Efficient and Accurate Contextual Re-Ranking for Knowledge Graph Question Answering

Kexuan Sun[1*] **Nicolaas Jedema**[2] **Karishma Sharma**[2] **Ruben Janssen**[2]
Jay Pujara[1] **Pedro Szekely**[2] **Alessandro Moschitti**[2]
[1] University of Southern California
[2] Amazon AGI
{kexuansu, jpujara}@usc.edu
{jedem, karishsc, janruben, szekelyp, amosch}@amazon.com

Abstract

The efficacy of neural "retrieve and generate" systems is well established for question answering (QA) over unstructured text. Recent efforts seek to extend this approach to knowledge graph (KG) QA by converting structured triples to unstructured text. However, the relevance of KG triples retrieved by these systems limits their accuracy. In this paper, we improve the relevance of retrieved triples using a carefully designed re-ranker. Specifically, our pipeline (i) retrieves over documents of triples grouped by entity, (ii) re-ranks triples from these documents with context: triples in the 1-hop neighborhood of the documents' subject entity, and (iii) generates an answer from highly relevant re-ranked triples. To train our re-ranker, we propose a novel "triple-level" labeling strategy that infers fine-grained labels and shows that these significantly improve the relevance of retrieved information. We show that the resulting "retrieve, re-rank, and generate" pipeline significantly improves upon prior KGQA systems, achieving a new state-of-the-art on FreebaseQA by 5.56% Exact Match. We perform multiple ablations that reveal the distinct benefits of our contextual re-ranker and labeling strategy and conclude with a case study that highlights opportunities for future works.

Keywords: Knowledge Graph Question Answering, Contextual Re-ranker, RAG

1. Introduction

Question answering (QA) poses a significant challenge impacting many downstream applications. A popular and effective approach to QA over textual data is to employ a "retrieve-then-generate" (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022) pipeline, in which a retriever extracts important information from textual documents and a generator synthesizes this retrieved information to produce an answer. Recent works have demonstrated that employing a re-ranker to increase the relevance of retrieved information before answer generation achieves state-of-the-art results on popular text QA benchmarks (Lee et al., 2022; Chowdhury et al., 2022). Given the broad efficacy of these retrieval-based approaches, the extension and application of such systems to structured data sources, such as Knowledge Graphs, is a natural research direction.

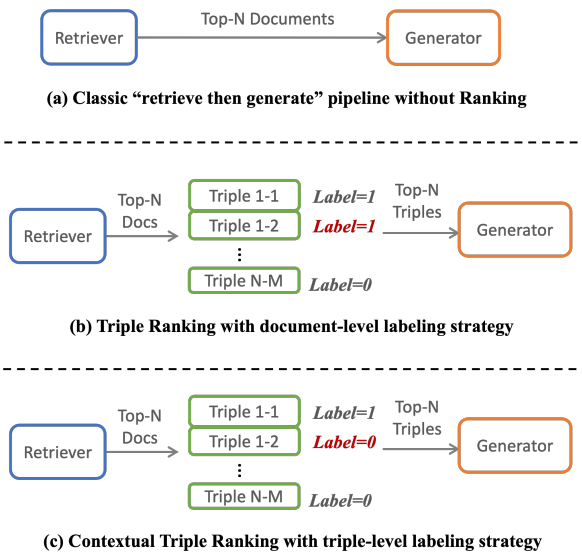
Knowledge Graphs (KGs), such as Freebase (Bollacker et al., 2008) and Wikidata (Vrandečić and Krötzsch, 2014), are high-quality, richly structured sources that contain unique information not easily found in unstructured text. QA over KGs (KGQA) remains a popular parallel QA challenge. While semantic parsing approaches have traditionally dominated KGQA leaderboards, recent studies like Unik-QA (Oguz et al., 2022) and DecAF (Yu et al., 2023) have sought to extend the *retrieve-then-generate* approach to KGQA by converting *<subject, relation, object>* triples into natural language sentences. The primary challenge of this line of research is the substantial scale and complexity of KGs, which make it difficult to identify the relevant triples necessary to generate a correct answer. Prior works like DecAF (Yu et al., 2023) attempt to overcome this challenge with a specialized "reader" module that produces a logical form, which is executed against the KG to retrieve relevant triples before answer generation. While effective, this approach is limited: 1) requires extensive logical form training data, which is not always available, and 2) introduces latency due to logical form execution.

