

Towards Tailored Knowledge Base Modeling using Masked Language Models

Riley Capshaw^{1,*}, Eva Blomqvist¹

¹Linköping University, Linköping, Sweden

Abstract

We propose a methodology for leveraging aspects of ontology design principles to guide the use of a masked language model (MLM) as a query engine over raw text documents. By using targeted fill-in-the-blank-style prompts to define relations, we show how a domain expert could use BERT, a well-known MLM, to extract triples from unseen documents without any fine-tuning. We evaluate our proposed methodology using a modified document-level relation extraction task, highlighting early successes but also numerous areas that need improvement. Despite these shortcomings, we then discuss why we are still hopeful that this paves the way toward flexible text-based query engines which use collections of unstructured documents.

Keywords

Knowledge Graphs, Masked Language Models, Ontologies, Document-level Relation Extraction

1. Introduction

A Knowledge Graph (KG) is a labeled, directed graph where the nodes are entities of interest and all edges represent the existence of a typed relation between two of those entities. A KG can also be viewed as a set of subject-predicate-object triples representing the edges of the graph. KGs are often used for information retrieval and storage, and can be incorporated into larger systems for tasks like recommendation and question answering. In these settings, KGs are often paired with an ontology to provide added semantics to the entities and relations, thus making up a knowledge base (KB) that can be queried and reasoned over. This ontology provides basic vocabulary to the KG while also specifying the logical constraints that should hold. However, in many cases the actual knowledge underlying the KG is not structured data, but rather is found in unstructured or semi-structured forms like text documents. The larger the KG grows, the harder it is to guarantee its coherency and consistency with regards to both the ontology and the underlying knowledge sources that it represents. This also presents a challenge when needing to update the knowledge base, as adding or removing assertions may affect or even invalidate others. Even more complex is the task of changing the ontology, e.g to represent more or less the same information but with a different ontology design pattern, such as a different level of detail or granularity, after the KG has been developed or extracted from text.


Text2KG 2023: Second International Workshop on Knowledge Graph Generation from Text, Co-located with ESWC 2023, May 28–June 1 2023, Hersonissos, Greece

*Corresponding author.

✉ riley.capshaw@liu.se (R. Capshaw); eva.blomqvist@liu.se (E. Blomqvist)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

At the same time, advances in language modeling have yielded results which can provide (an approximation of) the same services as the KB envisioned above with no logical modeling, i.e. no actual KG or ontology. Some work even considers large language models (LLMs) as knowledge bases in their own right [1, 2, 3], with aspects like the structured data and requirements being implicit in the learned patterns and distributions. For instance, consider the recently released GPT3 [4], which can, among other things, answer arbitrary queries concerning factual statements. While impressive, these generative systems often rely on clever prompt engineering and may still assert false claims, so called “hallucinations,” requiring significant work on the part of the user to ensure the quality of the generated text. Further, because these systems are intended to be as general-purpose as possible, there is no obvious way to force them to only generate text supported by a given document or corpus when using them out-of-the-box. This means that potentially irrelevant or erroneous information present in the language model’s training data could bias its output, and hampers their use as KBs.

However, LLMs can still provide a natural language interface to knowledge, and this is what we attempt to exploit in this work. Our goal is not to automatically generate a KG from raw text, nor to use LLMs as KBs. Instead, we wish to treat text corpora as semi-structured KGs, i.e. replacing the extraction step with a virtual querying using an LLM, and enable users to explore the information within a text corpora through natural-language queries in a guided setting, e.g. using a specific ontology, as if knowledge would have been already represented in a KB.

Another aspect of this work is the opportunity to change the ontology as we go along. By incrementally defining the relations as text queries with logical restrictions, users simultaneously build both a query and the ontology backing it. Hence, we also address the problem of viewing the data through the lens of different ontologies or ontology patterns.

