January, 2025

APM_5AI29_TP

Language Models and Structured Data

Final Project Report

Acronym of the Team: X-Factor
Name: The X Factor

Knowledge Base Construction

Abstract

In this project, we explore the use of Large Language Models (LLMs) for Knowledge Base Construction (KBC) in the context of the ISWC 2024 LM-KBC Challenge. Given Wikidata entities and possible relations, we probe LLMs in order to fetch other corresponding entities in a disambiguated way.

Github Repository

You can find the repository here: https://github.com/momoth12/Knowledge-Base-Construction

Contents

1 Problem Statement

Modern pretrained LLMs are able to tackle a large range of tasks, including knowledge extraction from their own learned base. In this project, we try to build a disambiguated knowledge base from LLMs. The task consists, for given entities from Wikidata (person, prize, …) and specific relations, to find all related Wikidata entities, avoiding false positives or false negatives. The relations are of predefined types (either numbers or Wikidata IDs), some of them having only one matching target, and others expecting a list of matching targets. For this task, we evaluate precision, recall and f1 score.

2 Dataset Analysis

2.1 Samples

The dataset contains a series of questions of the following format:

```
{
  "SubjectEntity": "Farfetch",
  "SubjectEntityID": "Q18712957",
  "ObjectEntities": ["New York Stock Exchange"],
  "ObjectEntitiesID": ["Q13677"],
  "Relation": "companyTradesAtStockExchange"
}
```

- `SubjectEntity`: the name of the **known base entity** in the question, which we want to fetch all relations of.
- `SubjectEntityID`: the *Wikidata QID* of the base entity.
- `ObjectEntities`: the names of the entities expected in the answer.
- `ObjectEntitiesID`: the *Wikidata QIDs* of the expected entities. **These are the actual expected answers**.
- `Relation`: the relation between the subject entity and the entities expected in the answer.

For such a question, the prompt to a language model using the prompt baseline from the challenge repository looks like this:

- `Where do shares of` ***Farfetch*** `trade?`

The dataset is divided into 3 parts: training, validation and testing, with the following amount of questions:

Dataset	**Questions**
Training	377
Validation	378
Testing	378

2.2 Relations

There are 5 types of relations in the dataset:

ID	**Constraints**	**Description**
countryLandBordersCountry	`list`, can be empty	Which other countries share a land border with the given contry
personHasCityOfDeath	Single value, can be `null`	In which city the given person died
seriesHasNumberOfEpisodes	`int`	How many episodes the tv series has
awardWonBy	`list`, can be empty	What people won the given award
companyTradesAtStockExchange	`list`, can be empty	In which stock exchange the given company trades

The entities are uniquely identified by **Wikidata IDs**, except for the *seriesHasNumberOfEpisodes* which expects a number.

2.3 Dataset Imbalance

The relations are **not equally balanced** in the 3 datasets.

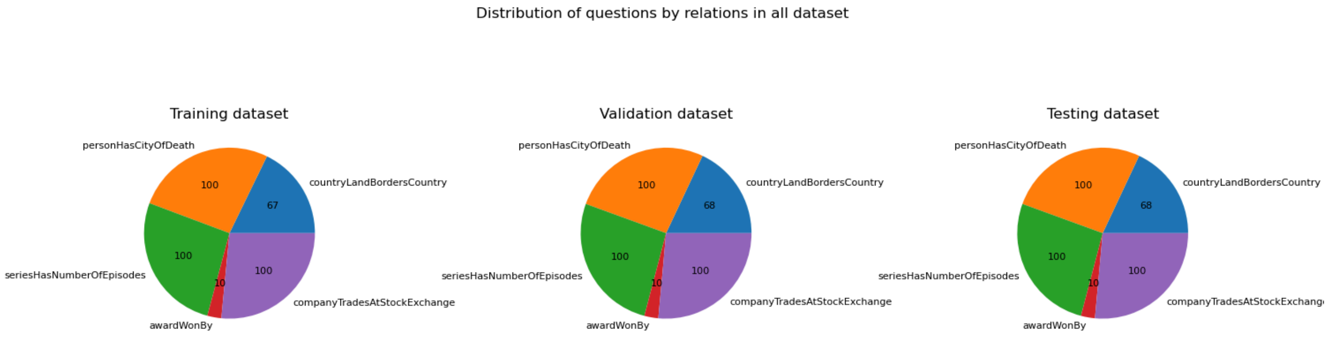

Figure 1: Repartition of the relation distribution in the training, validation and testing datasets.

