# LLM2KB: Constructing Knowledge Bases using instruction tuned context aware Large Language Models

Anmol Nayak, Hari Prasad Timmapathini

*ARiSE Labs at Bosch, Bangalore, India*

Abstract
The advent of Large Language Models (LLM) has revolutionized the field of natural language processing, enabling significant progress in various applications. One key area of interest is the construction of Knowledge Bases (KB) using these powerful models. Knowledge bases serve as repositories of structured information, facilitating information retrieval and inference tasks. Our paper proposes LLM2KB, a system for constructing knowledge bases using large language models, with a focus on the Llama 2 architecture and the Wikipedia dataset. We perform parameter efficient instruction tuning for Llama-2-13b-chat and StableBeluga-13B by training small injection models that have only ≈0.05 % of the parameters of the base models using the Low-Rank Adaptation (LoRA) technique. These injection models have been trained with prompts that are engineered to utilize Wikipedia page contexts of subject entities fetched using a Dense Passage Retrieval (DPR) algorithm, to answer relevant object entities for a given subject entity and relation. Our best performing model achieved an average F1 score of 0.6185 across 21 relations in the LM-KBC challenge held at the ISWC 2023 conference.

## 1. Introduction

The rapid advancements in natural language processing (NLP) have propelled the development of large language models, revolutionizing the way machines understand and generate human language. One of the pivotal applications of these sophisticated models lies in the construction of knowledge bases, which serve as repositories of structured information essential for a multitude of NLP tasks, including information retrieval, question answering, and knowledge inference.

Knowledge bases hold immense potential for enhancing machine understanding of the world, but constructing them manually is a laborious and time-consuming process. However, the emergence of large language models such as GPT-4 [1], Llama 2 [2] Stable Beluga 2 [3], has opened new possibilities for knowledge base construction. These models possess extensive linguistic knowledge and factual knowledge that can be leveraged for automated entity recognition, relation extraction, and knowledge representation.

One of the prominent sources for constructing knowledge bases is the vast repository of human-curated information available on Wikipedia. This publicly accessible dataset contains

*KBC-LM'23: Knowledge Base Construction from Pre-trained Language Models workshop at ISWC 2023*

Anmol.Nayak@in.bosch.com (A. Nayak); HariPrasad.Timmapathini@in.bosch.com (H. P. Timmapathini)

CEUR Workshop Proceedings (CEUR-WS.org)

[*]Both authors contributed equally to this work.

[+]LLM2KB code: https://github.com/anmoln94/Team_LLM2KB_LM-KBC-2023 an extensive wealth of knowledge on diverse topics, making it an ideal resource for building comprehensive knowledge bases. Integrating Wikipedia data into knowledge bases allows for a wider coverage and a well-rounded understanding of various domains.

Nevertheless, the success of constructing knowledge bases using large language models hinges on the ability to fine-tune these models effectively. Traditional fine-tuning methods often suffer from scalability issues and demand an excessive amount of computational resources. However, recent advancements in parameter-efficient fine-tuning techniques, like LoRA (Low-Rank Adaptation) [4], have showcased promising results in reducing the complexity and computational requirements while preserving model performance. By efficiently fine-tuning large language models, researchers can unlock their true potential in knowledge base construction and empower them to comprehend and generate valuable information.

In this paper, we describe our system developed for Track 2 of the LM-KBC challenge at ISWC 2023 [5], which focuses on using language models of any size for knowledge base construction. Given an input subject-entity (s) and relation (r), the system attempts to predict all the correct disambiguated object-entities. Our system performs parameter efficient instruction tuning of Llama-2-13b-chat and StableBeluga-13B using the LoRA technique while leveraging Wikipedia text as context which is retrieved using a Dense Passage Retrieval (DPR) model.

## 2. Related Work

The inquiry into the potential of language models (LM) in replacing or aiding the creation and curation of knowledge bases was initially posed by [6] and later explored by other researchers [7]. The LAMA dataset which probes relational knowledge in language models through masked language modeling tasks to complete cloze-style sentences was introduced in [6].

While the initial study utilized manually designed prompts to probe the language model, subsequent research has demonstrated the advantages of automatically learning prompts. Various methods have emerged to mine prompts from large text corpora and select the most effective ones [8, 9]. Additionally, prompts can be directly learned through back-propagation [10, 11], showcasing how learned prompts can enhance the performance on LAMA tasks.

The performance of probing language models can be significantly improved through various approaches, such as directly learning continuous embeddings for prompts [12, 13], fine-tuning the LM on the training data [14], or few-shot learning [15]. The authors demonstrate that combining few-shot examples with learned prompts achieves the best probing results.

While probing language models has been extensively studied in the NLP community, the idea of utilizing language models to support knowledge graph curation has not received adequate attention [7]. Some works have demonstrated the combination of language models with knowledge bases to complete query results effectively [16]. Others have investigated how language models can be employed to identify errors in knowledge graphs [17] or explored using language models to weigh KG triples from ConceptNet for measuring semantic similarity. [18] have showcased the utility of language models in entity typing by predicting entity classes using language model-based approaches.