1.1. Motivating example

Take for example the document in Table 1, which is a part of the DocRED dataset [5] presented later on in this paper, where texts are annotated with entities and relations originating in WikiData [6]. Depending on the underlying ontology, a user might wish to extract the statement “Skai TV is located in the country of Greece.”. In a traditional KG, these facts are likely stored as triples such that a simple SPARQL query could be used to retrieve them using a basic graph pattern: $?x \text{ p17 } ?y$. In this case, the queries are essentially independent of the ontology; if p17 were undefined, there would simply be no matching triples returned. Further, the semantics of the predicate p17 are disjoint from its use. To understand $\text{Skai_TV p17 Greece}$, requires looking into the ontology directly. Had we named the predicate `country` instead, we risk making wrong assumptions about the actual semantics of the predicate and when it applies.

Now consider the case where that same query is written in a way which more closely matches the statement from before: “?x is located in the country of ?y.” This query makes a few more aspects of the relation clear, such as the direction of the relationship, the fact that the relationship is a spatial one, and that ?y needs to be a country. These aspects carry over directly to the results that the user gets. A user would also be able to see more clearly when incorrect statements are retrieved. If the object of a statement ended up not being a country, the user could add in a range constraint and try again. Finally, assume that the query is resolved by somehow matching it to information in text documents. Then, if the results just simply seem incorrect overall, the

Document	Relations
Skai TV is a Greek free-to-air television network based in Piraeus. It is part of the Skai Group, one of the largest media groups in the country. It was relaunched in its present form on 1st of April 2006 in the Athens metropolitan area, and gradually spread its coverage nationwide. Besides digital terrestrial transmission, it is available on the subscription-based encrypted services of Nova and Cosmote TV. Skai TV is also a member of Digea, a consortium of private television networks introducing digital terrestrial transmission in Greece. At launch, Skai TV opted for dubbing all foreign language content into Greek, instead of using subtitles. This is very uncommon in Greece for anything except documentaries (using voiceover dubbing) and children’s programmes (using lip-synced dubbing), so after intense criticism the station switched to using subtitles for almost all foreign shows.	(‘Piraeus’, ‘P17’, ‘Greece’) (‘Skai Group’, ‘P17’, ‘Greece’) (‘Athens’, ‘P17’, ‘Greece’) (‘Skai TV’, ‘P159’, ‘Piraeus’) (‘Skai TV’, ‘P127’, ‘Skai Group’) (‘Skai TV’, ‘P159’, ‘Athens’) (‘Skai TV’, ‘P17’, ‘Greece’)

Table 1

Document 3 from the DocRED data set, along with all the relations that should be extracted from it.

user could adjust the text of the query until the results were satisfactory. This encourages a guided approach to querying for information within the document, in a sense building up the logical structure alongside the query.

The system which performs the querying could additionally keep track of the underlying predicate it relates to, as well as any other information that enhances the query, such as constraints like domain and range. This would also allow for multiple different queries for the same relation, each representing different aspects, potentially with their own constraints. Relation P131 (“located in the administrative territorial entity”) is a particularly difficult one, with WikiData listing 47 alternate names¹. For that relation, it feels obvious that no single text query could describe all instances.

1.2. Research Questions

The work in this paper is a starting point towards answering the following research questions:

- RQ1: What are the limitations and challenges of using a masked language model to retrieve facts from natural-language text using fill-in-the-blanks queries and a perplexity-based scoring metric?
- RQ2: How robust is such a system to small changes in prompts and restrictions provided, and where are important points of future improvement?

While we cannot definitively answer these questions yet, our results are analyzed in Section 4 with these in mind, and they guide our discussion of future work in Section 5. Delimitations of the work include that we specifically analyze the case where the model is BERT [7]. Other details of the experimental setup are provided in Sections 3.

1.3. Contributions and Paper Outline

The main contributions of this work are (1) a method and technical setup for using a masked language model (MLM), i.e. BERT, in order to answer fill-in-the-blanks questions representing

¹See <https://www.wikidata.org/wiki/Property:P131> under “Also known as.” Accessed March 15, 2023.

typical KB triples, based on a specific text, and (2) a set of experimental results, allowing to analyse and explore the limitations and challenges of using BERT for this task, and derive future work directions.

