LLM-SPARQL: A Hybrid Framework for NEL and NED to Wikidata

Muhammad Salman, Haoting Chen, Sergio J. Rodríguez Méndez, Armin Haller

Muhammad.Salman@anu.edu.au School of Computing, Australian National University, Canberra Australia

Australian National University

CCKS-IJCKG 2024

Overview

SALMON This research tackles the complex issue of Named Entity Disambiguation (NED) by harnessing the power of LLMs, and integrating these with structured information from KG like Wikidata. The overarching goal is to markedly enhance NED accuracy through a synergistic approach that combines the predictive prowess of advanced LLMs with the detailed semantic repositories of KGs. Our key innovation is the formulation of a novel methodology that em- ploys SPARQL queries to methodically retrieve potential Uniform Resource Identifiers (URIs) from Wikidata. We then apply cutting-edge generative AI techniques to accurately match each entity's mention within texts to the most contextually relevant URI. This refined approach allows for a more sophisticated, context-aware resolution of entity references. We conducted extensive experiments to evalu- ate our method, which indicates a substantial achievemnet in performance, as evidenced by precision, recall, and F-measure.

SPARQL Query

SPARQL Query for Candidate Extraction

```
SELECT ?entity ?type ?Description WHERE
{ ?entity rdfs:label "entity_label"@en .
OPTIONAL {{?entity wdt:P31 ?type.}
{?entity schema:description ?Description.}}
Filter(!bound(?type) || ?type!=wd:Q4167410)
SERVICE wikibase:label { bd:serviceParam
wikibase:language "en".} } limit (1...n)
```

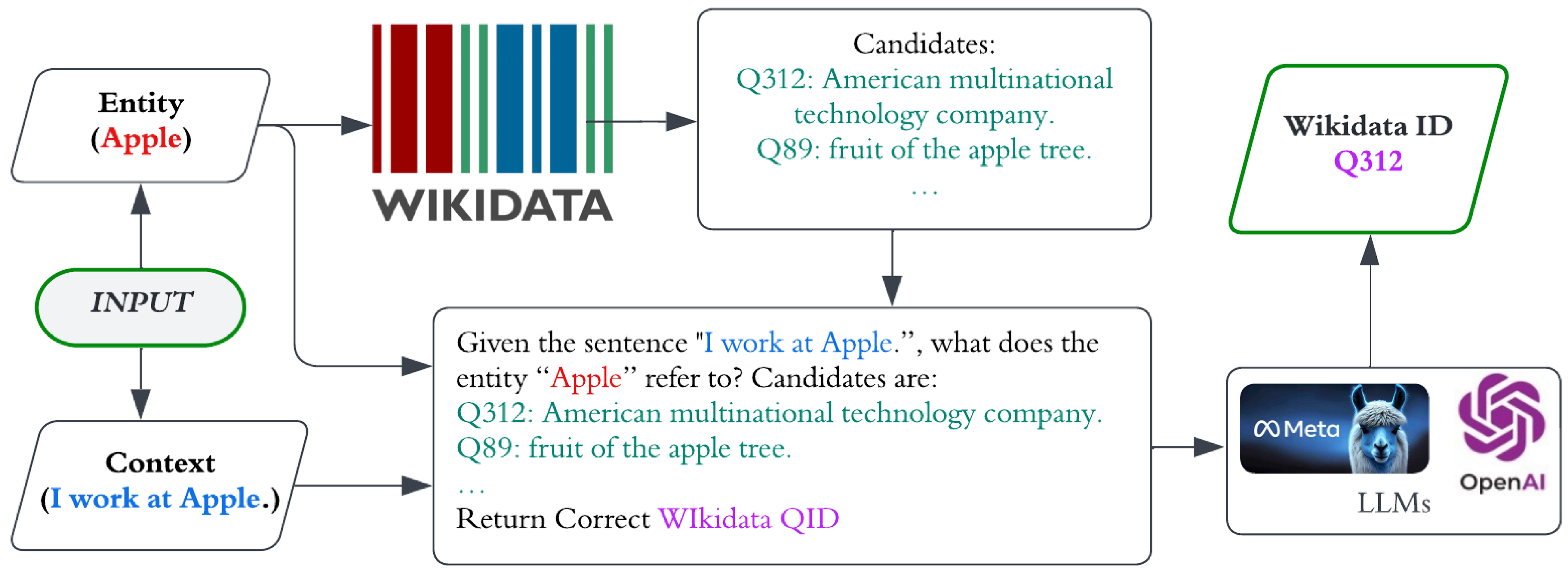
Evaluation and Dataset

For the evaluation of our entity linking and disambiguation approach, we utilised the Wikidata-Disamb dataset, which was developed by converting the Wiki-Disamb30 dataset. The Wiki-Disamb30 dataset, initially curated by Ferragina and Scaiella, was designed for linking entities to their correspond- ing Wikipedia pages. The Wikidata-Disamb dataset extends this by mapping these Wikipedia pages to their respective Wikidata items (QIDs), as documented by Cetoli et al.