This paper aims to identify important triples by employing a carefully designed re-ranker. The re-ranker improves the selection of triples by exploiting "context": triples that share a subject entity with the retrieved candidate triple. In this way, the context provides extra information to help determine the relevance of candidate triples to user questions. This rich context helps disambiguate similar candidates by uncovering dependencies between questions and the subgraph surrounding retrieved triples, enhancing the accuracy of rankings. Specifically, we 1) retrieve over "entity-centric" documents that contain triples with the same subject entity, 2) re-rank triples by relevance using context obtained from the 1 hop neighborhood of retrieved subject entities, and 3) generate a final answer using top-K re-ranked triples.

Existing KGQA benchmarks do not include la-

[*] Work done during an internship at Amazon.

LREC-COLING 2024, pages 5585–5595
20-25 May, 2024.



(a) Classic "retrieve then generate" pipeline without Ranking

(b) Triple Ranking with document-level labeling strategy

(c) Contextual Triple Ranking with triple-level labeling strategy

Figure 1: The illustration of 1) classic retrieve-then-generate pipeline without re-ranking, 2) re-ranking with the document-level labeling strategy (*triples from the same document have the same label, e.g., Triple 1-1 and 1-2 are both labeled as positive*), and 3) re-ranking with the triple-level labeling strategy (*triples from the same document may have different labels because they are individually labeled based on relevance, e.g. Triple 1-1 and 1-2 are labeled as positive and negative, respectively*).

bels for re-ranker training. This work studies two labeling strategies that exploit KG structure to derive these labels automatically: *1) a naive "document-level" labeling strategy* that coarsely categorizes triples by their presence in a relevant document and *2) a novel "triple-level" labeling strategy* that leverages the co-occurrence of topic and answer entities to granular, higher quality labels. For example, given the question "*Who is Justin Bieber's brother*," "document-level" labeling considers all triples containing "Jaxon Bieber" as positives. In contrast, the "triple-level" strategy only considers the most relevant triple `<Jaxon Bieber, sibling, Justin Bieber>` as positive. Figure 1 shows the difference between 1) the traditional "retrieve then generate" pipeline and the "retrieve, re-rank and generate" pipelines, and 2) "document-level" and "triple-level" labeling strategies for training the re-ranker.

We extensively study the ability of the contextual re-ranker to improve the relevance of retrieved triples and overall KGQA performance on the popular FreebaseQA (Jiang et al., 2019) and WebQSP (Yih et al., 2016) benchmarks. This paper: **1)** shows that incorporating a contextual re-ranker is an efficient and accurate way to increase the relevance of the information provided to the generator and improve KGQA performance, increasing the Exact Match score on FreebaseQA by 5.56% over the prior state of the art; **2)** investigates two novel labeling strategies to infer labels for contextual re-ranker training and studies its benefits on re-ranker and overall KGQA performance; **3)** demonstrates how to construct "entity-centric" retrieval documents to produce context that improves the re-ranker performance for the KGQA task.

2. Related Work

In this section, we discuss prior studies closely related to this work.

KGQA *Semantic parsing* and *Information retrieval* are two primary approaches to KGQA, with semantic parsing receiving the bulk of study historically. In general, given the question, these approaches aim to rank or generate SPARQL queries that can answer the question using the entity relationships. For example, approaches such as QGG (Lan and Jiang, 2020), and SPARQA (Sun et al., 2020) use pre-defined templates and/or constraints to identify logic forms. RnG-KBQA (Ye et al., 2022) enumerates and ranks a pool of candidate logic forms and applies a sequence-to-sequence generator to provide final ones. Another recent approach ArcaneQA (Gu and Su, 2022) reduces the search space with dynamic program induction. These approaches usually rely on the SPARQL executor to provide final answers.

Another major class of approaches is based on information retrieval. For example, PullNet (Sun et al., 2019) iteratively retrieves subgraphs around the topic entity and identifies answer entities using graph convolutional networks. Besides KGs, Graft-Net (Sun et al., 2018) also considers textual information and retrieves heterogenous subgraphs to answer questions. EmbedKGQA (Saxena et al., 2020) learns representations of all entities and ranks entities based on scores between the question and entity representations.