The remainder of this paper is structured as follows. In Section 2 we briefly present some areas of related work, before presenting our method and technical setup in Section 3. Results of the experiments are then presented in Section 4. Finally, we conclude with a discussion and outlining future work in Section 5.

2. Related Work

There are approaches that try to use LMs as KBs in their own right, as exemplified by [1], [2] and [3]. However, although using an LM as if it was a KB, these approaches assume that the LM already contains the facts the user is asking for, while we make a different assumption, i.e. that the facts reside in some text corpora, rather than in the LM itself. Nevertheless, the idea of using natural language to access a KB and having the query interpreted by a LM is similar.

For automatically extracting a KG from text, on the other hand, the most prominent methods currently apply to Open Information Extraction (OpenIE) data sets. Although quite effective in many cases, OpenIE attempts to extract triples directly from text without any predetermined schema or ontology, which has several problems, e.g. as discussed in [8]. For instance this means that the resulting KG will have a diverse set of properties linking the entities, and no obvious way for the user to understand their overlap and relation, in order to formulate good queries over the data. Hence, using these techniques is not really suitable when the end-user needs a uniform way to access the resulting KG, e.g. through an ontology.

More in detail, the restricted part of the problem we focus on in the experiments in this paper is related to document-level relation extraction. In this area many approaches have emerged in the last few years, e.g. [9]. However, it is not possible to directly compare our results to such systems due to the different scoring system used in our method, pertaining to the different aim of our approach.

3. Method

In this section we describe the overall method and experimental setup used to explore the research questions. A diagram outlining this process is shown in Figure 1. First of all we describe the dataset, and how we generated prompts and other input to the experiments. Next we describe the LM used, the different experiments undertaken, and finally the evaluation metrics used for assessing the results.

3.1. Dataset & Queries

We chose to use the Document-level Relation Extraction Data set (DocRED) [5] for our experiments. This data set differs from OpenIE in several ways. First, it is a slightly easier, more restricted setting since all relations, entities, and mentions are provided a priori, so preprocessing steps are not needed and noise from those steps is minimized. However, this setting also

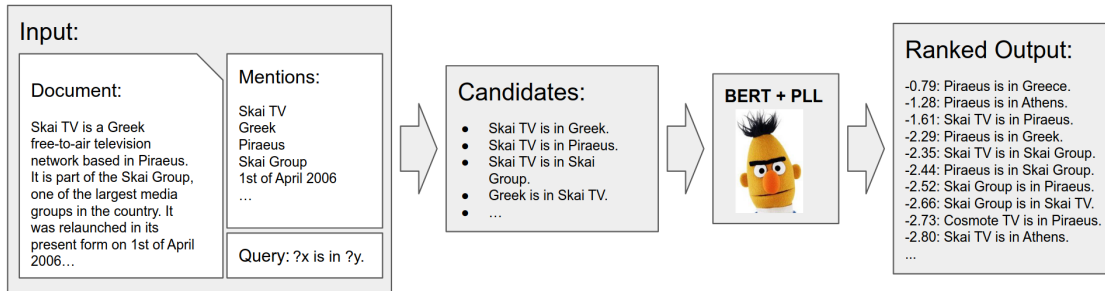


Figure 1: A pipeline showing the basic flow of information through the system. For any given document, all mentions are extracted and used along with the query to generate candidate statements. BERT then ranks these statements according to their PLL score, then returns an ordered list by score. Note that scores are always negative and higher (closer to 0) is better.

matches our intended task very well, since we are assuming that we already have an ontology based on which we would like to query our virtual KG. Second, the focus of the data set is on facts whose support spans multiple sentences, thus requiring entire documents to be considered in order for all facts to be extracted. The latter makes this an excellent data set for experimenting with fact extraction due to the need for sub-tasks like coreference resolution to be performed across sentence boundaries. This ensures that our results are relevant for realistic scenarios of using a text corpus to back the virtual KG.