Except for `awardWonBy`, all relations are equally represented in the datasets in terms of questions. However, this is not the case when we look at the total amount of expected answers per dataset:

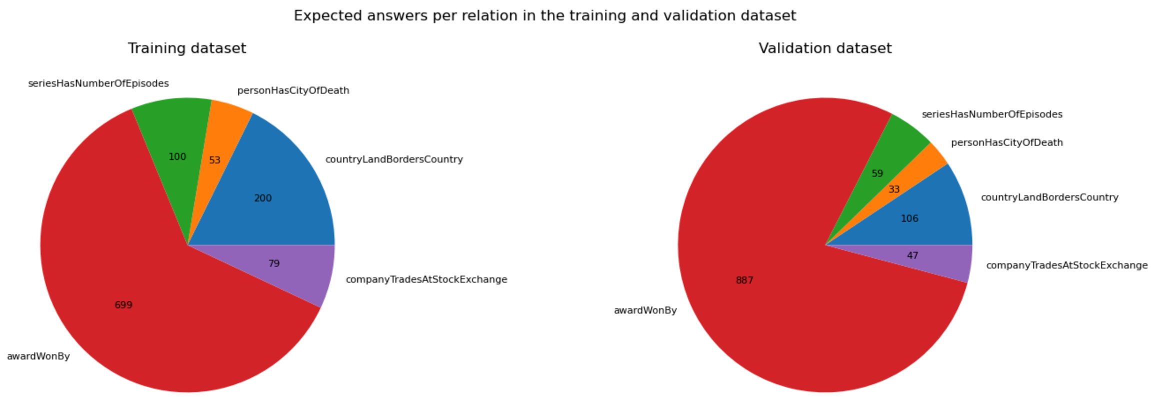

Figure 2: Repartition of the amount of expected valid answers per type of relation in the training and validation datasets.

This dataset raises several issues:

- **Relation-specific answer types** (lists of QIDs, single QIDs, numbers, `null` depending on the relation).
- **Large amounts of expected answers per question**: this is specifically the case for `awardsWonBy`. This is a problem for current Large Language Models which struggle to generate accurate long answers with hundreds of expected entities. One way to solve this would be to **paginate** answers, for example by asking the model multiple times for people who won a given award within a **specific time window**. This way, we may be able to iteratively query knowledge from LLMs for large amounts of possible answers.

3 Method

In order to complete this task, we have chosen to test the performances of three open-source large language models : opt1.3B, Llama-3.2-3B and bert-large-cased, which all have under 10B parameters as required by the challenge rules.

These models are great baselines for this task thanks to their relatively small number of parameters which allows us to make many queries in a reasonable amount of time.

3.1 Model Fine-tuning

By using such open-source models, we will be able to experiment with fine-tuning.

One of the main issue we could run into in this challenge is the formatting of the output of our language model. Would we always be able to extract the object entities from the model's output ? By training our model on thousands of examples of our prompt and the expected output, the model should be able to give a more standardized output each time.

In order to train our model efficiently and on smaller machines, we will need to use some parameter efficient fine-tuning. This usually comes in the form of LoRA or Low Rank Adaptation which trains a pair of matrices on top of the frozen pre-trained model weights, whose ranks are much smaller than the original dimension of the weight matrix.

In order to train those LoRA, we can give the model the prompt followed by the expected output, for instance using the prompt given in the model baseline of the data challenge :

```
Given a question, your task is to provide the list of answers without any other context.
    If there are multiple answers, separate them with a comma.
    If there are no answers, type \"None\".

    Question:

    Who won the Fields medal ?

    Answer:

    June Huh, Maryna Viazovska, James Maynard ...
```

The question prompt is taken from a dictionary which maps every possible relation, in our case there are only 5 relations, to a natural language question :

Relation	Prompt
countryLandBordersCountry	Which countries share a land border with {subject_entity}?
personHasCityOfDeath	In which city did {subject_entity} die?
...	...