Further, the LM-KBC challenge at ISWC 2022 [19] produced interesting submissions on using LM for KB construction. [20] developed a system that performed task-specific pre-training of

BERT, used prompt decomposition for generating candidate objects progressively, and employed adaptive thresholds for candidate selection. They utilized additional knowledge triples from Wikidata KB for BERT pre-training and experimented with cloze-style prompts, but found that masking nearby tokens of the object-entity did not improve performance. By mining prompts from Wikipedia and using an ensemble approach with averaged voting, they achieved final object-entity predictions. They also proposed sticky thresholds for candidate selection based on likelihood comparison.

[21] introduces the ProP system, which employs GPT-3 [22] under a few-shot setting for knowledge base (KB) construction. ProP uses various prompting techniques, including manual prompt creation and question-style prompts, to verify the accuracy of GPT-3 generated claims. They utilize context examples with specific properties, such as varying answer sets, subjects with empty answer sets, and question-answer pairs, to train the model effectively.

[23] utilized manual prompts, generated from three automated sources, and applied ensemble learning for final predictions. Descriptive information from Wikidata was used to create prompts through ”middle-word”, ”dependency-based” and ”paraphrasing-based” strategies. The BERT-large model was probed using these prompts, and the five most frequent and likely objects were selected from the ensemble. Before selecting the top-5 objects, the candidate list was post-processed by removing stopwords. The threshold for candidate selection was treated as a hyper-parameter, and its tuning was done separately for each relation on the train dataset.

[24] proposes manual prompts tailored for each relation to probe the BERT-large model, utilizing semantics and domain knowledge. They uniquely utilize word co-occurrences in context to design prompts and reason about the relationship between subjects and objects in questions to optimize prompt design for various relations, observing improvements through simple modifications like changing articles in the prompt for the ”plays-instrument” relation. [25] conducts experiments with different manual prompts and candidate selection thresholds for each relation during BERT model probing. They also investigate the impact of selecting a larger number of options in the object list (100, 150, 180, or 200) on overall performance. Additionally, they create an ensemble of their manually crafted prompts to further enhance the probing results.

## 3. The LM-KBC 2023 Challenge

### 3.1. Description

The LM-KBC 2023 challenge is centered around constructing disambiguated knowledge bases from language models based on given subjects and relations. Unlike existing probing benchmarks like LAMA [6], LM-KBC 2023 does not assume any specific relation cardinalities, allowing subjects to have zero, one, or multiple object-entities. Submissions are required to not only rank predicted surface strings but also materialize disambiguated entities in the output. The evaluation process involves calculating precision and recall for each test instance compared to the ground-truth values. For every relation, the macro-averaged precision, recall and f1-score are computed. The final ranking of participating systems is determined based on the average F1-score across all 21 relations. Formally, the task involves predicting all correct object-entities (o1, o2, ..., ok) using LM probing given the input subject-entity (s) and relation (r).

The challenge comes with two tracks:

- Track 1: A small-model track with low computational requirements (<1 billion parameters) and usage of context is not allowed.
- Track 2: An open track, where participants can use any LM of their choice and usage of context is allowed.

### 3.2. Dataset

The LM-KBC dataset comprises both training and validation sets, encompassing 21 diverse relations. A test set containing only subject entities and relations was released towards the end of the challenge. The train, validation and test set each comprise of 1940 records. Each relation covers distinct subject-entities, and for each subject-relation pair, a comprehensive list of ground truth object-entities is provided. Each row in the dataset files includes the subject-entity ID, subject-entity name, a list of all possible object-entity IDs, a list of all possible object-entities, and the corresponding relation. The entity IDs correspond to the Wikidata ID. To perform instruction fine-tuning on the models, we consume the provided dataset and generate training samples in 2 ways:

- Method 1: First, the Train set and Validation set are combined to produce a super set. With each record, we begin generating a separate instruction tuning dataset (see Section 4.2 for details on the Prompts) by slot filling Prompt 1 and 2. Thus, each record in the super set begins by producing 2 samples for the instruction tuning dataset. Further, for each ObjectEntity in the record we slot fill Prompt 3, which additionally generates as many new samples for the instruction tuning dataset as there are object entities. The generated instruction tuning dataset is shuffled, keeping 14310 samples for training and 1000 samples for validation.
- Method 2: This method also utilises Prompt 1,2 and 3 in the same way as method 1 however the instruction tuning training dataset is generated only using the Train set, while the entire Validation set is used to generate the instruction tuning validation dataset. 7666 samples were used for training and 7644 samples were used for validation.

## 4. LLM2KB System Description

### 4.1. Components

#### 4.1.1. Large Language Model

We have selected 'Llama-2-13b-chat' model from Meta AI and 'StableBeluga-13B' model from Stability AI as our base models, as they have been fine-tuned for instruction following, are open source and have achieved state-of-the-art performance on numerous benchmarks. Both the models are fine-tuned versions of the original Llama-2-13b base model and have 13 billion parameters. Llama-2-13b-chat is fine-tuned for chat instructions using Supervised Fine Tuning (SFT) and Reinforcement Learning with Human Feedback (RLHF). StableBeluga-13B is a Llama-2-13b model fine-tuned on an Orca [26] style instruction dataset.

