Complementary Roles of Inference and Language Models in QA

Liang Cheng[†] **Mohammad Javad Hosseini**[‡] **Mark Steedman**[†]
[†]University of Edinburgh [‡]Google Research
L.Cheng-13@sms.ed.ac.uk javadh@google.com steedman@inf.ed.ac.uk

Abstract

Answering open-domain questions through unsupervised methods poses challenges for both machine-reading (MR) and language model (LM) -based approaches. The MR-based approach suffers from sparsity issues in extracted knowledge graphs (KGs), while the performance of the LM-based approach significantly depends on the quality of the retrieved context for questions. In this paper, we compare these approaches and propose a novel methodology that leverages directional predicate entailment (inference) to address these limitations. We use entailment graphs (EGs), with natural language predicates as nodes and entailment as edges, to enhance parsed KGs by inferring unseen assertions, effectively mitigating the sparsity problem in the MR-based approach. We also show EGs improve context retrieval for the LM-based approach. Additionally, we present a Boolean QA task, demonstrating that EGs exhibit comparable directional inference capabilities to large language models (LLMs). Our results highlight the importance of inference in open-domain QA and the improvements brought by leveraging EGs.

1 Introduction

Unsupervised open-domain question answering (QA), the task of learning knowledge from a large collection of documents of diversified topics to answer questions, has been a long-standing challenge in NLP, information retrieval and related fields (Moldovan et al., 2000; Brill et al., 2002; Ferrucci et al., 2010).

The traditional machine-reading (MR) approach first extracts a knowledge graph (KG) from an open-domain corpus and then uses the KG for QA (Harrington and Clark, 2007; Reddy et al., 2014; Khot et al., 2017; Meng et al., 2017). This approach offers *explainability*, since the information in KGs is directly supported by the text. However, the relevant assertions need to be exactly stored in the extracted KG, which is often not the case because assertions can be stated in many different ways, while usually only a small subset of them are available in the KG.

On the other hand, language models have been claimed to be capable of performing a wide range of NLP tasks when used in zero-shot or few-shot prompting mode, including open-domain QA, where they have been argued to act as a latent KG over the pretraining data for querying (Petroni et al., 2019; Adolphs et al., 2021; Ali et al., 2021; Onoe et al., 2022; Wang et al., 2020; Radford et al., 2019; Raffel et al., 2019). Advocates of LMs argue that traditional MR approaches relying on KGs built by open relation extraction are prone to errors arising from components like open information extraction and entity linking. In addition to querying LMs directly, it is shown that when relevant context is available and added to the query, the LMs' performance increases significantly (Petroni et al., 2020; Kassner and Schütze, 2020; Chen et al., 2022a). However, while LMs have performed impressively in answering questions on the basis of manually selected contextual documents, their practical usage is limited since automatic retrieval methods do not always return relevant contextual documents to the query.

In this paper, we show that we can leverage directional predicate entailment effectively to alleviate the limitations of both unsupervised approaches to QA. The contributions of this paper are as follows:

(1) We present a comparative analysis of the MR-based and LM-based approaches in multiple QA scenarios. For the MR-based approach, we extract knowledge to construct KGs by parsing a corpus (English Wikipedia in our experiments). For LM-based approach, we follow the previous work in querying the pre-trained LMs. We perform experiments with multiple LMs including BERT (Devlin et al., 2019) and GPT-3.5 (Brown et al., 2020).

Proceedings of the 2nd Workshop in Pattern-based Approaches to NLP in the Age of Deep Learning , pages 75–91
December 6, 2023. ©2023 Association for Computational Linguistics

(2) We alleviate the sparsity issues of the MR-based approach by leveraging directional predicate entailments to infer novel assertions for augmenting the parsed KGs.

(3) For LM-based approaches, we propose an unsupervised method to use predicate entailments for more accurate context-document retrieval, showing significant improvements in cloze-style QA tasks.

(4) We propose a novel Boolean QA task to compare the directional inference capabilities of LMs and EGs, presenting evidence that smaller LMs (BERT and RoBERTa) are far behind in inferential capabilities compared to EGs, while larger LMs (GPT-3.5) have similar but complementary capabilities with EGs. Our analysis suggests a role for both EGs and LMs in open-domain QA.

2 Related Work

Open-domain QA with Machine Reading. MR-based approaches aim to extract knowledge from corpora to answer open-domain questions. It is common to express knowledge as a collection of "facts" in the form of triples (subject, relation, object), where subject and object are entities connected by the relations. The extracted KGs store the collection with entities as nodes and relations as edges, which can be used to answer questions.

Semantic parsing is an efficient open-domain information extraction approach for large corpora (Etzioni et al., 2011; Reddy et al., 2014). Harrington and Clark (2007) propose an effective pipeline that extracts facts by utilizing a localized update algorithm, which transfers sentences into syntax structures and generates KGs incrementally. These MR-based approaches are explainable for QA because every answer is supported by source sentences in the text. However, KGs built in this way are limited to exact match between the question form and the triples in the graph. For example, if a triple *(Amon Bazira, be assassinated in, Kenya)* is extracted from the sentence *"Amon Bazira was assassinated in Kenya"*, the KG would not provide an answer to the question *"Where did Amon Bazira die?"* because the training corpus lacks any sentence constituting an exact match, such as *"Amon Bazira died in Kenya"*. As a result, the parsed KG exhibits high precision but low recall on the task.

Using pre-trained LMs as Latent KG. Petroni et al. (2019) claim that pre-trained LMs encode the knowledge presented in large amounts of texts. They query LMs using "fill-in-the-blank" cloze statements, such as *"Amon Bazira was assassinated in [MASK]"*. They report results on Masked Language Models (MLMs) such as BERT, which are optimized to predict the next word in a sequence or fill in masked words. They show promising performance on cloze-style QA tasks. Ali et al. (2021) propose a method for fact extraction based on BERT, using the BERT sentence-encoding algorithm on a corpus already annotated for named entities. Additionally, Petroni et al. (2020) demonstrate the value of retrieved documents in enhancing BERT's performance. Lin et al. (2021); He et al. (2021); Perez et al. (2021) show improved performance for LMs under few-shot settings. Moreover, Alivanistos et al. (2022); Fichtel et al. (2021) propose approaches to train prompt-learning models with supervised datasets, using the generated prompts to enhance LM performance on open-domain QA. Larger LM models, as shown in the works of Brown et al. (2020), demonstrate better performance.

These results suggest that LMs could work as latent KGs by memorizing vast corpora. However, LLMs are expensive to train, and impractical to update for tasks like questions involving recent news events. Smaller neural LMs are faster to retrain, but fail when natural language inference from limited context is required (Petroni et al., 2020). Attempts to fine-tune these LMs with supervision from Natural language inference (NLI) datasets tend to pick up artifacts and show little evidence of learning *directional* common-sense inferences, such as that, *"be assassinated in"* entails *"die in"* but not the reverse (Li et al., 2022a). In this paper, we query LMs for factual knowledge in a zero-shot setting, but show how the LM-based approach could benefit from the MR-based approach and predicate entailment.

Relational Entailment Graphs. Where a KG has entities as nodes and relations as edges, an Entailment Graph (EG) has relations as nodes and directed edges corresponding to the entailment relation. EGs are usually built by first detecting Distributional Inclusion (Dagan et al., 1999; Geffet and Dagan, 2005) among the set of entity tuples involved in pairs of predicates, and then applying global graph learning algorithms (Berant et al., 2010, 2011; Hosseini et al., 2018, 2021). In this paper, we propose methods that utilize EGs to enhance the performance of MR-based and LM-based methods in knowledge completion, leading to significant improvements in open-domain QA.