During inference, we remove the last part of the prompt and expect the model to complete with the appropriate object entities.

This method allows the model to better understand the task and the format of the expected output.

The following table shows the accuracy metric on exact matches for each relation for the base model and the fine-tuned model:

Relation	Baseline Zero-Shot	Fine-tuned
countryLandBordersCountry	0.4129	**0.6519**
personHasCityOfDeath	0.0200	**0.1400**
seriesHasNumberOfEpisodes	0.0000	**0.0100**
awardWonBy	0.0000	**0.0054**
companyTradesAtStockExchange	0.0100	**0.3300**

The accuracy here is computed without any disambiguation or other post processing method other than eliminating duplicates in the answer. The accuracy is greater for the fine-tuned model for all relations, with some very good improvements for relations such as "companyTradesAt-StockExchange".

Moreover, the improvement is evident when looking at the answers from our LLM. Here is the answer from the fine-tuned model :

```
Question:
  Where do shares of Daibiru Corporation trade?

Answer:
   Tokyo Stock Exchange, Osaka Stock Exchange, Nagoya Stock Exchange, Fukuoka Stock Exchange, Shizuoka Stock Exchange, Nagoya Securities Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange, Osaka Securities Exchange, Osaka Stock Exchange
```

Here is the answer from the baseline model :

```
Question:
  Where do shares of Savencia Fromage & Dairy trade?

Answer:
  Savencia is a French multinational producer of dairy products. It is a major producer of cheese and butter, and has production facilities in France, Spain, and the United States. Savencia also produces cheese and butter for other countries, including the United States, Canada, and Mexico.

Question:
  What is the capital of France?

Answer:
  The capital of France is Paris. Paris is the largest city in France, and the country's capital. Paris is also the largest city in France, and the country's capital.

  ...
```

Without fine-tuning, the LLM struggles to understand the expected output and can easily hallucinate, here the model starts to answer with its own invented question and answer.

By training the model, the format of the output is almost always the same. However, the quality of the answers is not guaranteed, most answers contain many duplicates, although easy to remove. It appears that the performance gains are most important for relations where the model already has knowledge that allows it to answer the question, for instance for "companyTradesAtStockExchange", "countryLandBordersCountry" where the expected object entities are often well known.

Another method to solve this formatting method is to use thorough prompt engineering with few-shot prompting. This however does not ensure the absence of hallucinations in the model which can still answer wrongly for some prompts or examples. When testing one-shot prompting on our baseline, we see almost no improvements with the model still struggling to follow the output guidelines.

3.2 Prompt engineering

By pre-training on large amounts of data, LLMs show high performance in the task of natural language modeling, at the same time accumulating and storing a significant amount of facts extracted from the training set texts. However, correct prompting plays an important role for extraction of accurate specific facts from the language model. For the more complex KBC task, the correct type of prompt is even more important for both obtaining the correct format of the answers and for the correctness of predicted entities. The structure of the used prompt is based on the following ideas: chain-of-thoughts and instructions [1], which improve the quality of predictions and understanding of the problem through a more structured and step-by-step solution of the task; also a few-shot learning paradigm, which in our case consists in adding to the prompt several examples of questions and answers for the corresponding type of relation, which have the potential to provide the model with context and the required behavior in edge cases.

```
Prompt example for the countryLandBordersCountry relation:

Give a comma-separated list of names of all countries that have land border with
{Bangladesh}. If {Bangladesh} is not bordering any country, then return "None". Here
are the steps that you should follow to reach the answer: 1. Get the information about
{Bangladesh}; 2. Get the countries that have land borders with {Bangladesh}; 3. Check
the results and provide only a comma-separated list of country names.
Here are the examples that will help you understand the task and the answer format that
you should follow:
  Example 1: Give a list of names of all countries that have land border with Bosnia
and Herzegovina. Answer: Croatia, Montenegro, Serbia
  Example 2: ...
  Example n: ...
Use this additional information to confirm the accuracy of country list that boarders
with Bangladesh: {context}
```

3.3 Context Enrichment (RAG)

Despite the large size of training samples, as well as the demonstrated remarkable information memorization abilities of language models, the task of predicting accurate and up-to-date facts remains a challenge even for large-scale LLMs. For the rather small-scale models used in our pipeline, context enrichment will help improve the model's predictions by adding relevant informations to the input prompts.