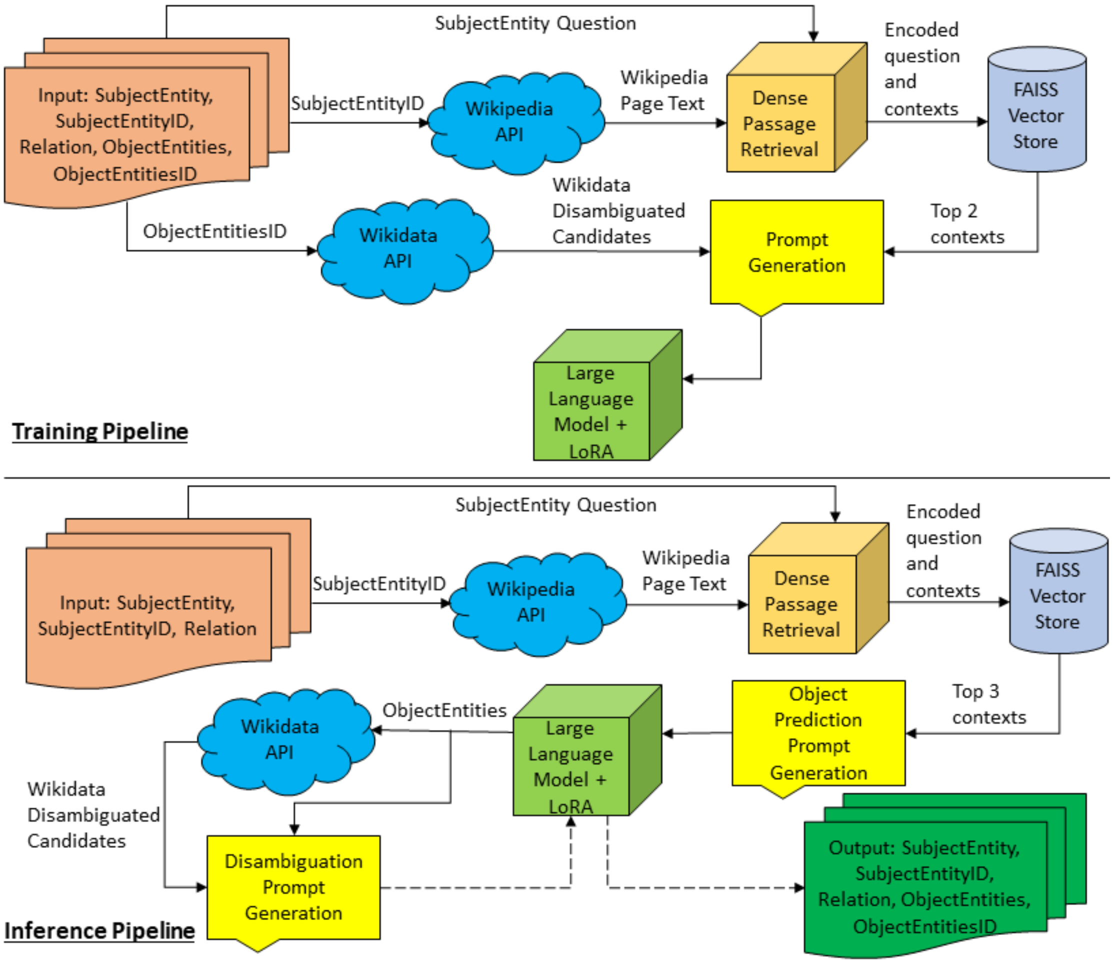
SubjectEntity Question
Input: SubjectEntity, SubjectEntityID, Relation, ObjectEntities, ObjectEntitiesID
SubjectEntityID
Wikipedia API
Wikipedia Page Text
Dense Passage Retrieval
Encoded question and contexts
FAISS Vector Store
Top 2 contexts
ObjectEntitiesID
Wikidata API
Wikidata Disambiguated Candidates
Prompt Generation
Large Language Model + LoRA
Training Pipeline
SubjectEntity Question
Input: SubjectEntity, SubjectEntityID, Relation
SubjectEntityID
Wikipedia API
Wikipedia Page Text
Dense Passage Retrieval
Encoded question and contexts
FAISS Vector Store
Top 3 contexts
Object Prediction Prompt Generation
Large Language Model + LoRA
ObjectEntities
Wikidata API
Wikidata Disambiguated Candidates
Disambiguation Prompt Generation
Output: SubjectEntity, SubjectEntityID, Relation, ObjectEntities, ObjectEntitiesID
Inference Pipeline

**Figure 1:** LLM2KB System Architecture.

### 4.1.2. Context Retrieval Model

We have selected Dense Passage Retrieval (DPR) from Meta AI [27] for subject entity context retrieval. DPR comprises a collection of tools and models used in state-of-the-art open-domain Question and Answer research. We used 'dpr-ctx_encoder-single-nq-base' for encoding questions and 'dpr-ctx_encoder-multiset-base' for encoding contexts. Further, we store the dense vectors of the contexts in a Facebook AI Similarity Search (FAISS) vector store for fast search and retrieval. We believe that augmenting prompts with a context helps the LLM in two ways:

1. Since the answers are factual in nature, the context can aid the LLM to generate answers with a higher degree of accuracy instead of just trying to recollect them from its stored knowledge.
2. The chance of a LLM hallucination decreases since it can use the context as a reference point while generating an answer.