3.1.1. Prompts

We complemented the DocRED dataset with custom prompts for every relation, as a representation of typical queries that could be submitted to the virtual KG. At least one prompt was written for each relation, with the ten most frequent relations receiving four prompts, each written by a different person (where 3 of those persons had no prior knowledge of the dataset, to ensure a realistic set of queries), to examine sensitivity to prompt variation. The prompts are in English, with the expected subject and object replaced with variable markers. For instance, the most frequent relation P17 (“country”) had the following prompts written for it:

1. *?x is in the country of ?y.
2. ?x is located in ?y.
3. ?x is located in the country of ?y.
4. ?x is in country ?y.

The prompt marked with * is used in the primary experiments in Section 3.3, while the other three are used in the sensitivity experiment described in Section 4.2.1. Table 2 gives details about the ten most frequent relations in DocRED, including the main prompts used in the experiments.

3.1.2. Domain and Range Restrictions

In order to get an idea of the effects of including logical restrictions in the prediction process, we included domain and range restrictions on the relations in one of our experiments. Such

Relation	Name	Description	Prompt
P17	country	sovereign state that this item is in (not to be used for human beings)	?x is in the country of ?y.
P27	country of citizenship	the object is a country that recognizes the subject as its citizen	?x is a citizen of ?y.
P131	located in the administrative territorial entity	the item is located on the territory of the following administrative entity	?x is located in ?y.
P150	contains administrative territorial entity	(list of) direct subdivisions of an administrative territorial entity	?x contains ?y within its borders.
P161	cast member	actor in the subject production	The cast of ?x includes ?y.
P175	performer	actor, musician, band or other performer associated with this role or musical work	?x was performed by ?y.
P527	has part	part of this subject	One part of ?x is ?y.
P569	date of birth	date on which the subject was born	?x was born on ?y.
P570	date of death	date on which the subject died	?x died on ?y.
P577	publication date	date or point in time when a work was first published or released	?x was published on ?y.

Table 2

Explanations of the ten most common relations from the data set, including the prompts.

restrictions would typically be found in an ontology, used for querying the virtual KG. For example, the “country” relation clearly can only have a country as the object, so its range was set to LOC, i.e. only location entities during this specific experiment. These restrictions were mined from the training portion of the DocRED data set. To account for labeling noise, we excluded types if they did not appear in at least 5% of the relation instances.

3.2. Language Model

For our baseline, we use the large, cased variant of the Bidirectional Encoder Representations from Transformers (BERT) model [7]. BERT is a pre-trained masked language model (MLM) which, when originally published, obtained state-of-the-art scores on eleven benchmark tasks through fairly minimal task-specific fine tuning. As a MLM, BERT was trained on an infilling task, where tokens in a sequence are masked out and need to be recovered. It was also trained on a next-sentence prediction task, but we do not focus on that here. To accomplish this, BERT learns contextualized vector representations for every token in the corpus that encapsulate enough information such that nearby masked words can be recovered reliably.

3.2.1. Token-Level Conditional Probabilities

Despite MLMs not being probabilistic models in a formal sense, they can produce a score analogous to conditional probabilities for every token in a sequence [10]. Given a token w_t at position t in a sequence \mathbf{W} , the score for w_t is the value

$$S_t(\mathbf{W}) := P_{\text{BERT}}(w_t | \mathbf{W}_{\setminus t}). \quad (1)$$

For ease of discussion, we adopt the language of related work and use conditional probability notation for these scores. The value of S_t is calculated by masking out token t to yield a new sequence $\mathbf{W}_{\setminus t}$, then using BERT to score replacing the masked-out token with the original token w_t . Consider the string “Hello world!” as an example. The score for “world” is calculated (after tokenization) as

$$S_1(['hello', 'world', '!']) = P_{\text{BERT}}('world' | ['hello', [\text{MASK}], '!']) = 0.344.$$