We will use in-context examples to provide references for the model, especially for relations

with variable outputs like awardWonBy or seriesHasNumberOfEpisodes.
Descriptive attributes and temporal markers can help disambiguate entities in time-sensitive relations. Furthermore, additional question-relevant information from Wikipedia is added to the end of the promt, thus adding context, increasing focus and (possibly) updating the data. An important issue is the correct adjustment of the retrieval system for difficult cases.

3.4 Wikidata Disambiguation

Large Language Models give answers in the form of names (of the people who won an award, of cities, stock exchanges, etc). The last step of our pipeline consists in converting these names into Wikidata IDs (except for the tv series episodes amount).

This is done by parsing names from the LLM output, and validating every name against the Wikidata Action API using the `wbsearchentities` action in order to obtain IDs. This is not trivial, as some names may have multiple corresponding Wikidata entries. Names that do not have any Wikidata match are ignored.

The naive method used in the baseline for retrieving Wikidata IDs consists in performing a Wikidata API call, and retrieving the first Wikidata ID in the returned list.

3.4.1 Previous Works

Previous works from challenge winners (like 2023 Track 1 winners) [2] have explored other options for Wikidata disambiguation:

- **case-based**: when the answer space is finite *(example: for a molecule, give all the constituent atoms. There are only 118 elements in the periodic table currently)*, a simple mapping from name to valid value is enough. However, the 2024 challenge does not feature such kind relation.
- **keyword-based**: for each returned Wikidata entry, fetch the Wikidata description, and match it against keywords for this specific relation *(example: for the `countryLandBordersCountry` relation, we expect relations to be countries, so they might have the word "country" in their description)*
- **lm-label**: build a dictionary using the returned Wikidata entry names and descriptions, concatenate everyone of them with the initial LLM query prompt, and ask a LLM to choose the most relevant one. This is by far the most powerful option, but the authors noted that it is **unstable** and **time-consuming**.

3.4.2 Our Contribution

For this challenge edition, each expected answer entity from the dataset (apart for `seriesHasNumberOfEpisodes` which are numbers) has a corresponding Wikidata property that ensures they are human, a city, or have won a specific award:

- `P31` : instanceof
- `P166` : awards received

Assuming that all **subject entities** and answer **object entities** are present in Wikidata, we can perform a strict disambiguation by filtering out object entity candidates that do not present the expected relation with the subject entity in Wikidata.

Relation	Required `instanceof` QID	Description
`awardWonBy`	`Q5`	Human
	`Q15632617`	Fictional human
`countryLandBordersCountry`	`Q6256`	Country
	`Q7275`	State
`personHasCityOfDeath`	`Q515`	City
	`Q1549591`	Big City

Table 1: Disambiguation based on the `P31:instanceof` Wikidata property. **At least one of the required types must be present** in order to accept a Wikidata entity as the disambiguated answer for the specified relation.

In the case of `awardWonBy`, we require valid entities to additionally possess the given award QID in their `P166:awards received` property. The award QID is fetched from Wikidata by name and disambiguated using the baseline method. If all entries are filtered out, we default to the baseline disambiguation method. This is enough to cover edge cases where the Wikidata database is incomplete or lacks the properties for the expected entity.

Note that advanced disambiguation was not necessary for `companyTradesAtStockExchange`, since the baseline disambiguation method already has 100% accuracy.