#### 4.1.3. Entity Disambiguation

Once the LLM has predicted an object entity, we send the object entity text to the Wikidata API to fetch the potential candidates for the disambiguated entity. The candidates returned by Wikidata are then given to the LLM to pick the correct disambiguated entity. We use the LLM to perform disambiguation instead of just picking the top result returned by Wikidata API due to the following reasons:

1. Since the Wikidata API returns candidates by using the object entity surface string as a search key, it lacks the information of the subject entity and the relation to accurately rank the correct disambiguated entity as the top result. We even attempted passing the subject entity and relation along with the object entity surface string to the API, but the response was either empty or irrelevant.
2. In special cases where the predicted object entity surface string could be disambiguated in one or more ways, it is vital that the LLM is utilised to find the correct disambiguation. For e.g. if the question is ”Who is William Shakespeare married to?”, and the LLM correctly answers ”Anne Hathaway”, the Wikidata API will return entities of both Anne Hathaway (wife of William Shakespeare) and Anne Hathaway (American actress).
3. Even if we assume that the top result returned by Wikidata is correct, it is risky to rely on it since the API could in the future start returning results in a shuffled order.

### 4.2. LLM prompts

We created 4 different prompts based on the format of each of the base models, where Prompts 1,2 and 3 are used for performing instruction tuning and Prompt 4 is used only during inference for in-context learning. Note: During inference, Prompts 1,2 and 3 are trimmed after *[/INST]* for Llama-2-13b-chat and after *Answer:* for StableBeluga-13B. Instruction tuning is a technique to perform supervised fine tuning of models to make them learn to produce valid responses for specific instructions. In our case, the instruction tuning helps the model in 2 ways:

1. Learn to answer relevant object entities for a given question (see Table 1) with and without context. This is achieved with Prompt 1 and Prompt 2.
2. Given a predicted surface string of an object entity, learn to pick the correct Wikipedia entity title (but not Wikidata ID) from a list of candidate options. This is achieved with Prompt 3.

In-context learning is a method where few examples of the expected output are supplied in the prompt and the model uses them to produce an output in a similar fashion. Prompt 1,2 and 3 are expecting the LLM to produce an answer in a Python list of string format, however we noticed that the model sometimes produces a syntactically incorrect list. Hence, we use Prompt 4 to demonstrate to the LLM an incorrect format Answer vs. correct format Answer.

#### 4.2.1. Llama-2-13b-chat prompts

1. *<s>[INST] «SYS»*

*You are a helpful, respectful and honest assistant. Your answers should be crisp, short and not repititive.*
*Give valid wikipedia page titles in the answer. The answer should be in a python list of string format.*
*If you dont know the answer from both the given context and your past knowledge, answer should just be a python empty list.*
*«/SYS»*
*context: ’{context}’*
*{question} [/INST] Answer: {answer} </s>*

In this prompt, the {question} variable is formed by first picking the relevant question corresponding to the relation (see Table 1) and then replacing the subject entity of the data row in the {question}. {context} variable is formed by concatenating the strings of the top 2 contexts returned by the DPR system for the given {question}. It is important to note that to fetch the top 2 contexts, the DPR context encoder is only fed the Wikipedia page textual content of the subject entity and we do not use the content of the Wikipedia Infobox (found at the top right corner of a Wikipedia page), since the Infobox already contains semi-structured information about an entity which would defeat the purpose of using a LLM. The {answer} variable is formed by replacing the object entities of the data row.

2. *<s>[INST] «SYS»*
   *You are a helpful, respectful and honest assistant. Your answers should be crisp, short and not repititive.*
   *Give valid wikipedia page titles in the answer. The answer should be in a python list of string format.*
   *If you dont know the answer from both the given context and your past knowledge, answer should just be a python empty list.*
   *«/SYS»*
   *context: ”*
   *{question} [/INST] Answer: {answer} </s>*

   In this prompt, the {question} variable and {answer} variable are formed in the same method as the previous prompt, however we do not supply any context.
3. *<s>[INST] «SYS»*
   *You are a helpful, respectful and honest assistant. Your answers should be crisp, short and not repititive.*
   *Choose an answer from the options in the context.*
   *If you dont know the answer from the given context, answer should just be a python empty list.*
   *«/SYS»*
   *context: ’{options}’*
   *{question} [/INST] Answer: {answer} </s>*

In this prompt, the {question} variable is formed in the same method as the previous prompts, the {options} variable is a list of titles (appended with their descriptions) for the Wikidata entities returned by the API when queried with a given object entity text. The titles and descriptions are fetched from the API response. Each separate object entity will lead to 1 unique training sample. For e.g. if a data row has 4 object entities, then we will generate 4 separate training samples for it. The {answer} variable is the title of the response entity which has the same Wikidata ID as the object entity.