3 Method

In §3.1, we propose an unsupervised MR-based method that consists of three key steps: A) constructing a KG by semantic parsing (§3.1.1), B) constructing EGs from text (§3.1.2), and C) augmenting the KG with EGs in an unsupervised way to infer latent knowledge (§3.1.3). We then further augment the KG with LM backoff (§3.1.4). In §3.2, we discuss the LM-based approach and propose a method to enhance the performance by extracting highly-relevant contexts using EGs (§3.2.1).

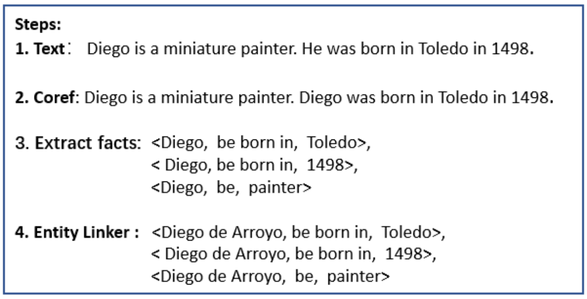
3.1 Machine-Reading Approach

3.1.1 Constructing KG from Corpus

We propose a pipeline to extract KG from corpora with semantic parsing. First, we preprocess the Wikipedia corpus in order to improve the performance of semantic analysis tools by reducing the ambiguity of the raw text. We employ a coreference resolution tool (Lee et al., 2018) to handle coreferences of texts, and then follow Hosseini et al. (2018) and use GraphParser (Reddy et al., 2014) to extract triples from the processed text. GraphParser[1] utilizes a combinatory categorial grammar (CCG) parser (Steedman, 2000) to convert sentences into semantic graphs, which are subsequently transformed into triples. Previous works (Hosseini et al., 2018) show the parser based on CCG performs better than Stanford Open IE (Etzioni et al., 2011; Angeli et al., 2015) in open-domain relation extraction. These extracted triples consist of predicates associated with two arguments. We then assign types to entities by linking them to their corresponding FreeBase IDs using a Named Entity Linking tool, Aidalight (Nguyen et al., 2014). Figure 1 illustrates an example of extracted triples from a raw sentence. After the process, the extracted knowledge is represented in the form of binary predicates and associated entities[2].

3.1.2 Constructing Entailment Graphs

We utilize the EGs extracted from news corpora by Hosseini et al. (2018) as a source of predicate entailments, which is based on the Distributional Inclusion Hypothesis (Dagan et al., 1999; Geffet and Dagan, 2005). The EGs construction algorithm consists of two key steps: local learning and global learning.



Steps:

1. Text: Diego is a miniature painter. He was born in Toledo in 1498.

2. Coref: Diego is a miniature painter. Diego was born in Toledo in 1498.

3. Extract facts: <Diego, be born in, Toledo>,
< Diego, be born in, 1498>,
<Diego, be, painter>

4. Entity Linker : <Diego de Arroyo, be born in, Toledo>,
< Diego de Arroyo, be born in, 1498>,
<Diego de Arroyo, be, painter>

Figure 1: The workflow of extracting knowledge from text.

In the local learning step, we use GraphParser to extract binary relations between a predicate and its arguments from sentences. Subsequently, we compute local distributional similarity scores to learn entailments between predicates with typed arguments. We compute the co-occurrence of predicates associated with the same entities of the same types. Such predicates with matching entities of the same types are assumed to concern the same event or episode. In the global learning step, the EGs learn globally consistent similarity scores based on soft constraints that consider both the structures across typed entailment graphs and inside each graph. In our EGs construction process, we compute the BInc score (Szpektor and Dagan, 2008) as the directional entailment score between predicates and use it as the input to the global graph learning step.[3]

3.1.3 Augmenting KG with EG

To augment the KG, we infer latent facts using the EGs. For every triple (e_i, p, e_j) in the KG, we add triples (e_i, q, e_j) for all q in the EG where p entails q. The additional triples result in a larger augmented KG with reduced sparsity. Figure 2 illustrates an example of adding latent links to a KG. In this example, the EG indicates that the predicate *"be assassinated in"* entails *"die in"* for arguments of types *(person, location)*. Given the fact *(Amon Bazira, be assassinated in, Kenya)* stored in our KG, we add the latent fact *(Amon Bazira, die in, Kenya)*. A query such as *"Where did Amon Bazira die?"* now returns the correct answer. It is crucial to note that the inference is directional. In this in-

[1] The code of GraphParser is available at https://github.com/sivareddyg/graph-parser

[2] The works of KG construction are available at https://github.com/LeonChengg/entGraphQA.git

[3] We also experimented with two other EGs (Hosseini et al., 2021; Chen et al., 2022b) which resulted in consistent results (Appendix B).

stance, we can not infer *"be assassinated in"* from *"die in"*.

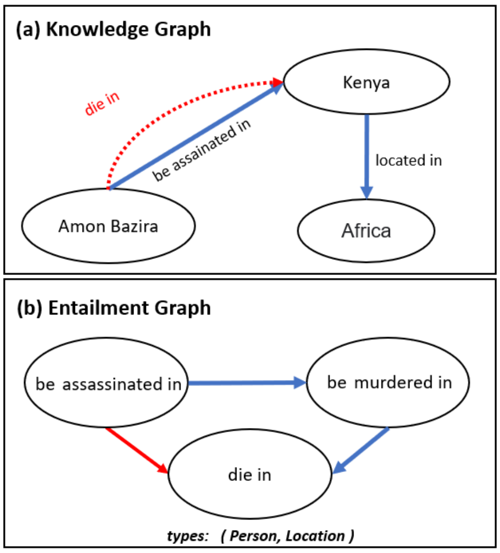


Figure 2: An example of adding latent knowledge. (a) The missing relation *"die in"* is added by using the entailment *"be assassinated in"* entails *"die in"*. (b) Part of the EG for arguments of types *(person, location)*.

If we augment the entire KG extracted from Wikipedia with the EG in an offline manner, the memory requirements for storing the KG becomes prohibitively large. To address this issue, we propose an online approach for KG augmentation for open-domain QA, reducing the storage requirements of the KG without compromising precision. For each query, we simultaneously use both the KG and EGs. If a query *(entity, q, [target entity])* does not yield any results in the KG, it returns *"not found"* even if the target entity could be inferred. To resolve that, we query the EG to get candidate predicates p that entail q. The predicates are sorted based on their entailment scores into a list P = [p_1, p_2, ... , p_n], where each p_i (1 $\leq$ i $\leq$ n) entails q. We start from the beginning of the list and iteratively query the KG with *(entity, p_i, [target entity])*. We return the first matched target entity, or *"not found"* if there is no match.

For instance, if a query such as *(Amon Bazira, die in, [MASK])* does not yield any matching facts in our KG, we search the EG. In the EG, *"suicide in"* and *"be assassinated in"* entail *"die in"*. We sort *"suicide in"* and *"be assassinated in"* based on their entailment scores. First, we replace *"die in"* with *"suicide in"*, generating a modified query *(Amon Bazira, suicide in, [MASK])*. If this query still does not return any facts, we query the KG with *(Amon Bazira, be assassinated in, [MASK])*, which returns an answer *"Kenya"*. This method utilizes the EG as a plug-in without explicitly adding large numbers of triples to the KG.