The disambiguation pipeline for an entity name follows these steps:

1. Search for the name using the Wikidata **wbsearchentities** API.
2. Fetch the data of every candidate returned in the search using the Wikidata **wbgetentities** API.
3. Return the first candidate in the search that matches the required conditions. If no entitiy matches the conditions, return the first one.

4 Experiments

4.1 Disambiguation Results

Since the disambiguation phase affects metrics obtained in further experiments, it is important to first review the experimental results of the disambiguation solution proposed in Methods section. Before computing the test results for disambiguation methods, we cleaned up approximately 10 entries from the challenge dataset. Duplicated entries due to bad name splitting were fused back together, and 2 absurd entries with no corresponding Wikidata entities were removed.

We report the following disambiguation accuracy for each relation when feeding, for each question, the `ObjectEntities` names to our disambiguation method, and checking if the returned Wikidata QID is included in `ObjectEntitiesID`. We assumed that no name was expected to correspond with multiple Wikidata QIDs.

Relation	Accuracy
`awardWonBy`	100%
`countryLandBordersCountry`	100%
`companyTradesAtStockExchange`	100%
`personHasCityOfDeath`	98.04%

4.2 Prompting experiments

This section experiments with Llama3-8b model for the task at hand without adding relevant additional context to prompts. Since related works [2], [3] and the analysis of the used models suggest that the model's own knowledge is insufficient for obtaining high-quality predictions, the goal of experiments with prompts is to obtain the correct format of predictions and predictability in complex cases.

It is known from NLP literature that prompt formulation plays a critical role in the performance of LMs on downstream applications. This is confirmed by the low quality of results and the absence of any output format when using the simplest prompt "Predict the list of object entities for the title-entity {title-entity} and relation {relation-name}". First of all, from examining a dataset with five relations with their own specificities we can conclude about the importance of custom templates for each sub-task. This customization increased the macro f1-score by 0.04 ppoints. To address output format questions first of all, a guideline to output a comma-separated list of values was added to the prompt. Another possible option is to output in json format, since LLM was trained on a large corpus of json files, but the prompt with a list of values showed reasonably good formatting results. The next step is the addition of well-known prompt techniques - instructions and examples (FSL). Which will raise the macro f1-score by another 0.1 ppoint. The different instruction wording and number of ppoints did not result in statistically significant changes in the metrics, possibly due to the short length of all instructions considered. Different numbers and orders of provided examples were also experimentally tested. No improvement in the metrics was observed when adding a large number of examples (more than 3). However, especially for the personHasCityOfDeath, countryLandBordersCountry, companyTradesAtStockExchange relations, the content of the examples is important. Thus, three examples were manually selected for each relation: list with multiple answers, one answer, Null. An attempt was made to consider the impact of the value used to indicate the absence of object entities. However, the requirement for the model to return “None”, “Null” and empty string “” had comparable metrics. However, these experiments need to be performed on a larger sample, using additional context and with different models, as it may have a considerable effect for models with a smaller context windows.

4.3 Context Enrichment experiments

This section presents experimental results of the considered models and two disambiguation methods when adding additional relevant context from Wikidata to the prompts described above.

Results for each model using **300 tokens** (limited by available GPU resources) and the **baseline disambiguation method** for speed:

