feedback and adjust the ranking based on specific requests from the community.

Personnel expertise and capacity: The PI primarily works with undergraduate students and provides them with structured training and mentorship. The priority will be given to recruiting students in their Senior year and having Machine Learning experience. If necessary, we will explore hiring a part-time research assistant to supplement expertise.

Community impact plan

Our project will engage Wikipedia editors, developers, and affiliates through interactive workshops, hands-on tutorials, and live demonstrations. As one of the expected outcomes is an MVP, we will partner with volunteer communities and ambassadors from select Wikipedia language editions to integrate gap-detection recommendations into their editing workflows, maintain documentation and support channels to promote adoption, and gather feedback and requirements. We will apply a similar approach with the Wikidata community. In particular, developers creating data-ingestion bots will benefit from access to knowledge-gap indicators to inform the design of new data-extraction pipelines.

Evaluation

We will assess our project's success against four concrete deliverables with required functionalities and ideal (stretch) goals.

Data-Extraction Toolkit: The data preparation toolkit must seamlessly process Wikipedia/Wikidata snapshots to produce the Graph-Text corpus. If the size is manageable, we will ideally publish the precomputed dataset.

RAG Infrastructure: The code to recreate the retrieval-augmented generation pipeline is published (with documentation) and can be reproduced in the cloud or a self-hosted instance.

Method Contributions: The two research work packages WP3 and WP4 will be implemented using at least regular baselines, plus at least one (ideally two) novel contributions, to target a top-tier conference.

MVP Interface: A live web UI with 1,000 selected Wikipedia articles, or Wikidata entities, must be deployed and evaluated by volunteer editors or human experts.

Budget

The budget will mainly cover the following project costs

- Student research assistants.
- LLM API usage and cloud deployment.
- PI summer salary.
- Conference Travel.

Details can be found in the following sheet.

Research Fund Budget Template

References

[1] Hogan, Aidan, et al. "Knowledge Graphs." *ACM Computing Surveys*, vol. 54, no. 4, 2022, article 71, pp. 1–37.

[2] Rossi, Andrea, et al. "Knowledge Graph Embedding for Link Prediction: A Comparative Analysis." _ACM Transactions on Knowledge Discovery from Data_, vol. 15, no. 2, 2021, art. 14, pp. 1–49.

[3] Zhu, Zhaocheng, et al. "Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction." Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 29476–29490.

[4] Zhang, Canlin, and Xiuwen Liu. "Inductive Link Prediction in Knowledge Graphs Using Path-Based Neural Networks." *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2024, pp. 1–9.

[5] Jiang, Pengcheng, et al. "Text Augmented Open Knowledge Graph Completion via Pre-Trained Language Models." *Proceedings of the ACL (Findings)*, Association for Computational Linguistics, 2023, pp. 11161–11180.

[6] Wang, Liang, et al. "SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-Trained Language Models." *Proceedings of ACL (1)*, Association for Computational Linguistics, 2022, pp. 4281–4294.

[7] OpenAI. "GPT-4 Technical Report." *CoRR*, vol. abs/2303.08774, 2023.

[8] Yang, An, et al. "Qwen2 Technical Report." *CoRR*, vol. abs/2407.10671, 2024.

[9] Dubey, Abhimanyu, et al. "The Llama 3 Herd of Models." *CoRR*, vol. abs/2407.21783, 2024.

[10] Adiwardana, Daniel, et al. "Towards a Human-like Open-Domain Chatbot." *CoRR*, vol. abs/2001.09977, 2020.

[11] Camburu, Oana-Maria. "Explaining Deep Neural Networks." PhD diss., University of Oxford, 2020.

[12] Huang, Lei, et al. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions." *CoRR*, vol. abs/2311.05232, 2023.

[13] Zhang, Yue, et al. "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models." *CoRR*, vol. abs/2309.01219, 2023.

[14] Wei, Jason, et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2022)*, vol. 35, 2022.

[15] Wang, Xuezhi, et al. "Self-Consistency Improves Chain-of-Thought Reasoning in Language Models." *Proceedings of the International Conference on Learning Representations (ICLR)*, OpenReview.net, 2023.

[16] Fan, Wenqi, et al. "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models." *Proceedings of the 30th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2024, pp. 6491–6501.

[17] Gao, Yunfan, et al. “Retrieval-Augmented Generation for Large Language Models: A Survey.” *CoRR*, vol. abs/2312.10997, 2023.

[18] Edge, Darren, et al. “From Local to Global: A Graph RAG Approach to Query-Focused Summarization.” *CoRR*, vol. abs/2404.16130, 2024.

[19] Guo, Zirui, et al. “LightRAG: Simple and Fast Retrieval-Augmented Generation.” *CoRR*, vol. abs/2410.05779, 2024.

[20] Peng, Boci, et al. “Graph Retrieval-Augmented Generation: A Survey.” *CoRR*, vol. abs/2408.08921, 2024.

[21] Zhu, Yuqi, et al. “LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities.” *World Wide Web*, vol. 27, no. 5, 2024, p. 58.

[22] Saxena, Apoorv, et al. “Sequence-to-Sequence Knowledge Graph Completion and Question Answering.” *Proceedings of ACL (1)*, Association for Computational Linguistics, 2022, pp. 2814–2828.

[23] Raffel, Colin, et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *Journal of Machine Learning Research*, vol. 21, no. 140, 2020, pp. 1–67.

[24] Yao, Liang, et al. “Exploring Large Language Models for Knowledge Graph Completion.” *CoRR*, vol. abs/2308.13916, 2023.

[25] Liu, Z., et al. “Can We Soft Prompt LLMs for Graph Learning Tasks?” *Companion Proceedings of the ACM Web Conference 2024*, Association for Computing Machinery, 2024, pp. 481–484.

[26] Chen, R., et al. “LLAGA: Large Language and Graph Assistant.” 2024.

[27] Sun, Y., et al. “GraphICL: Unlocking Graph Learning Potential in LLMs through Structured Prompt Design.” 2025.

[28] Sarasua, Cristina, et al. “The Evolution of Power and Standard Wikidata Editors: Comparing Editing Behavior over Time to Predict Lifespan and Volume of Edits.” *Computer Supported Cooperative Work*, vol. 28, no. 5, Sept. 2019, pp.

[29] Luggen, Michael, et al. “Non-Parametric Class Completeness Estimators for Collaborative Knowledge Graphs—The Case of Wikidata.” *ISWC 2019*, Springer, 2019, pp. 453–469.

[30] Luggen, Michael, et al. “Wiki2prop: A Multimodal Approach for Predicting Wikidata Properties from Wikipedia.” *Web Conference 2021*, pp. 2357–2366.

[31] Ostapuk, Natalia, et al. “ParaGraph: Mapping Wikidata Tail Entities to Wikipedia Paragraphs.” *2022 IEEE International Conference on Big Data*, pp. 6008–6017, 2022.

[32] Gerlach, Martin, et al. “Multilingual Entity Linking System for Wikipedia with a Machine-in-the-Loop Approach.” *CIKM* 2021, pp. 3818–3827.

[33] Ostapuk, Natalia, Djellel Difallah, and Philippe Cudré-Mauroux. "SectionLinks: Mapping Orphan Wikidata Entities onto Wikipedia Sections." *Proceedings of the 1st Wikidata Workshop (Wikidata 2020)*, vol. 2773, CEUR-WS.org, 2020, pp. 1–15.