4. *<s>[INST] «SYS»*
   *Example 1: Wrong Format: ['People's Republic of China', 'Laos', 'Thailand', 'India', 'Bangladesh']"]. Correct Format: Answer: "People's Republic of China", "Laos", "Thailand", "India", "Bangladesh"] </s>*
   *Example 2: Wrong Format: ['Artibonite', 'Nord-Est Department', 'South Department','West Department', 'Centre Department', 'Grand'Anse Department', 'North Department']. Correct Format: Answer: "Artibonite", "Nord-Est Department", "South Department", "West Department", "Centre Department", "Grand'Anse Department", "North Department"] </s>*
   *Example 3: Wrong Format: ['book's and page's']. Correct Format: Answer: ["book's and page's"] </s>*
   *Your answer should only be a valid python list of string format. Do not give any explainations.*
   *«/SYS»*
   *Use the examples to convert {answer} into a correct python list. [/INST] Answer:*

   Once an answer is generated from Prompt 1, 2 or 3, it is used as the {answer} variable in this prompt to format the answer in a correct Python list of string format.

#### 4.2.2. StableBeluga-13B prompts

Please refer the Appendix A for the prompts used for StableBeluga-13B. The {question}, {answer}, {context} and {options} variables are generated in the same method as described for the Llama-2-13b-chat prompts.

### 4.3. Training

For both base models namely Llama-2-13b-chat and StableBeluga-13B, we load them in 4 bit quantized state with frozen weights and only train an injection model using LoRA technique. The injection model has ≈0.05 % of the parameters of the base models. The time taken for training was ≈9 hours for method 1 data and ≈5 hours for method 2 data. The training setup was as follows:

- **Libraries**: BitsandBytes [28, 29], HuggingFace [30], PyTorch [31]
- **Base model**: BitsandBytesConfig(load_in_4bit=True, bnb_4bit_use_double_quant=True, bnb_4bit_quant_type="nf4", bnb_4bit_compute_dtype=torch.bfloat16)
- **LoRA model**: alpha=16, dropout=0.05, r=4, bias="none", task_type="CAUSAL_LM"

**Table 1**
Questions corresponding to each relation.

| Relation | Question |
|---|---|
| BandHasMember | Who are the members of _? |
| CityLocatedAtRiver | Which river is _ located at? |
| CompanyHasParentOrganisation | What is the parent organization of _? |
| CompoundHasParts | What are the components of _? |
| CountryBordersCountry | Which countries border _? |
| CountryHasOfficialLanguage | What is the official language of _? |
| CountryHasStates | Which states are part of _? |
| FootballerPlaysPosition | What position does _ play in football? |
| PersonCauseOfDeath | What caused the death of _? |
| PersonHasAutobiography | What is the title of _'s autobiography? |
| PersonHasEmployer | Who is _'s employer? |
| PersonHasNoblePrize | In which field did _ receive the Nobel Prize? |
| PersonHasNumberOfChildren | How many children does _ have? |
| PersonHasPlaceOfDeath | Where did _ die? |
| PersonHasProfession | What is _'s profession? |
| PersonHasSpouse | Who is _ married to? |
| PersonPlaysInstrument | What instrument does _ play? |
| PersonSpeaksLanguage | What languages does _ speak? |
| RiverBasinsCountry | In which country can you find the _ river basin? |
| SeriesHasNumberOfEpisodes | How many episodes does the series _ have? |
| StateBordersState | Which states border the state of _? |

- **Trainer**: epochs=3, optimizer="paged_adamw_32bit", gradient_accumulation_steps=2, per_device_train_batch_size=1, per_device_eval_batch_size= 4, fp16=True, learning_rate=2e-5, max_grad_norm=0.3, warmup_ratio=0.03, lr_scheduler_type="constant", evaluation_strategy="epoch"
- **GPU**: 2x NVIDIA V100

### 4.4. Inference

The overall architecture can be seen in Fig. 1. During inference, we first try to fetch the English language Wikipedia page text of the subject entity using its Wikidata ID. If the subject entity does not have an English page, we then pick the text from the primary alternate language page. We then split the text into chunks of 300 tokens (with an overlap of 50 tokens to maintain continuity) due to the following reasons:

- LLM have a limit on the context length, which in the case of Llama-2-13b-chat and StableBeluga-13B is 4096 tokens. Since the Wikipedia text is usually large and may cross the maximum context length supported by the LLM, it is not feasible to consume the entire context in one shot.
- Due to the size of the LLM, passing lengthy inputs increases the inference time. Further, it makes much more sense to pick those chunks of information from the entire Wikipedia page that are relevant to a given question.