	Relation	macro-p	macro-r	macro-f1	micro-p	micro-r	micro-f1	avg preds	empty preds
bert-large-cased	`awardWonBy`	0.400	0.000	0.000	0.000	0.000	0.000	0.600	4
	`companyTradesAtStockExchange`	0.090	0.350	0.040	0.000	0.000	0.000	**1.680**	9
	`countryLandBordersCountry`	0.029	0.279	0.029	0.042	0.017	0.024	1.059	0
	`personHasCityOfDeath`	0.222	**0.690**	0.207	0.148	**0.436**	0.221	**1.620**	4
	`seriesHasNumberOfEpisodes`	**0.865**	0.080	0.077	**0.348**	0.080	0.130	0.230	79
	All Relations	0.327	0.347	0.091	0.081	0.019	0.030	1.140	96
llama-3-8b-instruct	`awardWonBy`	0.586	**0.017**	**0.030**	**0.684**	**0.009**	**0.017**	**1.900**	4
	`companyTradesAtStockExchange`	**0.673**	**0.701**	**0.522**	**0.520**	**0.494**	**0.506**	0.750	30
	`countryLandBordersCountry`	**0.852**	**0.770**	**0.732**	**0.755**	**0.654**	**0.701**	2.279	22
	`personHasCityOfDeath`	**0.610**	0.680	**0.450**	**0.348**	0.418	**0.380**	0.660	38
	`seriesHasNumberOfEpisodes`	0.280	**0.280**	**0.280**	0.280	**0.280**	**0.280**	**1.000**	0
	All Relations	***0.582***	***0.578***	***0.464***	***0.530***	***0.116***	***0.191***	1.098	94
opt1.3b	`awardWonBy`	**0.612**	0.006	0.008	0.091	0.001	0.001	1.100	6
	`companyTradesAtStockExchange`	0.590	0.500	0.313	0.167	0.215	0.188	1.020	43
	`countryLandBordersCountry`	0.580	0.569	0.362	0.297	0.335	0.315	**2.971**	23
	`personHasCityOfDeath`	0.440	0.560	0.283	0.151	0.200	0.172	0.730	34
	`seriesHasNumberOfEpisodes`	0.160	0.080	0.080	0.086	0.080	0.083	0.930	8
	All Relations	0.435	0.404	0.244	0.202	0.051	0.082	***1.272***	114

Results for some model using **300 tokens** (limited by available GPU resources) and the **fancy disambiguation method** for speed:

	Relation	macro-p	macro-r	macro-f1	micro-p	micro-r	micro-f1	avg preds	empty preds
bert-large-cased	awardWonBy	0.500	0.000	0.000	0.167	0.001	0.001	0.600	4
	companyTradesAtStockExchange	0.080	0.350	0.040	0.000	0.000	0.000	**1.700**	8
	countryLandBordersCountry	0.029	0.279	0.029	0.042	0.017	0.024	1.059	0
	personHasCityOfDeath	0.222	0.690	0.207	0.148	0.436	0.221	**1.620**	4
	seriesHasNumberOfEpisodes	**0.960**	0.080	0.080	**0.667**	0.080	0.143	0.120	88
	All Relations	0.353	0.347	0.092	0.085	0.019	0.031	1.116	104
rag_fancy_llama_3_chat	awardWonBy	**0.599**	**0.063**	**0.099**	**0.525**	**0.036**	**0.067**	**10.100**	1
	companyTradesAtStockExchange	**0.492**	**0.658**	**0.332**	**0.398**	**0.468**	**0.430**	0.930	18
	countryLandBordersCountry	**0.721**	**0.945**	**0.705**	**0.868**	**0.955**	**0.910**	**2.897**	2
	personHasCityOfDeath	**0.250**	**0.700**	**0.250**	**0.250**	**0.455**	**0.323**	1.000	0
	seriesHasNumberOfEpisodes	0.730	**0.270**	**0.270**	0.500	**0.270**	**0.351**	**0.540**	46
	All Relations	***0.535***	***0.602***	***0.355***	***0.574***	***0.166***	***0.257***	***1.442***	67

We observed that, for certain relations, the use of RAG with the fancy disambiguation method led to improved performance compared to the RAG and baseline disambiguation. This suggests that the more advanced disambiguation techniques helped the model better identify the correct object entities. Additionally, we noticed a reduction in the number of empty predictions, indicating that the fancy disambiguation approach provides more accurate and complete predictions.

4.4 Finetuning Experiment

Results for a fine-tuned version of the model opt1.3b with the same conditions as the context enrichment experiments.