3.1.4 Backoff augmented KGs with LMs

While the symbolic KGs suffer from sparsity, even when augmented with EGs, LMs return the prediction of a masked token for every question in open-domain QA. To further analyze how we can alleviate the sparsity issues, we evaluate the performance of completing the augmented KG using LMs in QA. For each query, if the augmented KG fails to provide predictions, we utilize the predictions generated by pre-trained LMs to answer it. Both the augmentation method with EGs and the backoff approach with LMs are set up in an unsupervised way to ensure a fair comparison.

3.2 LM-based Approach

In open-domain QA, we utilize pre-trained LMs as latent KGs to provide answers. We explore two conditions when analyzing the prompts of LMs: non-contextual and contextual settings.

Non-Contextual Settings. In this setting, we utilize the original questions as inputs without any additional information. In generative LMs, we directly query the question and consider the returned tokens as the answer. For MLMs, the questions are transformed into "fill-in-blank" statements, where the target tokens are masked and regarded as the answer to be predicted.

Contextual Settings. To analyze the impacts of contexts, we use unsupervised methods to retrieve documents from open-domain corpora. These documents are considered relevant to the questions. For each query, we extract the first paragraph of the most relevant document as the context and concatenate it with the query to generate a new input for LMs.

3.2.1 Retrieving Context with EGs

To measure the enhancements introduced by EGs, we adopt the DrQA (Chen et al., 2017) retriever to extract context from open-domain corpora. This approach enables us to replicate the experimental setup of Petroni et al. (2020), guaranteeing a fair and comparable evaluation. This widely-used and efficient unsupervised retriever relies on term frequency-inverse document frequency (TF-IDF) calculations. However, the limitation of DrQA retriever is lacking inferential capabilities, which results in the omission of relevant documents. For example, when faced with a question like *“Who **played against** Arsenal?”*, the retriever, lacking inferential reasoning, may ignore a relevant document stating *“Manchester City **beat** Arsenal 3-0 to book a place in the Premier League final.”*.

To enhance the inferential capabilities of the retriever, we add EGs into the retrieval process. For each question, we extract new predicates from EGs to generate new questions involving the same entity arguments. According to Distributional Inclusion Hypothesis, if the generated question entails the original question, the answers to the generated question can be used to answer the original question. For example, if the original question is *“Who **played against** Arsenal?”*, we can generate a new question *“Who **beat** Arsenal?”* when the predicate *“beat”* entails *“play against”*. The retrieved document *“Manchester City **beat** Arsenal 3-0 to book a place in the Premier League final.”* contains information that can answer the original question.

To rank the retrieved documents, we define a new scoring function that combines entailment scores:

$$
\begin{aligned}
Score(d_i) &= (1-\alpha) * f(q_{ori}, d_i) \\
&\quad + \alpha * \sum_{j=1}^{k} f(q_j, d_i) * E(q_j, q_{ori})
\end{aligned}
$$

Where q_{ori} represents the original question, and q_j denotes the j_{th} generated question, ordered by entailment scores. The function $f(q_j, d_i)$ calculates the retriever’s score, evaluating the relevance between q_j and the i_{th} document. $E(q_j, q_{ori})$ estimates the probability of q_j entailing q_{ori} using the entailment score from the EG. In our experiments, we set $\alpha = 0.5$ and generate three questions ($k = 3$). By leveraging this scoring function, we concatenate the first paragraph of the most relevant document with the original question as input.

4 Experiment 1: Cloze-style QA

Cloze-style QA aims at answering queries structured as “fill-in-the-blank” cloze statements, which is easy to be evaluated on different LMs without requirements of fine-tuning, especially for MLMs, like BERT-based models. This task has been widely used to measure the capabilities of LMs in memorizing knowledge from the pretraining corpus for open-domain QA. To add both pre-trained Masked LMs and Generative Pre-trained LMs into our analysis of LM-based approaches, we choose this QA task to compare the MR-based and LM-based approaches, and their variants, described in Section 3.

Corpus	**Relation**	**Statistics**	
		Facts	**Rel**
Google-RE	Place-of-Birth	2937	1
	Date-of-Birth	1852	1
	Place-of-Death	796	1
	Total	5527	3
T-REx	Total	31051	41

Table 1: Statistics for the test data

4.1 Dataset

4.1.1 Training and Development Data

We use the English Wikipedia and NewsSpike (Zhang and Weld, 2013) corpora as the training dataset to generate the KG and EGs, respectively. We use YAGO3-10 (Rebele et al., 2016) in our experiments as the development set.

Wikipedia: To include all Wikipedia entities in the training set, we use the whole Wikipedia corpus to extract the KG. The Wikipedia corpus contains 5.4M documents[4]. We extract about 158M binary relations using the semantic parser of (Reddy et al., 2014), GraphParser.

NewsSpike: We use the multiple-source NewsSpike corpus to train the EGs. NewsSpike was deliberately built to include different articles from different sources describing identical news events. The corpus scraped RSS news feeds from January–February 2013 and linked them to full stories collected through a Web search of the RSS titles. It contains 550K articles (20M sentences). We extracted 29M binary relations using the same semantic parser, GraphParser[5] . We train the EG on the NewsSpike corpus independently and use it as a plug-in to augment open-domain KGs for QA.

YAGO3-10: YAGO3-10 is a large semantic knowledge base, derived from Wikipedia, WordNet, WikiData, GeoNames, and other data sources. There are 123K entities and 37 relations in the YAGO3-10. We choose YAGO3-10 as the development set because it is derived from multi-sources, containing low overlaps between our test sets.

[4] The dataset utilized in our research is based on a Wikipedia dump from the year 2021.

[5] The constructed EGs contain all relations of the test set.

4.1.2 Test Set

The LAMA probe (Petroni et al., 2019) dataset requires the models to answer cloze-style questions about relational facts. Our evaluation focuses on the Google-RE and T-REx subsets of LAMA, which is aimed at measuring factual knowledge. For each relation, the LAMA probe provides a manual prompt for querying as well as the Wikipedia snippet evidence aligned with questions.

Google-RE: The Google-RE corpus is manually extracted from Wikipedia and contains 5.5K facts. It covers five relations, where three of them are used in the LAMA probe. The query prompts are pre-defined manually, e.g. *"Steve Jobs was born in [Y]"* for relation *"Place-of-Birth"*. Each fact in Google-RE dataset is associated with a manually selected snippet of text from Wikipedia that supports it. These associated snippets are regarded as the golden context in our contextual experiments.

T-REx: The T-REx (Elsahar et al., 2018) knowledge source is a subset of Wikidata triples. The T-REx in LAMA probe has 41 relations with manual prompts for querying and it subsamples at most 1000 facts per relation. In contrast to the Google-RE knowledge source, which is defined manually, the facts in T-REx were associated with an automatically extracted, and hence possibly irrelevant, Wikipedia snippet. Elsahar et al. (2018) report an accuracy of 97.8% for the alignment.

4.2 Baselines

To compare with the results in LAMA probe, we consider the following baselines.

IE: For the relation-based knowledge sources, we consider the pre-trained Information Extraction (IE) model of Sorokin and Gurevych (2017). This model was trained on a subcorpus of Wikipedia annotated with Wikidata relations. It extracts relation triples from a given sentence using an LSTM-based encoder and an attention mechanism. We add this approach to the baselines because it explicitly stores triples, unlike the LMs.

BERT: Petroni et al. (2019) proved the efficacy of pre-trained MLMs in cloze-style QA. The aim of MLMs is learning to fill the word at the masked position. We add BERT-large (Devlin et al., 2019) in our baselines, which employs a Transformer architecture and trains it on the BookCorpus (Zhu et al., 2015) as well as a crawl of English Wikipedia. The training corpus contains the Wikipedia articles employed in LAMA probe.