**Table 2**
LLM2KB results (Precision, Recall and F-1 score are macro average per relation) on Test set with Llama-2-13b-chat. Train + Validation set were used to generate training samples.

| Relation | Precision | Recall | F1 score |
|---|---|---|---|
| BandHasMember | 0.7178 | 0.4151 | 0.4857 |
| CityLocatedAtRiver | 0.7400 | 0.5205 | 0.5586 |
| CompanyHasParentOrganisation | 0.7200 | 0.7200 | 0.6400 |
| CompoundHasParts | 0.9318 | 0.8495 | 0.8784 |
| CountryBordersCountry | 0.8492 | 0.5848 | 0.6719 |
| CountryHasOfficialLanguage | 0.9154 | 0.7500 | 0.8015 |
| CountryHasStates | 0.5687 | 0.3872 | 0.4399 |
| FootballerPlaysPosition | 0.7050 | 0.6683 | 0.6783 |
| PersonCauseOfDeath | 0.8500 | 0.8200 | 0.8200 |
| PersonHasAutobiography | 0.6700 | 0.4075 | 0.4173 |
| PersonHasEmployer | 0.5100 | 0.2957 | 0.3327 |
| PersonHasNoblePrize | 0.9900 | 0.9350 | 0.9367 |
| PersonHasNumberOfChildren | 0.4900 | 0.4900 | 0.4900 |
| PersonHasPlaceOfDeath | 0.8300 | 0.7600 | 0.7200 |
| PersonHasProfession | 0.6600 | 0.4628 | 0.5034 |
| PersonHasSpouse | 0.7700 | 0.5850 | 0.5867 |
| PersonPlaysInstrument | 0.7633 | 0.6159 | 0.6354 |
| PersonSpeaksLanguage | 0.8500 | 0.7538 | 0.7704 |
| RiverBasinsCountry | 0.8883 | 0.7689 | 0.7896 |
| SeriesHasNumberOfEpisodes | 0.4500 | 0.4500 | 0.4500 |
| StateBordersState | 0.5090 | 0.3416 | 0.3815 |
| **Average** | **0.7323** | **0.5991** | **0.6185** |

Each of the context chunks are encoded using the DPR context encoder and stored in a FAISS vector store for fast search and retrieval. To ensure that the LLM is not just fixated on using only top 2 contexts, during inference we pick top 3 contexts to test its robustness towards handling variability. To pick the top 3 relevant context chunks for a given question, the question is encoded using the DPR question encoder and then passed to FAISS. The top 3 retrieved context chunks are concatenated to form the {context} variable for the prompts. In cases where a subject entity does not have a Wikipedia page, the {context} variable will be empty for the prompt, and the LLM will have to rely on its stored knowledge to answer the question. Prompt 1 is executed if a context was found whereas Prompt 2 is executed if no context was found.

Once the LLM processes the prompt, the answer can have 0 or more object entities. For disambiguating each of these surface strings, we query the Wikidata API with each of these object entities separately. The API will return a list of candidate Wikidata entities, from which we attempt to find the correct disambiguated entity. To do this, we collect all the candidate entities and put them in a list to form the {context} in Prompt 3. The expectation is that the LLM picks the correct disambiguated entity relevant to the question. The Wikidata ID of each disambiguated object entity is then fetched and stored. In cases where the LLM generated a string which was not an exact match of the provided options, we pass the LLM output to Wikipedia API for fetching the most relevant entity.

## 5. Results and Future Work

The relation level performance of our system can be seen in Tables 2, 3 and 4 (Tables 3 and 4 are in the Appendix B). Our primary experiments involved utilising training samples generated using method 1 for Llama-2-13b-chat and StableBeluga-13B. Additionally, we also trained a Llama-2-13b-chat model with method 2 used to generate training samples. As described in Section 3.2, method 1 utilized samples from both the training set and validation set as part of the instruction tuning training data, whereas method 2 utilized samples only from the training set for the instruction tuning training data. Our best performing model was Llama-2-13b-chat when trained on instruction samples generated using method 1. In training with both method 1 and method 2 data generation strategies, Llama-2-13b-chat performed better than StableBeluga-13B. We found that while supplying additional training samples using method 1 gave a better overall F1-score for Llama-2-13b-chat (0.6185), it was not significantly higher than when it was instruction tuned with method 2 data (0.6016).

'PersonHasNoblePrize' is the highest scoring relation from our best model and 'PersonHasEmployer' is the lowest scoring relation from our best model. All the 3 models did not perform very highly on the 2 relations which only expected numeric answers namely 'PersonHasNumberOfChildren' and 'SeriesHasNumberOfEpisodes' even after supplying context. After going through the Wikipedia contexts for the subject entities of these relations, we observed that many of them never mention anything about number of children or number of episodes, leaving the LLM to solely rely on its memory of factual knowledge. Overall we observed that the following are some of the practical challenges after subjective analysis of the results:

- Fragility of LLM during inference towards minor changes in prompts.
- Hallucinations of LLM.
- LLM produces a correct surface string of an object but Wikidata API is unable to return any relevant entities for it.

Our future work will focus on utilising 30 billion and 70 billion versions of the LLM to see if that can push the performance further and also experiment with Chain-of-Thought prompt techniques.

## References

[1] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[2] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288.

[3] Stability ai stable beluga, 2023. URL: https://stability.ai/blog/stable-beluga-large-instruction-fine-tuned-models.

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[5] S. Singhania, J.-C. Kalo, S. Razniewski, J. Z. Pan, Lm-kbc: Knowledge base construction from pre-trainedlanguage models, semantic web challenge @ iswc, CEUR-WS (2023). URL: https://lm-kbc.github.io/challenge2023/.