3.2.2. Pseudo-Log-Likelihood Sequence Ranking

Since BERT is a masked language model, it does not generate likelihood values for a given text, making it difficult to use for generative tasks. To circumvent this limitation, Wang and Cho [11] use BERT to calculate pseudo-log-likelihood scores (PLL):

$$\text{PLL}(\mathbf{W}) := \sum_{t=0}^{|\mathbf{W}|} \log S_t(\mathbf{W}) = \sum_{t=0}^{|\mathbf{W}|} \log P_{\text{BERT}}(w_t | \mathbf{W}_{\setminus t}) \quad (2)$$

With the same example as before, we can calculate the PLL of “Hello world!”:

$$\text{PLL}(['hello', 'world', '!']) = \log(0.430) + \log(0.344) + \log(0.818) = -0.704.$$

3.3. Experimental Setup

The basic experiments are performed as follows. The DocRED dataset already provides certain information which would otherwise need to be extracted separately, so we avoid any further preprocessing of the text. In this sense, we are working under the assumption that similar data could be generated using a pipeline that includes typed named-entity recognition, entity-mention disambiguation, and identification of all possible relations. Then, one fill-in-the-blanks prompt was written for every relation present in the dataset (see Table 2). Next, all possible pairs of entity mentions are collected from each document, which are in turn used to populate the prompts, generating all possible statements as described in Section 3.1.1. The statements are then ranked by their PLL values using BERT as in Equation 2 (larger is better).

For each relation, the training set is used to derive a threshold value which determines whether a statement is kept (considered true/valid) or discarded. Standard metrics can then be used for a quantitative analysis. However, that is not enough to fully evaluate whether such a system is viable for use in the future. To assess that, we gauge the robustness of the approach and to what extent human input is necessary by performing the following experiments:

1. Logical constraints: What happens when we impose domain and range restrictions on the possible responses?
2. Prompt stability: How do different user-generated prompts affect the scores?

3.4. Evaluation Metrics

We evaluate our system using a variety of simple metrics with the aim of exploring how well it works in both automatic and expert-guided settings. For the automatic setting, we calculate a per-relation prediction threshold, then use that to calculate precision, recall, and F_1 scores for every relation. For the expert-guided setting, we use top- k metrics, varying k between 1 and 5. The top-1 metric measures how often the first returned result is correct, such as when a user wants a single, quick answer. The top-5 metrics instead measures how often an expert will be able to locate at least one correct answer within the first 5 responses. As such, each top- k score answers the question *What is the perceived quality of the system if an expert is willing to accept only $k - 1$ incorrect responses?*

4. Results and Analysis

In this section, we present and analyze the scores of the various experiments, in relation to the research questions and our overall aim of the work.

4.1. Automatic Relation Extraction.

The results for the ten most common relations are presented in Table 3. As can be seen in the tables, scores vary a lot, but are in general at a level not usable in practical applications. The same remains true even when the results are enhanced with the domain and range constraints. This means that without further modifications, it is not reasonable to use our system setup as an automatic KG extraction tool. Although these are negative results, it can be seen as a result supporting our hypothesis that not actually extracting a KG from the text, but merely allowing the user to query a virtual KG backed by the text is a much more reasonable route to pursue. In the next section we make a deeper analysis of one of the problems currently affecting these results.

4.1.1. Rare Token Bias

In many cases, statements which included long strings of non-English words were scored higher than expected. This is likely due to how the BERT tokenizer works. Longer, uncommon words are broken down into sub-word tokens. Instances of these tokens may be uncommon in the overall data set, but extremely common when appearing together as a sequence. This means that the scoring metric often disproportionately ranks a token due to its adjacent neighbors, rather than due the sentence as a whole. For example, take the clearly incorrect statement “École nationale supérieure des Beaux - Arts was born in Paris.” This statement has per-token scores of:

[0.999, 0.999, 0.998, 1.000, 1.000, 0.999, 1.000, 1.000, 1.000, 0.159, 0.005, 0.997, 0.620, 0.993] .