	Models	Precision@1	Recall
Single Model	KG	**58.8**	8.5
	BERT	10.5	10.5
	GPT-3.5	19.0	19.0
Augmented Models	KG+EG	**41.7**	17.0
	KG+BERT	20.2	20.2
	KG+GPT	24.3	24.3
	KG+EG+BERT	23.5	23.5
	KG+EG+GPT	26.0	**26.0**

Table 2: We show the Precision@1 and Recall of parsed-KG, BERT-large, GPT-3.5, EG-augmented KG and the EG-augmented KG with LM backoff in non-contextual settings[7].

GPT-3.5: Large Language Models (LLMs), like GPT series models, have shown impressive capabilities in QA. To analyze the performance on LLMs, we take text-davinci-003 (GPT-3.5) as the baseline of evaluation, as it is the largest and best-aligned version[6]. Unlike BERT, the GPT-3.5 is generative. We manually transfer the LAMA probe cloze-style prompts to natural questions for GPT-3.5, like using "where was Steve Jobs born?" instead of "Steve Jobs was born in [MASK]". All prompts for GPT-3.5 are shown in Appendix H.

4.3 Results: Cloze-style QA

The performance of parsed-KGs (MR-based approaches) and LM-based approaches in cloze-style QA is evaluated under two settings: non-contextual and contextual.

Table 2 demonstrates the precision@1 and recall of different models under non-contextual settings. The parsed KG exhibits impressive precision performance due to its high proportion of exact matches but is limited in recall by its sparsity. After being augmented with EGs, the recall improves significantly and the precision is much higher than other combinations (e.g. see KG+EG vs KG+GPT, and KG+EG vs KG+BERT). It demonstrates that EGs perform stronger capabilities of inferring latent knowledge to alleviate the sparsity of parsed KGs. This experiment shows that the MR-based approaches exhibit significantly higher precision compared to LM-based approaches. Additionally, the augmentation of KGs with EGs effectively addresses the recall limitation, still outperforming LMs and their combinations in precision.

[6] In our experiment, we evaluate GPT-3.5 model via the OpenAI API (https://platform.openai.com/), with the temperature setting fixed as 0.

[7] In LAMA probe, there are no negatives so the recall is same as Precision@1 when LMs return prediction for every query.

Dataset		Single Model				EG-Augmented KG	EG-Augmented KG with LM backoff	
	Rel	IE	KG	BERT	GPT-3.5	KG+EG	KG+EG+BERT	KG+EG+GPT
Google-RE	*PoB*	13.8	19.9	16.1	30.3	27.7	30.7	37.0
	DoB	1.9	7.7	1.0	2.0	8.5	9.9	11.3
	PoD	7.2	14.6	14.0	24.7	26.0	29.6	29.7
	Average	7.6	14.0	10.5	19.0	20.7	23.5	**26.0**
TREx	Average	33.8	29.2	31.5	59.1	35.1	64.7	**79.3**

Table 3: Main results on cloze-style QA without context. This table shows the F-score on BERT-large, GPT3, parsed-KG and its augmented versions across the set of evaluation corpora.

	Dataset	BERT-large	GPT-3.5	KG+EG
context$_{NULL}$	Google-RE	10.5	19.0	20.7
	TREx	31.5	59.1	35.1
context$_{DrQA}$	Google-RE	40.8	72.1	20.7
	TREx	43.1	81.7	35.4
context$_{DrQA+EG}$	Google-RE	59.9	**84.0**	20.7
	TREx	54.2	**80.6**	35.4
context$_{Golden}$	Google-RE	78.0	**98.4**	29.6
	TREx	62.6	**95.1**	38.0

Table 4: The F-score of different models in cloze-style QA when context documents are provided, with subscripts "Golden", "DrQA", and "DrQA+EG", indicating the context extraction methods from original snippets, the DrQA retriever, and the version with EGs, respectively.

Models		Google-RE	
		infrequent	*frequent*
MR-based	KG	**15.1**	14.7
	KG+EG	18.7	**19.2**
LM-based	BERT	6.7	**11.2**
	GPT-3.5	16.2	**20.6**

Table 5: The table shows F-scores for subsets of the Google-RE dataset categorized based on frequency.

To further analyze the impact of introducing EGs to various models on F-scores, we present the non-contextual results of cloze-style QA across a range of corpora in Table 3. Among the single models without EGs, GPT-3.5 outperforms other methods, and the parsed KG exhibits better performance compared to BERT. Furthermore, KG+EG presents that augmenting the KGs with EGs leads to an improvement in F-scores. Moreover, the incorporation of LM backoff yields additional improvements in EG-augmented KGs, as shown in the comparison between KG+EG+GPT and KG+EG. The combination of EG-augmented KGs with the GPT-3.5 model backoff (KG+EG+GPT) demonstrates the highest level of performance in terms of F-scores among all combinations. This combination utilizes the high precision benefits provided by EG-augmented KGs while effectively addressing the low recall limitations through the use of LLMs.

Table 4 presents the performance of LMs and KG when provided with contexts. LM-based methods show significant improvement with context, but the impact of context on the KG is limited. This finding indicates that contexts have a more significant impact on LMs compared to parsed KGs. Furthermore, the experiments show that the contexts retrieved by DrQA+EGs outperform those retrieved by the DrQA retriever alone, highlighting the importance and complemantary roles of entailment in retrieving highly relevant contexts for QA. EGs introduce entailment between questions and documents in the retrieval process, contributing to this improved performance.

In order to compare the performance of different EGs trained on different corpora and score functions, we report the results of different EGs in Appendix B. and report the error analysis in Appendix A.

We also analyze the impact of query frequency on LM-based approaches. We run experiments on two subsets of Google-RE queries: the 5% least frequent (*infrequent*) queries by calculating the mentioned entities occurrence in the NewsCrawl corpus (Barrault et al., 2019), and the 5% most frequent queries (*frequent*). As shown in Table 5, LM-based approaches achieve higher F-scores for frequent queries compared to infrequent queries. However, the question frequency appears to have less impact on parsed KG. The results show that LM is limited in effectively answering queries involving infrequent entities, indicating the challenges faced by LM in handling long-tail scenarios.

In conclusion, MR-based approaches reach higher precision but suffer from sparsity, causing low recall in QA. On the other hand, the quality of retrieved contexts is the main limitation of LMs. The contexts extracted by various unsupervised approaches exhibit significant improvements in the LM-based methods, but these approaches show different capabilities in contextual extraction. EGs can enhance both approaches by utilizing tex-

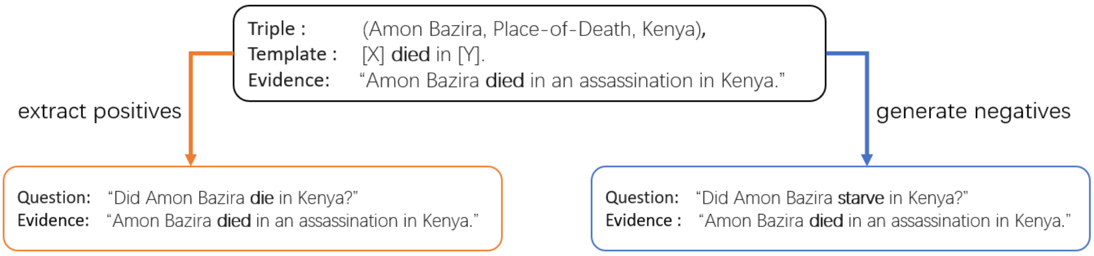

Triple : (Amon Bazira, Place-of-Death, Kenya),
Template : [X] died in [Y].
Evidence: "Amon Bazira died in an assassination in Kenya."
extract positives
generate negatives
Question: "Did Amon Bazira die in Kenya?"
Evidence: "Amon Bazira died in an assassination in Kenya."
Question: "Did Amon Bazira starve in Kenya?"
Evidence : "Amon Bazira died in an assassination in Kenya."