[6] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2463–2473.

[7] S. Razniewskia, A. Yatesa, N. Kassnerc, G. Weikuma, Language models as or for knowledge bases (2021).

[8] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know?, Transactions of the Association for Computational Linguistics 8 (2020) 423–438.

[9] Z. Bouraoui, J. Camacho-Collados, S. Schockaert, Inducing relational knowledge from bert, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7456–7463.

[10] A. Haviv, J. Berant, A. Globerson, Bertese: Learning to speak to bert, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 3618–3623.

[11] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4222–4235.

[12] G. Qin, J. Eisner, Learning how to ask: Querying lms with mixtures of soft prompts, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 5203–5212.

[13] Z. Zhong, D. Friedman, D. Chen, Factual probing is [mask]: Learning vs. learning to recall, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 5017–5033.

[14] L. Fichtel, J.-C. Kalo, W.-T. Balke, Prompt tuning or fine-tuning-investigating relational knowledge in pre-trained language models, in: 3rd Conference on Automated Knowledge Base Construction, 2021.

[15] T. He, K. Cho, J. Glass, An empirical study on few-shot knowledge probing for pretrained language models, arXiv preprint arXiv:2109.02772 (2021).

[16] J.-C. Kalo, L. Fichtel, P. Ehler, W.-T. Balke, Knowlybert-hybrid query answering over language models and knowledge graphs, in: International Semantic Web Conference, 2020, pp. 294–310.

[17] H. Arnaout, T.-K. Tran, D. Stepanova, M. H. Gad-Elrab, S. Razniewski, G. Weikum, Utilizing language model probes for knowledge graph repair, in: Wiki Workshop 2022, 2022.

[18] R. Biswas, R. Sofronova, M. Alam, N. Heist, Do judge an entity by its name! entity typing using language models, The Semantic Web: ESWC 2021 Satellite Events (2021) 65.

[19] S. Singhania, T.-P. Nguyen, S. Razniewski, Lm-kbc: Knowledge base construction from

pre-trained language models, CEUR-WS (2022).

[20] T. Li, W. Huang, N. Papasarantopoulos, P. Vougiouklis, J. Z. Pan, Task-specific pre-training and prompt decomposition for knowledge graph population with language models, CEUR-WS (2022).

[21] D. Alivanistos, S. B. Santamaría, M. Cochez, J.-C. Kalo, E. van Krieken, T. Thanapalasingam, Prompting as probing: Using language models for knowledge base construction, CEUR-WS (2022).

[22] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[23] X. Ning, R. Celebi, Knowledge base construction from pre-trained language models by prompt learning, CEUR-WS (2022).

[24] X. Fang, A. Kalinowski, H. Zhao, Z. You, Y. Zhang, Y. An, Prompt design and answer processing for knowledge base construction from pre-trained language models (kbc-lm), CEUR-WS (2022).

[25] S. Dalal, A. Sharma, S. Jain, M. Dave, Manual prompt generation for language model probing (2022).

[26] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, A. Awadallah, Orca: Progressive learning from complex explanation traces of gpt-4, arXiv preprint arXiv:2306.02707 (2023).

[27] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781.

[28] T. Dettmers, M. Lewis, Y. Belkada, L. Zettlemoyer, Llm. int8 (): 8-bit matrix multiplication for transformers at scale, arXiv preprint arXiv:2208.07339 (2022).

[29] T. Dettmers, M. Lewis, S. Shleifer, L. Zettlemoyer, 8-bit optimizers via block-wise quantization, 9th International Conference on Learning Representations, ICLR (2022).

[30] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

# A. StableBeluga-13B Prompts

1. *### System:*
   *You are a helpful, respectful and honest assistant. Your answers should be crisp, short and not repititive.*

*Give valid wikipedia page titles in the answer. The answer should be in a python list of string format.*
*If you dont know the answer from both the given context and your past knowledge, answer should just be a python empty list.*
*### User:*
*context: {context}*
*{question}*
*### Assistant*
*Answer: {answer}*

2. *### System:*
   *You are a helpful, respectful and honest assistant. Your answers should be crisp, short and not repititive.*
   *Give valid wikipedia page titles in the answer. The answer should be in a python list of string format.*
   *If you dont know the answer from both the given context and your past knowledge, answer should just be a python empty list.*
   *### User:*
   *context: ”*
   *{question}*
   *### Assistant*
   *Answer: {answer}*
3. *### System:*
   *You are a helpful, respectful and honest assistant. Your answers should be crisp, short and not repititive.*
   *Choose an answer from the options in the context.*
   *If you dont know the answer from both the given context, answer should just be a python empty list.*
   *### User:*
   *context: {options}*
   *{question}*
   *### Assistant*
   *Answer: [’{answer}’]*
4. *### System:*
   *Example 1: Wrong Format: [’People’s Republic of China’, ’Laos’, ’Thailand’, ’India’, ’Bangla desh’]”]. Correct Format: Answer: ”People’s Republic of China”, ”Laos”, ”Thailand”, ”India”, ”Bangladesh”] </s>*
   *Example 2: Wrong Format: [’Artibonite’, ’Nord-Est Department’, ’South Department’, ’West Department’, ’Centre Department’, ’Grand’Anse Department’, ’North Department’]. Correct Format: Answer: ”Artibonite”, ”Nord-Est Department”, ”South Department”, ”West Department”, ”Centre Department”, ”Grand’Anse Department”, ”North Department”] </s>*
   *Example 3: Wrong Format: [’book’s and page’s’]. Correct Format: Answer: [”book’s and page’s”] </s>*
   *### User:*
   *Your answer should only be a valid python list of string format. Do not give any explainations.*