All scores related to “École nationale supérieure des Beaux - Arts” are close to 1.0, which is the highest possible. The only words that have low scores are “was,” which BERT suggests replacing with “.”, and “born,” which BERT suggests replacing with “founded”. The fact that

Relation	Threshold	F ₁	Precision	Recall	#TP	#FP	#FN
P17	-1.59	0.09	0.16	0.21	766	3879	2921
	-1.78	0.12	0.23	0.27	993	3414	2638
P27	-1.29	0.05	0.07	0.11	158	1968	1224
	-2.25	0.11	0.15	0.37	500	2909	845
P131	-1.08	0.03	0.04	0.10	158	3487	1348
	-1.58	0.05	0.06	0.21	317	4798	1158
P150	-1.19	0.02	0.02	0.05	34	1486	697
	-1.26	0.02	0.04	0.06	40	1010	668
P161	-1.66	0.01	0.01	0.05	21	1441	382
	-2.53	0.05	0.07	0.21	86	1137	314
P175	-2.76	0.00	0.00	0.12	57	12727	422
	-3.19	0.01	0.01	0.20	95	6665	378
P527	-2.00	0.00	0.00	0.04	10	4616	242
	-2.00	0.00	0.00	0.04	10	3647	235
P569	-2.04	0.01	0.01	0.08	51	6169	601
	-3.40	0.06	0.07	0.43	279	3775	371
P570	-1.95	0.01	0.01	0.07	35	3179	469
	-2.79	0.06	0.08	0.25	124	1477	377
P577	-1.65	0.01	0.01	0.02	8	893	502
	-3.78	0.05	0.06	0.29	146	2211	361

Table 3

Precision, recall, and F₁ scores for the ten most common relations. All scores reported are for the development portion of the data set. The second row for a relation is after applying domain and range restrictions. True and false positives (TP and FP), as well as false negatives (FN) are reported to show the difficulty of discerning correct from incorrect statements using this scoring system.

these two key words have such a low score would imply to a human that the statement as a whole is wrong, yet it still received a very high score of -0.54, due to the remaining scores that were much higher. As a point of comparison, the highest threshold in Table 3 is only -1.08. This means that, in order for the system to function in an automatic setting, a more robust scoring method is needed which can handle these sorts of situations.

4.2. Expert-Guided Relation Extraction

Table 4 again presents results for the ten most frequent relations, this time for the top-*k* metrics as described in Section 3.4. These results are more positive, showing that in most cases, a user will be able to find at least one example of what they are looking for in the top 5 returned statements. We can see that the simple act of restricting the results by domain and range results in a nearly ten-fold increase in scores for a few relations. This supports the idea that a user will need to guide the extraction, either through imposing restrictions based on an ontology, adjusting the text of the query, or adding in additional queries with their own restrictions.

Relation	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
P17	30.3	43.6	51.0	56.3	59.3
	38.7	53.5	63.3	69.7	73.1
P27	17.6	27.0	33.7	38.9	42.0
	37.1	51.3	60.0	65.0	71.0
P131	12.3	19.4	23.5	28.2	34.6
	18.0	27.1	34.8	40.0	45.9
P150	3.5	5.8	7.7	9.6	11.6
	11.4	18.3	22.9	26.8	31.8
P161	11.2	20.9	22.5	24.1	29.0
	29.5	37.7	42.6	60.6	67.2
P175	3.9	9.9	10.8	11.8	13.8
	14.0	27.0	37.0	40.0	43.0
P527	0.0	5.3	8.5	8.5	9.5
	1.1	6.5	8.7	10.9	15.3
P569	2.1	4.2	6.2	7.3	10.4
	26.2	38.8	46.8	52.8	61.8
P570	3.3	3.8	7.2	8.8	12.2
	21.1	33.3	43.8	54.4	62.7
P577	3.6	7.2	8.2	8.7	9.7
	18.0	36.0	43.8	54.1	58.7

Table 4

Top-k scores for the ten most common relations. All scores reported for the development portion of the data set. The second row for a relation is after applying domain and range restrictions.