Figure 3: Constructing Boolean QA data by Google-RE and T-REx. (a) The left part shows we extract positives from LAMA probe. (b) We generate the negatives by using the hyponyms to replace the predicate.

tual entailment between common-sense in questions and open-domain corpora. EGs augment the parsed KG by inferring latent knowledge through the entailment between common-sense, enhancing the performance of MR-based methods. For LM-based methods, EGs provide ways to retrieve highly-relevant contexts for questions, by inferring common sense from original questions to latent related documents.

5 Experiment 2: Boolean QA

The LAMA probe is basically an intrinsic evaluation dataset for measuring the capabilities of LMs in extracting knowledge for QA, but it has limitations for evaluating inferential capability (Rogers et al., 2020). One such limitation is that the LAMA probe is derived from the Wikipedia corpus, which is likely to have been included in LMs training data. The LMs tend to choose as answer those triples in the evidence that are similar to those seen in the training data, minimizing dependency on inference, and leading to overestimation of the capabilities of LMs in cloze-style QA. As a consequence, the LAMA probe task fails to evaluate the sensitivity of LMs to directionality of entailment from evidence to the answer to the question.

We propose a Boolean QA task, which adds negative test items to the positive items in the original Google-RE and T-REx datasets (in §4.1.2). We follow McKenna et al. (2021) in automatically generating questions whose answer is not entailed by the original evidence by replacing the relation in the original question by a WordNet hyponyms (Miller, 1998—see figure 3). Such questions are likely to appear to the LMs to be similar to propositions in the evidence, despite not being entailed. The Boolean QA task thereby measures the models' sensitivity to the direction of entailment, as well as the extent to which the EG improves cloze-style QA.

5.1 Boolean QA Data

5.1.1 Extracting Positives

Each instance in Google-RE and T-REx is formed as a triple, like the one shown in Figure 3(a). We transform the fact *(Amon Bazira, die in, Kenya)* into a natural boolean question, such as *"Did Amon Bazira die in Kenya?"*. Then we use the associated Wikipedia snippets from the LAMA probe as the evidence. Since these snippets are provided in the Google-RE and T-REx data, we know that these questions are answerable by the snippets.

5.1.2 Generating Negatives

Negative questions are generated from the positive questions by identifying a hyponym of the relevant predicate using WordNet. Hyponyms usually entail that predicate but are not entailed by it. Therefore it is unlikely that the Google-RE evidence snippet supports the hyponym relation[8]. Such negative questions are difficult for LMs to reject because they are similar to the positive and hence to the text in the evidential snippet.

Figure 3(b) demonstrates an example of negatives generation. In this example, we identify "**starve**" as the hyponym of "**die**" using WordNet. Then a negative *"Did Amon Bazira* ***starve*** *in Kenya?"* will be generated from the positive question *"Did Amon Bazira* ***die*** *in Kenya?"*. The performance in Boolean QA presents the capabilities of directional common-sense inferences, which is crucial for inferring latent knowledge from texts.

5.2 Evaluation on Boolean QA

BERT, RoBERTa (Liu et al., 2019), and GPT-3.5 are the baselines for this task. We evaluate the BERT and RoBERTa by computing cosine similarity between the predicate vector in the question and

[8] We manually checked 100 random samples of generated negatives, and found only 4 cases where a positive answer would be appropriate.

Models	Dataset	
	Google-RE	T-REx
BERT	64.0	47.2
RoBERTa	61.9	49.5
GPT-3.5	87.6	68.1
EG	85.3	67.7
EG+BERT	85.3	71.2
EG+GPT-3.5	**88.5**	**75.0**

Table 6: The F-score in Boolean QA task

the predicate vector in the answer, following the evaluation of McKenna et al. (2021).

For GPT-3.5, we convert the token probability from its outputs using the following mapping:

$$\begin{aligned} score = 0.5 + 0.5 * \mathbb{I}[(\text{output} = True)] * S_{\text{output}} \\ -0.5 * \mathbb{I}[(\text{output} = False)] * S_{\text{output}} \end{aligned}$$

In the equation, $\mathbb{I}$ represents the indicator function. $score$ estimates the probability of positive classification based on the textual model output probability S_{output}, using a linear transformation, which preserves the ordering of model confidences. Note that we add an offset 0.5 to ensure that $0 \leq S_{\text{output}} \leq 1$.

We evaluate EGs by looking for entailment scores between predicates, which are defined on a scale of 0 to 1. For fairness, our EGs are trained on the NewsSpike corpus, which is independent of the evaluation datasets, Google-RE and T-REx. If the predicate in answers is absent from EGs, the model returns the answer as false.

5.3 Results: Boolean QA

To compare the capabilities of directional inference, we report the F-score of Boolean QA in Table 6. The results demonstrate that the EGs and GPT-3.5 perform at a similar level, and they significantly outperform BERT and RoBERTa. We combine the score of EG and LMs with a linear function and show improvement in Boolean QA. The experiments suggest that EGs exhibit stronger capabilities of directional common-sense inference than BERT and achieve a similar level to LLMs, like GPT-3.5, with less training resources and more efficient computation (shown in Appendix E).

Furthermore, the results also prove EGs can identify the directional inference between questions and documents, presenting evidence to explain why EGs can augment the pre-parsed KG and retrieve high-quality contexts for LMs. The successful augmentation explains the efficient enhancement of the parsed KG using EGs in cloze-style QA. The limitation of LMs in directional inference indicates that LMs tend to exhibit a propensity for memorization of factual knowledge rather than a reliance on inferential reasoning in QA scenarios, potentially constraining the practical utility of LMs in QA applications.

6 Conclusion

In this paper, we have conducted a comprehensive analysis of the limitations of Machine-Reading and LM-based approaches in QA. We propose a novel method that utilizes entailment graphs to infer directional relations, addressing the sparsity issue and low relevance of retrieved contexts. Additionally, we have introduced an open-domain Boolean QA task to evaluate the capabilities of directional inference. In Boolean QA, the entailment graphs present stronger capabilities in directional inference than BERT and RoBERTa, achieving comparable performance to GPT-3.5. These results demonstrate the effectiveness of the entailment graphs in enhancing performance under both unsupervised approaches, by making common-sense inference available to open-domain QA.

7 Limitations

We analyze the performance of MR-based and LM-based approaches in QA, and we propose to utilize the directional inference capabilities of EGs to enhance both approaches, showing improvement in QA. A limitation in this work is that it focuses on open-domain cloze-style QA only in English. We have not evaluated our methods on multi-lingual QA tasks, although Li et al. (2022b) have built a large entailment graph for Chinese, which could be applied. The parser, entity typing method used in the entailment graphs, the Boolean QA dataset which is constructed using WordNet, and the LMs, are only language-dependent components. In addition, the parsed KG is extracted from the whole English Wikipedia corpus. Although we can construct the KG incrementally, the program still requires large amounts of memory to run on large corpora. We were not able to construct KGs on more amount of text with our computational resources.

8 Acknowledgements

This research was supported by ERC Advanced Fellowship GA 742137 SEMANTAX, the University of Edinburgh Huawei Laboratory.

References

Leonard Adolphs, Shehzaad Dhuliawala, and Thomas Hofmann. 2021. How to Query Language Models? *arXiv preprint arXiv:2108.01928*.