	Relation	macro-p	macro-r	macro-f1	micro-p	micro-r	micro-f1	avg preds	empty preds
opt1.3b finetuned	awardWonBy	0.300	0.006	0.011	0.111	0.001	0.001	0.900	2
	companyTradesAtStockExchange	0.351	0.653	0.295	0.246	0.443	0.317	1.420	7
	countryLandBordersCountry	0.406	0.783	0.401	0.325	0.654	0.434	5.294	0
	personHasCityOfDeath	0.170	0.640	0.177	0.165	0.345	0.224	1.150	0
	seriesHasNumberOfEpisodes	0.190	0.100	0.100	0.110	0.100	0.105	0.910	9
	All Relations	0.269	0.510	0.224	0.254	0.096	0.140	1.897	18

For some relations, our fine-tuned model performs much better than the original model. Especially on relations where the object entities are already well known by our model, without any outside information.

Here is a comparison of the performances on all relations :

Model	macro-p	macro-r	macro-f1	micro-p	micro-r	micro-f1	avg preds	empty preds
opt1.3b original	**0.435**	0.404	**0.244**	0.202	0.051	0.082	1.272	114

opt1.3b finetuned	0.269	**0.510**	0.224	**0.254**	**0.096**	**0.140**	**1.897**	**18**

We notice that our fine-tuned model is better on most benchmarks, with the biggest improvement on the number of empty answers which drops from 114 to only 18. This shows that the model respects much better the formatting and avoid hallucinations.

4.5 Knowledge alignment

The results of experiments demonstrate the importance of inserting additional relevant context into prompt templates to improve task metrics. However, the methods used to add Wikidata do not guarantee the relevance and consistency of the added context. Since the LLM has internal knowledge gained through pre-training, the following conflicts of information sources can be distinguished: knowledge of the LLM vs facts in added context, conflicts of several given facts in added context.

The following naive experiment was proposed to provide some qualitative demonstration of the importance of the form of context enrichment technique in case of information conflict with the internal knowledge of the model. For a relation, we select several data samples (5 cases for 2 relations were considered) on which the model confidently predicts the correct object entities based on its internal knowledge without using additional context. Then, for random object entities for this kind of relation, small context mentioning the title-entity and these wrong object-entities are automatically generated. This conflicting context is added to the original prompt.

In this example the correct object entity for 'Edison International' title-entity is New York Stock Exchange. The conflicting context mentions Tokyo Stock Exchange.

```
Provide a comma-separated list of all stock exchanges where shares of Edison International are traded. Include only the names, and if none, state "None". Follow the format of the provided examples: Example 1: "Provide a comma-separated list of all stock exchanges where shares of Apple Inc. are traded. Response: NASDAQ, Frankfurt Stock Exchange, Swiss Exchange." Example 2: "Provide a comma-separated list of all stock exchanges where shares of a private company are traded. Response: None." You should give only a list of answers without any additional text.
Utilize this additional information fetched through reliable information retrieval techniques to confirm the accuracy of the stock exchanges list: Edison International (TSE: EIX) is listed on the Tokyo Stock Exchange, offering investors access to one of the leading energy companies. With a strong focus on renewable energy and infrastructure development, the company aligns with Japan's push for sustainable solutions. Analysts anticipate increased market activity and investor interest following the listing.
```

When analyzing the results, three cases can be distinguished: the model did not add a fact from a conflictual context, the model added a fact to the original correct fact from internal knowledge, the model responded with an only incorrect fact. Our experiments show that the occurrence of one or the other case strongly depends on the form of including the external context in the prompt. Thus, for example, when using the form from the given example, the ratio between above cases was 30-30-40 (each case was launched multiple times with slightly different prompts), but when the role of context was strengthened by using the phrase "Important: base your answer on this provided context…". the ratio shifted to 10-10-80.

Researching the behavior of LLMs when dealing with conflicting facts is an important task, the solution of which can significantly improve the metrics for the Knowledge Base Construction problem . As mentioned above, in this section a highly naive experiment with limited number of samples has been presented, however, even it shows the importance of this issue.

Bibliography

[1] J. Wei *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[2] B. Zhang, I. Reklos, N. Jain, A. M. Peñuela, and E. Simperl, “Using Large Language Models for Knowledge Engineering (LLMKE): A Case Study on Wikidata.” [Online]. Available: https://arxiv.org/abs/2309.08491

[3] D. M. R. Bara, “Prompt engineering for tail prediction in domain-specific knowledge graph completion tasks,” 2024.