4.2.1. Prompt Sensitivity

Table 5 shows the top-1 and top-5 scores for the ten most common relations when using different prompts for the same relation. In cases where both the top-1 and top-5 scores are identical for a relation, the user-provided prompts are likely also identical, such as P569 and P570, where all but one participant provided “?x was born on ?y” and “?x died on ?y.” Overall, these results show a fairly large sensitivity to the formulation of the prompt. For example, for P569 and P570 again, the participant who used a different prompt received much lower scores overall. For P527, which was a generally difficult relation, the third participant’s prompt scored almost double that of any other prompt. They used the simplest query; compare “?y is part of ?x” with “One part of ?x is ?y”, “?x has a part which is ?y”, and “?x has ?y as a part of itself”. This implies that, when using such a system, it is important to explore the formulation of the query, as opposed to simply using the first prompt which comes to mind.

5. Discussions and Future Work

In this section, we discuss the implications of our results, with regards to both the proposed method and to outline directions of future research. While the negative results on the automatic KG extraction task clearly shows that this method is, at least for now, not suitable for such a completely automated task, it also supports our hypothesis that a virtual KG query system might

Relation	Original		Participant 1		Participant 2		Participant 3	
	$k=1$	$k=5$	$k=1$	$k=5$	$k=1$	$k=5$	$k=1$	$k=5$
P17	30.3	59.3	15.8	38.2	35.8	62.9	7.0	16.3
	38.7	73.1	24.4	56.4	42.5	73.1	15.1	37.0
P27	17.6	42.0	17.6	42.0	17.6	42.0	17.6	42.0
	37.1	71.0	37.1	71.0	37.1	71.0	37.1	71.0
P131	12.3	34.6	6.1	22.8	12.3	34.6	4.3	13.3
	18.0	45.9	14.9	42.1	18.0	45.9	7.2	26.3
P150	3.5	11.6	9.3	18.0	0.6	2.9	1.9	14.1
	11.4	31.8	17.3	38.6	6.5	17.0	9.8	36.0
P161	11.2	29.0	11.2	17.7	12.9	38.7	6.4	30.6
	29.5	67.2	27.8	54.1	31.1	62.3	34.4	59.0
P175	3.9	13.8	3.9	13.8	0.9	16.8	0.9	4.9
	14.0	43.0	14.0	43.0	11.0	40.0	11.0	34.0
P527	0.0	9.5	1.0	10.6	2.1	8.5	8.5	20.2
	1.1	15.3	1.1	14.2	2.2	13.1	10.9	25.2
P569	2.1	10.4	2.1	10.4	0.0	2.4	2.1	10.4
	26.2	61.8	26.2	61.8	16.0	42.3	26.2	61.8
P570	3.3	12.2	3.3	12.2	1.1	5.0	3.3	12.2
	21.1	62.7	21.1	62.7	11.1	36.6	21.1	62.7
P577	3.6	9.7	3.6	9.7	3.6	21.1	3.0	24.7
	18.0	58.7	18.0	58.2	22.1	67.5	23.2	67.0

Table 5

Top-1 and top-5 scores for the ten most common relations, varied by participant-submitted prompt. All scores reported are for the development portion of the data set. The second row for a participant is after applying domain and range restrictions.

be a fruitful direction. This is further supported by the promising results of the top-k results, in the expert-guided setting, and when studying the contributions that logical constraints (such as commonly present in an ontology) can provide, as well as the robustness to changes in the prompts written by different persons. Below, we discuss each of these aspects more in detail.