Manzoor Ali, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. 2021. Unsupervised Relation Extraction Using Sentence Encoding. In *ESWC2021 Poster and Demo Track*.

Dimitrios Alivanistos, Selene Báez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. 2022. Prompting as Probing: Using Language Models for Knowledge Base Construction. *arXiv preprint arXiv:2208.11057*.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global Learning of Focused Entailment Graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global Learning of Typed Entailment Rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022a. Meta-learning via Language Model In-context Tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730.

Zhibin Chen, Yansong Feng, and Dongyan Zhao. 2022b. Entailment Graph Learning with Textual Entailment and Soft Transitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5899–5910.

Ido Dagan, Lillian Lee, and Fernando CN Pereira. 1999. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine learning*, 34(1):43–69.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open Information Extraction: the Second Generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One*, pages 3–10.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building Watson: An Overview of the DeepQA project. *AI magazine*, 31(3):59–79.

Leandra Fichtel, Jan-Christoph Kalo, and Wolf-Tilo Balke. 2021. Prompt Tuning or Fine-Tuning-Investigating Relational Knowledge in Pre-Trained Language Models. In *3rd Conference on Automated Knowledge Base Construction*.

Maayan Geffet and Ido Dagan. 2005. The Distributional Inclusion Hypotheses and Lexical Entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114.

Brian Harrington and Stephen Clark. 2007. ASKNet: Automated Semantic Knowledge Network. In *AAAI*, pages 1862–1863.

Tianxing He, Kyunghyun Cho, and James Glass. 2021. An Empirical Study on Few-shot Knowledge Probing for Pretrained Language Models. *arXiv preprint arXiv:2109.02772*.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R Holt, Shay B Cohen, Mark Johnson, and Mark Steedman. 2018. Learning Typed Entailment Graphs with Global Soft Constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Mohammad Javad Hosseini, Shay B Cohen, Mark Johnson, and Mark Steedman. 2021. Open-Domain Contextual Link Prediction and its Complementarity with Entailment Graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2790–2802.

Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN Search Component to Pretrained Language Models for Better QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3424–3430.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering Complex Questions Using Open Information Extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Tianyi Li, Mohammad Javad Hosseini, Sabine Weber, and Mark Steedman. 2022a. Language Models Are Poor Learners of Directional Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 903–921, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyi Li, Sabine Weber, Mohammad Javad Hosseini, Liane Guillou, and Mark Steedman. 2022b. Cross-lingual Inference with A Chinese Entailment Graph. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1214–1233.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot Learning with Multilingual Language Models. *arXiv preprint arXiv:2112.10668*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Multivalent Entailment Graphs for Question Answering. In *2021 Conference on Empirical Methods in Natural Language Processing*, pages 10758–10768. ACL Anthology.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal Information Extraction for Question Answering Using Syntactic Dependencies in an LSTM-based Architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 887–896.

George A Miller. 1998. *WordNet: An Electronic Lexical Database*. MIT press.

Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. The Structure and Performance of an Open-Domain Question Answering System. In *Proceedings of the 38th annual meeting of the Association for Computational Linguistics*, pages 563–570.

Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. AIDA-light: High-throughput Named-entity Disambiguation. In *Workshop on Linked Data on the Web 2014*, pages 1–10. CEUR-WS. org.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2022. Entity Cloze By Date: What LMs Know About Unseen Entities. *arXiv e-prints*, pages arXiv–2205.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True Few-Shot Learning with Language Models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. 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)*, pages 2463–2473.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683*.

Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: a Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. In *The 15th International Semantic Web Conferenece*.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale Semantic Parsing without Question-Answer Pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting Closer to AI Complete Question Answering: A Set of Prerequisite Real Tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8722–8731.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Daniil Sorokin and Iryna Gurevych. 2017. Context-Aware Representations for Knowledge Base Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK. Coling 2008 Organizing Committee.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language Models are Open Knowledge Graphs. *arXiv preprint arXiv:2010.11967*.

Congle Zhang and Daniel S Weld. 2013. Harvesting Parallel News Streams to Generate Paraphrases of Event Relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Error Analysis of MR-based Approaches

We manually analyze 150 samples for the Machine-Reading approach. About 23% of them are caused by GraphParser and are cases where it returns wrong relations from text. Most of them are caused by non-standard sentences in Wikipedia documents. For example, *"Norman MacLeod (c. 1731 – 1796) was a British army officer, merchant, and official of the British Indian Department."*, the parser cannot extract the fact *(Norman MacLeod, bear in, 1731)* from the sentence because it cannot analyze "(c. 1731 – 1796)". It also leads to bad performance in the relation *"Date-of-Birth"*.

20% of the errors in the KG are due to entity linking returning wrong entities or types, caused by ambiguity in Google-RE and T-REx. For example, a sentence in Googe-RE is *"Jason then continued to Sparta, where he died and was buried"* and the fact in Google-RE is *(Jason, Place-of-Death, Sparta)*. But in evaluation, *"Jason"* is linked to *"Jason Hu"*, who is a modern politician.

About 44% are caused by the mismatch between training and test corpora. For example, the relation *"is connected to"* describes the connections between airports, but we cannot get the knowledge from the training corpus, Wikipedia.

The rest of the errors (13%) are because of other reasons including entailment graphs errors, that are mainly caused by the ambiguity of some high-frequency predicates. For example, predicate *"bear in"* entails predicate *"be from"*. These predicates, like *"be from"*, are common in sentences. If the relation of the query contains these predicates, the KG will return wrong answers easily. When we use the predicate *"be from"* for querying the KG, it will return false results because the predicate has too many meanings. e.g., in the sentence *"Shane Doan is from Arizona"* may mean *"Shane comes from Arizona"*, not the Place-of-Birth. In our experiment, some entailment graphs errors are caused by spurious correlations. For example, there are many documents in Wikipedia like *"Steve Jobs was born on February 24, 1955, in California, ..., Jobs died at his Palo Alto, California home around 3 p.m."*. From these sentences, we may extract facts like *(Steve Jobs, bear in, California)* and *(Steve Jobs, die in, California)*. These predicates link the same entities. It is likely to incorrectly give the entailment relationship between the two predicates.

B Different Entailment Graphs on cloze-style QA

EGs play a crucial role in capturing the relationships between typed predicates, utilizing a score function to measure the probability of one predicate entailing another. Some works introduced various models for generating EGs with improved quality in NLI datasets. Hosseini et al. (2021) proposed the Contextualized and Non-Contextualized Embeddings (CNCE) model, which leverages contextual link prediction to calculate a novel relation entailment score. Similarly, Chen et al. (2022b) introduced the Entailment Graph with Textual Entailment and Transitivity (EGT2) method, demonstrating promising performance on Recognizing Textual Entailment (RTE) tasks.

To evaluate the performance of state-of-the-art (SOTA) entailment graphs in cloze-style QA, we compare their performance in augmenting parsed KGs. We specifically investigate the impact of different training sets by training the entailment graphs on three distinct corpora: Wikipedia, NewsSpike, and NewsCrawl (Barrault et al., 2019). We present the summarized results of the different entailment graphs in Table 7.