Conclusion

This study embarked on a quest to refine NED techniques by integrating LLMs with SPARQL queries. Throughout our exploration, we developed and assessed multiple approaches, each present- ing an evolution over the last in terms of sophistication and performance metrics. The comparative analysis of these approaches culminated in RecursiSelect, which showcased a recursive response of LLMs for candidate selection — demonstrating exemplary precision and recall across various data samples. While initial methods provided a foundational understanding of the challenges inherent in NED tasks, each subsequent approach incorporated additional refinements that progressively en- hanced the model's accuracy. The consistency in high-performance metrics emphasises the reliability and efficiency of this model.

LLM-SPARQL Methodology



NED Approach Refinement Overview

Approach	Method	Key Feature
LLMPredictURI	Direct URI Retrieval with LLM	utilises only LLM for URI prediction
SPARQLite	Enhanced SPARQL Querying	Includes case sensitivity handling
ClassMatch	Class Match	Leverages LLM for class matching in SPARQL results
CandidSelect	LLM-Enhanced Candidate Selection	Combines LLM prompts with candidate selection
RecursiSelect	Recursive CandidSelect	Recursive candidate generation with LLM response

LLM's Prediction and Selection Prompts

Ask_GPT_QID(prediction)

Input:

- *entity_label*: The label of the entity.
- *text*: The sentence that 'mentions' the entity.
- *wikidata_class*: The class of the entity.

Prompt:
Given the reference information and examples below, return the Wikidata ID of `'entity_label'` **Text**: `{text}` **Label**: `{Entity_Label}` **Class**: `{Wikidata_Class}`
Wikidata ID: `?`
Ex 1: Text: *Besides CSIRO, Australian National University is located in Canberra.*
Label: *Australian National University*
Class: *University* Wikidata ID: *Q127990*
Ex 2: Text: *Mantell was born in Bridgwater, Somerset, and studied at the University of Bath.*
Label: *Bridgwater*
Class: *Town* Wikidata ID: *Q914015*
Return `Wikidata_ID`

Ask_GPT_QID(selection)

Input:

- *entity_label*: The label of the entity.
- *text*: The sentence that 'mentions' the entity.
- *Candidates*: Candidate QIDs

Prompt: I am analyzing `'entity_label'`. Consider the reference sentence: `'query_text'` and the candidates Wikidata IDs: `List[Candidates]`
Based on the class, description, and the context provided by the text, what is the correct ID for `'entity_label'`?
Return `'NA'` if none of the candidates is correct.
Return *Wikidata_ID*

NED Approach Refinement Overview

Approach	Precision	Recall	F-measure
LLMPredictURI	27%	100%	42%
SPARQLite	50%	60%	57.98%
ClassMatch	68%	100%	80.95%
CandidSelect	79%	100%	88.27%
RecursiSelect (10)	88%	100%	93.61%
RecursiSelect (n)	92%	100%	95.83%

Test	Precision	Recall	F-measure
1	93%	99%	96.37%
2	**98%**	**100%**	**98.99%**
3	89%	98%	94.18%
4	92%	100%	95.83%
5	91%	97%	95.29%

Models' Performance

The further evaluation of the final refined approach, denoted as RecursiSelect, was conducted us- ing five distinct random 1000-entry samples from the dataset to assess the robustness and consis- tency of the entity disambiguation methodology. The performance metrics under consideration were Precision, Recall, and F-measure, which together offer a comprehensive view of the model's effectiveness.

	Model & Prompt	Precision	Recall	F-measure
OpenAI GPTs	**GPT-3.5-Turbo** *(prediction)*	18%	100%	30.51%
	Class_label + GPT-3.5-turbo *(prediction)*	18%	100%	30.51%
	GPT-4-Turbo *(prediction)*	27%	100%	42%
	Class_label + GPT-4-Turbo *(prediction)*	30%	100%	46.15%
	Case_Sensitivity + GPT-4-Turbo *(prediction)*	33%	100%	49.62%
	Case_Sensitivity + GPT4-turbo (Label, Class) *(prediction)*	36%	100%	52.94%
	SPARQL Candid (top 10) & GPT-4-turbo *(selection)*	77%	100%	87%
	SPARQL Candid (all) & GPT-4-turbo *(selection)*	79%	100%	88.27%
	SPARQL Candid (all) & GPT-4-turbo *(selection + recursive)*	**92%**	**100%**	**95.83%**
	SPARQL Candid (10) & GPT-3.5-turbo *(selection + recursive)*	91%	100%	95.29%
	SPARQL Candid & GPT-4-turbo *(selection + recursive)* MAX	**98%**	**100%**	**98.99%**
Meta's LLaMA	SPARQL Candid, LLaMA2-7B-Q4	40%	98.9%	57.14%
	SPARQL Candid, LLaMA2-7B-Q4 (Optimised)	59%	100%	74.21%
	SPARQL Candid, LLaMA2-7B-Q4 (Prompt Update)	68%	100%	80.95%
	SPARQL Candid, LLaMA3-8B-Q4	77%	100%	87%
	SPARQL Candid, LLaMA3-8B-instruct-Q4	**93%**	**100%**	**96.37%**