5.1. Logical Constraints

The experiments in Section 3.3 only dealt with domain and range restrictions. While these showed significant improvement in results when used, there are many more logical constraints available which could be included. As an example, we know that P1365 (replaces) and P1366 (replaced by) each imply the other, and P3373 (sibling) is symmetric. To further bolster the system as a guided way to build an ontology or knowledge graph, one could select any sort of rules allowed by a particular family of description logics (as commonly used for ontologies, such as in OWL) and use those to improve the filtering process.

5.2. Domain Adaptation

One clear shortcoming of this system is its inability to adapt to a given domain. The experiments as they were set up may be too heavily biased by accidental background knowledge. That is, if BERT was trained on any data from Wikipedia or WikiData pages, then there is a high likelihood that statements which are similar to the correct prompts have already been seen by the system. This, however, would require the system to be fine-tuned in some way on a given document. Salazar et al. [12] show how to use pseudo-perplexity (the MLM analogy to traditional LM perplexity using PLL) to measure domain adaptation of a MLM during fine-tuning. Such an approach could be used to fine-tune an MLM to an unseen document to improve ranking, which remains a self-supervised approach.

5.3. Other Language Models

BERT is not the only MLM which could have been considered in this study. Repeating similar experiments with other MLMs, such as RoBERTa [13], is still part of future work. Different classes of LMs should also be studied in this setting. Such exploration would seek to answer the question *How large is the impact of pretraining or model size on the quality of the results in a zero-shot setting?*

Acknowledgments

This work was funded by the Swedish National Graduate School in Computer Science (CUGS). Portions of this work were carried out using the AIOps/Stellar facilities funded by the Excellence Center at Linköping–Lund in Information Technology (ELLIIT).

References

- [1] B. Alkhamissi, M. Li, A. Celikyilmaz, M. Diab, M. Ghazvininejad, A review on language models as knowledge bases, 2022. URL: <https://arxiv.org/abs/2204.06031>. doi:10.48550/ARXIV.2204.06031.
- [2] 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.
- [3] B. Heinzerling, K. Inui, Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 1772–1791. URL: <https://aclanthology.org/2021.eacl-main.153>. doi:10.18653/v1/2021.eacl-main.153.
- [4] 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.

- [5] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, M. Sun, Docred: A large-scale document-level relation extraction dataset, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 764–777.
- [6] D. Vrandečić, Wikidata: A new platform for collaborative data collection, in: Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion, Association for Computing Machinery, New York, NY, USA, 2012, p. 1063–1064. URL: <https://doi.org/10.1145/2187980.2188242>. doi:10.1145/2187980.2188242.
- [7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: <https://aclanthology.org/N19-1423>. doi:10.18653/v1/N19-1423.
- [8] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, A. B. Rios-Alvarado, Openie-based approach for knowledge graph construction from text, Expert Systems with Applications 113 (2018) 339–355. URL: <https://www.sciencedirect.com/science/article/pii/S0957417418304329>. doi:<https://doi.org/10.1016/j.eswa.2018.07.017>.
- [9] G. Nan, Z. Guo, I. Sekulic, W. Lu, Reasoning with latent structure refinement for document-level relation extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1546–1557. URL: <https://aclanthology.org/2020.acl-main.141>. doi:10.18653/v1/2020.acl-main.141.
- [10] J. Shin, Y. Lee, K. Jung, Effective sentence scoring method using bert for speech recognition, in: W. S. Lee, T. Suzuki (Eds.), Proceedings of The Eleventh Asian Conference on Machine Learning, volume 101 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 1081–1093. URL: <https://proceedings.mlr.press/v101/shin19a.html>.
- [11] A. Wang, K. Cho, Bert has a mouth, and it must speak: Bert as a markov random field language model, in: Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation, 2019, pp. 30–36.
- [12] J. Salazar, D. Liang, T. Q. Nguyen, K. Kirchhoff, Masked language model scoring, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2699–2712.
- [13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).