	P@1	R	F
BERT-large	10.5	-	10.5
RoBERTa	4.8	-	4.8
Transformer-XL	1.6	-	1.6
GPT-3.5	19.0	-	19.0
KG	**58.8**	8.5	14.0
KG+EG$_{wiki\_binc}$	43.8	12.3	17.4
KG+EG$_{ns\_binc}$	41.7	15.0	20.7
KG+EG$_{ns\_cnce}$	40.7	**16.2**	**21.0**
KG+EG$_{ns\_egt2}$	56.6	9.6	18.7
KG+EG$_{nc\_binc}$	42.6	14.6	19.6
KG+EG$_{nc\_cnce}$	44.9	15.1	20.7

Table 7: Results of different entailment graphs on Google-RE in cloze-style QA. This table presents the mean average precision at one (P@1), recall, and F-score of Google-RE. The result shows the average per number of relations in Google-RE. In this table, the subscripts *wiki*, *ns* and *nc* means the entailment graphs are trained on Wikipedia, NewsSpike and NewsCrawl (Barrault et al., 2019). Subscripts *binc* means EGs constructed using the approach of Hosseini et al. (2018). Subscripts *cnce* and *egt2* means the entailment graphs are trained on CNCE and EGT2.

We notice that the entailment graphs trained on NewsSpike (EG$_{ns\_binc}$) outperform the entailment graphs trained on Wikipedia (EG$_{ns\_wiki}$). Dif-

	KG	KG + EG$_{wiki\_binc}$		KG + EG$_{ns\_binc}$		KG + EG$_{nc\_binc}$	
		local	global	local	global	local	global
P@1	**58.8**	43.2	43.8	42.0	41.7	41.7	42.6
R	8.5	12.3	12.3	13.7	**15.0**	14.3	14.6
F	14.0	16.9	17.4	18.0	**20.7**	19.1	19.6

Table 8: Knowledge graph combined with different entailment graphs. global means the entailment graph is based on global BInc score, local means the entaiment graph with local BInc score.

Corpus	BERT-large	GPT-3.5	KG$_{corpus}$	KG$_{document}$	KG$_{corpus}$+EG	KG$_{document}$+EG
Google-RE	10.5	19.0	14.0	12.9	**20.7**	20.3
T-REx	31.5	**59.1**	29.2	27.8	35.1	33.9

Table 9: The F-scores of different KGs. KG$_{corpus}$ and KG$_{document}$ means the KG is constructed using the whole Wikipedia corpus or retrieved documents.

ferent from the Wikipedia corpus, the articles in NewsSpike mainly describe the same news events by multiple authors. Hence, the predicates in NewsSpike have stronger relevance, which reduces sparsity issues. We analyze the performance of EGs trained by different approaches, EG$_{ns\_binc}$, EG$_{ns\_cnce}$ and EG$_{ns\_egt2}$. We notice the edges in EG$_{ns\_egt2}$ are fewer than the EG$_{ns\_cnce}$ and EG$_{ns\_binc}$. Although the EG$_{ns\_egt2}$ shows impressive performance on RTE tasks, it is limited in sparsity, resulting in bad performance on the QA task. The experiments suggest that the main limitation of augmented KG is the sparsity of EGs in QA.

In order to analyze the effects of global learning, we show the entailment graphs on local and global scores in cloze-style QA in Table 8. The entailment graph based on global scores performs better than entailment graphs just trained on local scores.

C Different Approaches of Open-domain KG Construction

We propose two approaches to construct the open-domain KG in the MR-based method: using the whole Wikipedia corpus (*corpus-based*) or using retrieved documents (*document-based*) to extract knowledge. We analyze the performance of different KG and show the results in Table 9. The document-based KGs require less memory with sacrificing a little performance.

D Analyzing the Impact of Prompts

Petroni et al. (2019) propose the MLM could work as a latent knowledge base for zero-shot cloze-style QA with manual prompts during querying.

Relation	Prompts$_{LAMA}$	Prompt$_{re-written}$
Google-RE	10.5	5.4
T-REx	32.3	16.3

Table 10: Precision of BERT-large querying by different prompts.

We notice some prompts in the LAMA probe are the high-frequency sentences chosen from the test set, Wikipedia. For example, the relation "Date-of-Birth" are labeled with the prompt "[S] (born [O])" for querying. This expression is common in Wikipedia but is not a natural sentence.

To analyze the effects of prompts on MLMs, we evaluate BERT-large on the cloze-style QA with automatic re-written prompts, like replacing the LAMA probe's prompt "[S] (born [O])" with a natural sentence "[S] was born on [O]". The precision of BERT-large is shown on Tabel 10. From the table, if we change the pre-defined manual prompts in the LAMA probe, the precision will decrease significantly. It indicates the LMs attempt to memorize the expression of training data for answering questions, instead of inferring knowledge. High-frequency pre-defined query prompts will improve the performance of LMs but will be limited for practical applications.

E Computational Costs

The KG construction process (MR-based approach) involves two steps: text preprocessing and knowledge extraction. In offline construction, the entire Wikipedia corpus is processed, which requires approximately 6 days when utilizing 20 CPU threads (Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz). However, by leveraging GPUs (GeForce RTX 2080 Ti) for coreference resolution during the preprocessing step, the processing time can be reduced to 36 hours with the use of 4 GPUs. The knowledge extraction step takes approximately 24 hours. Compared to the computational resources required for training GPT-3.5 or BERT-large, the MR-based approach necessitates fewer resources. Furthermore, the parsed KG can be constructed incrementally by adding more documents, and it does not need to load the whole model in KG construction. In online construction, we can dynamically parse the KG based on the retrieved documents.

In our experiments, the training of EGs on the NewsSpike corpus uses 220G of CPU resources over a period of 6 days. Notably, this resource requirement is significantly lower compared to the training of LLMs such as GPT3.5. When it comes to inference, GPT-3.5 necessitates online execution, whereas the augmented KG can be utilized on a local machine. We also experimented with other LLMs like LLaMA-65B (Touvron et al., 2023), which exhibited a response generation time of approximately 1.5 minutes using 4 x A100 (80G) GPUs. This extended response time renders it impractical for use in real-world QA system scenarios.

F Samples of Predicates in Entailment Graph

When querying with the relations from Google-RE, *"Place-of-Birth"*, *"Date-of-Birth"*, *"Place-of-Death"*, we show the samples ranked by the entailment score in EG. The top five predicates in the entailment graphs are shown in Table 11.

Predicate	Types	Top 5 predicates in EG
bear.in	**person-location**	grow.up.in be.in native.of live.in carry
bear.in	**person-time**	name.in address.in have.in be.in live.in
die.in	**person-location**	die.at.home.in die.at dead.found.in suicide.in kill.in

Table 11: Top 5 predicates in entailment graphs

G Additional Implementation Details

In KG construction, we do not perform any hyperparameter tuning when generating KG. We followed the configs of Hosseini et al. (2018) in training entailment graphs, which sets the minimum number of predicates (for each argument-pair), and the minimum number of argument-pairs (for each predicate) to 3. In the evaluation of Boolean QA, we utilize a linear function to combine EG with BERT and GPT-3.5. For the EG+BERT combination, we assign a weight of 0.94 to the EG and 0.06 to BERT. In the EG+GPT-3.5 combinations, the weight assigned to the EG is 0.42.

H Cloze-style Prompts to Natural Question

Questions in LAMA probe are manually formulatd as "fill-in-the-blank" cloze statements. The prompts in LAMA probe are designed for MLM, like BERT. We manually change the cloze-style prompts to natural questions for the generative model such as GPT-3.5, as shwon in Table 12. We conducted a series of experiments involving the utilization of AutoPrompt (Shin et al., 2020) for the automatic generation of prompts for GPT-3.5. However, the performance of prompts generated through this automated process was found to be inferior to those manually curated and labeled. In order to perform a comprehensive analysis of the LMs and make a valid comparison against MR-based approaches, we present the results based on the utilization of manually generated prompts.