**Table 3**
LLM2KB results (Precision, Recall and F-1 score are macro average per relation) on Test set with Llama-2-13b-chat. Only Train set was used to generate training samples.

| Relation | Precision | Recall | F1 score |
|---|---|---|---|
| BandHasMember | 0.7235 | 0.4132 | 0.4822 |
| CityLocatedAtRiver | 0.7200 | 0.4948 | 0.5291 |
| CompanyHasParentOrganisation | 0.7400 | 0.7550 | 0.6667 |
| CompoundHasParts | 0.9280 | 0.7909 | 0.8442 |
| CountryBordersCountry | 0.8439 | 0.5501 | 0.6442 |
| CountryHasOfficialLanguage | 0.9000 | 0.7372 | 0.7877 |
| CountryHasStates | 0.5602 | 0.3669 | 0.4221 |
| FootballerPlaysPosition | 0.7450 | 0.4900 | 0.4917 |
| PersonCauseOfDeath | 0.8600 | 0.8133 | 0.8150 |
| PersonHasAutobiography | 0.6900 | 0.4125 | 0.4240 |
| PersonHasEmployer | 0.4100 | 0.2237 | 0.2555 |
| PersonHasNoblePrize | 0.9700 | 0.9550 | 0.9567 |
| PersonHasNumberOfChildren | 0.4300 | 0.4300 | 0.4300 |
| PersonHasPlaceOfDeath | 0.8600 | 0.8100 | 0.7800 |
| PersonHasProfession | 0.6050 | 0.4560 | 0.4856 |
| PersonHasSpouse | 0.7600 | 0.5650 | 0.5667 |
| PersonPlaysInstrument | 0.7100 | 0.5982 | 0.6133 |
| PersonSpeaksLanguage | 0.8650 | 0.7327 | 0.7607 |
| RiverBasinsCountry | 0.8925 | 0.7851 | 0.7997 |
| SeriesHasNumberOfEpisodes | 0.4800 | 0.4800 | 0.4800 |
| StateBordersState | 0.5977 | 0.3352 | 0.3983 |
| **Average** | **0.7281** | **0.5807** | **0.6016** |

*Use the examples to convert {answer} into a correct python list.*
*### Assistant*
*Answer:*

## B. Additional Results

Tables 3 and 4 contain additional experimental results for Llama- 2-13b-chat and StableBeluga-13B.

**Table 4**
LLM2KB results (Precision, Recall and F-1 score are macro average per relation) on Test set with StableBeluga-13B. Train + Validation set were used to generate training samples.

| **Relation** | **Precision** | **Recall** | **F1 score** |
|---|---|---|---|
| BandHasMember | 0.7075 | 0.4227 | 0.4850 |
| CityLocatedAtRiver | 0.7300 | 0.4865 | 0.5225 |
| CompanyHasParentOrganisation | 0.8800 | 0.6500 | 0.6300 |
| CompoundHasParts | 0.8573 | 0.6581 | 0.7197 |
| CountryBordersCountry | 0.8360 | 0.4690 | 0.5654 |
| CountryHasOfficialLanguage | 0.9462 | 0.7372 | 0.8021 |
| CountryHasStates | 0.7984 | 0.3225 | 0.3853 |
| FootballerPlaysPosition | 0.7100 | 0.4933 | 0.5050 |
| PersonCauseOfDeath | 0.8800 | 0.8000 | 0.8000 |
| PersonHasAutobiography | 0.8100 | 0.3450 | 0.3533 |
| PersonHasEmployer | 0.6700 | 0.2128 | 0.2403 |
| PersonHasNoblePrize | 0.9800 | 0.7700 | 0.7700 |
| PersonHasNumberOfChildren | 0.4400 | 0.4400 | 0.4400 |
| PersonHasPlaceOfDeath | 0.9400 | 0.6700 | 0.6700 |
| PersonHasProfession | 0.6100 | 0.3962 | 0.4457 |
| PersonHasSpouse | 0.8500 | 0.4450 | 0.4467 |
| PersonPlaysInstrument | 0.7667 | 0.5577 | 0.6055 |
| PersonSpeaksLanguage | 0.8850 | 0.6190 | 0.6947 |
| RiverBasinsCountry | 0.9170 | 0.7684 | 0.8030 |
| SeriesHasNumberOfEpisodes | 0.4200 | 0.4200 | 0.4200 |
| StateBordersState | 0.6817 | 0.2269 | 0.2872 |
| **Average** | **0.7769** | **0.5195** | **0.5520** |