I Generating Prompts for Query Automatically

Unlike queries in Google-RE and T-REx using manual-labeld cloze-style prompts, we automatically generate a query for each triple in YAGO3-10 by concatenating the relation names and entities. For example, when querying the triple *(Kobe Bryant, playsFor, Los Angeles Lakers)*, it will be generated as the sentence *"Kobe Bryant plays for [MASK]"* for LMs.

Dataset	Relation Names	Cloze-Style Prompts from LAMA probe	Generated Natural Questions
Google-RE	place of birth	[X] was born in [Y] .	Where was [X] born?
	place of death	[X] died in [Y] .	Where did [X] die?
	date of birth	[X] (born [Y]).	When was [X] born?
T-REx	place of birth	[X] was born in [Y] .	Where was [X] born?
	place of death	[X] died in [Y] .	Where did [X] die?
	subclass of	[X] is a subclass of [Y] .	[X] is a subclass of what?
	official language	The official language of [X] is [Y] .	What is the official language of [X]?
	position played on team / speciality	[X] plays in [Y] position .	What position does [X] play?
	original network	[X] was originally aired on [Y] .	Where was [X] originally aired?
	shares border with	[X] shares border with [Y] .	[X] shares border with whom?
	named after	[X] is named after [Y] .	What is [X] named after?
	original language of film or TV show	The original language of [X] is [Y] .	What is the original language of [X]?
	member of	[X] is a member of [Y] .	[X] is a member of what?
	field of work	[X] works in the field of [Y] .	What field does [X] work in?
	occupation	[X] is a [Y] by profession .	[X] is a what by profession?
	has part	[X] consists of [Y] .	What does [X] consist of?
	diplomatic relation	[X] maintains diplomatic relations with [Y] .	Which conutry does [X] maintain diplomatic relations with?
	manufacturer	[X] is produced by [Y] .	Who produced [X]?
	country of citizenship	[X] is [Y] citizen .	What is the country of [X]?
	language of work or name	[X] was written in [Y] .	Which language was [X] written in?
	continent	[X] is located in [Y] .	Where is [X] located in?
	developer	[X] is developed by [Y] .	Who developed [X]?
	capital of	[X] is the capital of [Y] .	[X] is the capital of what?
	located in the administrative territorial entity	[X] is located in [Y] .	Where is [X] located in?
	languages spoken, written or signed	[X] used to communicate in [Y] .	Which language did [X] use to communicate in?
	employer	[X] works for [Y] .	Who does [X] work for?
	genre	[X] plays [Y] music .	What music does [X] play?
	country	[X] is located in [Y] .	Where is [X] located in?
	position held	[X] has the position of [Y] .	What position does [X] have?
	record label	[X] is represented by music label [Y] .	[X] is represented by what music label?
	location	[X] is located in [Y] .	Where is [X] located in?
	work location	[X] used to work in [Y] .	Where did [X] work?
	religion	[X] is affiliated with the [Y] religion .	[X] is affiliated with the what religion?
	instrument	[X] plays [Y] .	What does [X] play?
	owned by	[X] is owned by [Y] .	Who owns [X]?
	native language	The native language of [X] is [Y] .	What is the the native language of [X]?
	twinned administrative body	[X] and [Y] are twin cities .	Which city and [X] are twin cities?
	applies to jurisdiction	[X] is a legal term in [Y] .	[X] is a legal term in what?
	instance of	[X] is a [Y] .	[X] is a what ?
	country of origin	[X] was created in [Y] .	Where was [X] was created?
	headquarters location	The headquarter of [X] is in [Y] .	Where is the headquarter of [X]?
	capital	The capital of [X] is [Y] .	Where is the capital of [X]?
	location of formation	[X] was founded in [Y] .	Where was [X] founded?
	part of	[X] is part of [Y] .	[X] is part of what?

Table 12: For generative LMs, we generate the natural questions from the cloze-style prompts in LAMA probe. The table shows the mapping between manual prompts and generated questions.

Rels in YAGO	Generated prompts	Examples	
		Query	Answer
isLocatedIn	[X] is loctaed in [Y]	The Safety of Objects is located in [MASK]	United Kingdom
diedIn	[X] died in [Y]	Jean Genet died in [MASK]	Paris
wasBornIn	[X] was born in [Y]	Peter Creamer was born in [MASK]	Hartlepool
hasGender	[X] has gender [Y]	Robert Bly has gender [MASK]	male
playsFor	[X] plays for [Y]	Edgardo Abdala plays for [MASK]	Huachipato
actedIn	[X] acted in [Y]	Charles Durning acted in [MASK]	Tootsie
happenedIn	[X] happened in [Y]	Operation Anaconda happened in [MASK]	Afghanistan
isAffiliatedTo	[X] is affiliated to [Y]	Toni Kuivasto is affiliated to [MASK]	Helsingin Jalkapalloklubi
directed	[X] directed [Y]	Charles Walters directed [MASK]	Lili
isPoliticianOf	[X] is politician of [Y]	Mario Monti is politician of [MASK]	Italy
isCitizenOf	[X] is citizen of [Y]	Nusrat Bhutto is citizen of [MASK]	Iran
dealsWith	[X] deals with [Y]	Togo deals with [MASK]	France
hasOfficialLanguage	[X] has official language [Y]	Guntur has official language [MASK]	Urdu
edited	[X] edited [Y]	V. T. Vijayan edited [MASK]	Saamy
hasCapital	[X] has capital[Y]	Jharkhand has capital [MASK]	Ranchi
hasNeighbor	[X] has neighbor [Y]	Poland has neighbor [MASK]	Lithuania
created	[X] created [Y]	Ilaiyaraaja created [MASK]	Manassinakkare
livesIn	[X] lives in [Y]	Bradley Walsh lives in [MASK]	Essex
wroteMusicFor	[X] wrote music for [Y]	Johnson (composer) wrote music for [MASK]	Thazhvaram
isMarriedTo	[X] is married to [Y]	Livia is married to [MASK]	Augustus
isConnectedTo	[X] is connected to [Y]	Manas International Airport is connected to [MASK]	Kyrgyzstan
participatedIn	[X] participated in [Y]	United States Army participated in [MASK]	Marinduque
hasChild	[X] has child [Y]	William Hague has child [MASK]	Ron Davies
isInterestedIn	[X] is interested in [Y]	Muhammad Taqi Usmani is interested in [MASK]	Tafsir
hasWebsite	[X] has website [Y]	Rural Municipality of Frontier No. 19 has website [MASK]	www.mds.gov.sk.ca/app
isLeaderOf	[X] is leader of [Y]	Xi Jinping is leader of [MASK]	China
hasWonPrize	[X] has won prize [Y]	Philip Hall has won prize [MASK]	De Morgan Medal
influences	[X] influences [Y]	James M. Buchanan influences [MASK]	Elinor Ostrom
isKnownFor	[X] is known for [Y]	Friedrich Engels is known for [MASK]	Marxism
owns	[X] owns [Y]	The Walt Disney Company owns [MASK]	Walt Disney World
worksAt	[X] works at [Y]	Nicholas Kemmer works at [MASK]	University of Edinburgh
graduatedFrom	[X] graduated from [Y]	Ann Richards graduated from [MASK]	Baylor University
exports	[X] exports [Y]	Paraguay exports [MASK]	electricity
hasCurrency	[X] has currency [Y]	Portugal has currency [MASK]	Euro sign
hasMusicalRole	[X] has musical role [Y]	Danny Goffey has musical role [MASK]	piano
hasAcademicAdvisor	[X] has academic advisor [Y]	Robert Lee Moore has academic advisor [MASK]	Oswald Veblen
imports	[X] imports [Y]	Puerto Rico imports [MASK]	fish

Table 13: The queries generated from YAGO3-10