More recently, a few studies started to apply sequence-to-sequence language models to directly generate answers. For example, KGT5 (Saxena et al., 2022) trains a sequence-to-sequence model to directly generate answers; Unik-QA (Oguz et al., 2022) first uses an entity linking model to identify a subgraph, then applies a dense retriever to retrieve relevant facts and finally apply the sequence-to-sequence model to generate answers using the retrieved facts. Since most prior approaches are based on entity-linking systems, another recent paper DecAF (Yu et al., 2023) proposed to get rid of the entity linker and directly retrieve data from an indexed KB. With the retriever, DecAF is trained to generate both logic forms and direct answers and combine their results. Both Unik-QA and DecAF

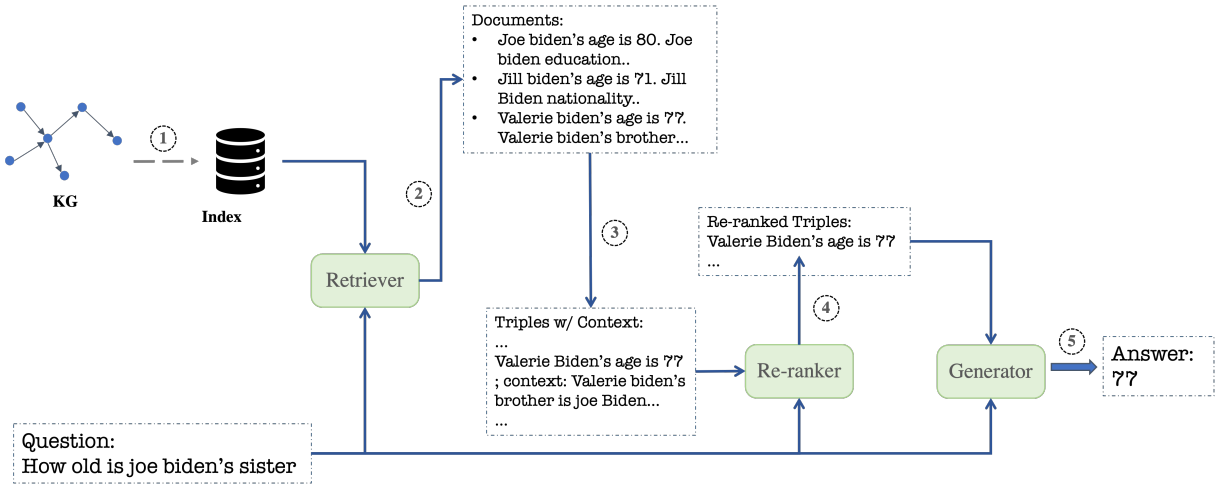


Figure 2: The proposed framework for KBQA. The framework contains three modules: **Retriever**, contextual **Re-ranker** and **Generator**. It works with the following steps: 1) verbalize KG and create an index for the documents, 2) retrieve documents with the question, 3) concatenate individual triples with context for triple-level re-ranking, 4) re-rank all retrieved triples, and 5) combine the question and top-ranked triples, and generate answers.

are closely related to this paper. Similar to DecAF, we remove the entity linking model for more efficiently solving the real-world problem. Similar to Unik-QA, our system focuses on direct answer generation without using logic form annotations.

Retriever - Re-ranker - Generator The retriever and re-ranker pipeline has been successfully used in many previous studies in different natural language tasks (Lewis et al., 2020; Borgeaud et al., 2022). Typically, an efficient retriever obtains relevant documents from a large collection and a more powerful cross-encoder re-ranker refines results with respect to a downstream task. Different from these prior works, we 1) focus on using a KG instead of natural language text which 2) significantly changes the techniques required to develop a powerful re-ranker.

Triple Selection Most factual questions require specific KG triples to provide answers; triple selection is thus an integral component in the development of KGQA systems. Since KG triples can be converted into natural language sentences, this task can be construed as a subgenre of the popular answer sentence selection task (Garg et al., 2019; Jedema et al., 2022). Recent studies (Lauriola and Moschitti, 2021; Han et al., 2021; Liello et al., 2023) have shown that contextual information can significantly improve AS2 accuracy.

3. Retrieve-Rerank-Generate

In this section, we first formalize the KGQA task and then provide details of the proposed pipeline consisting of *Retriever*, *Re-ranker*, and *Generator*, as shown in Figure 2.

3.1. KGQA Task Formulation

A KG can be denoted as $G = \{\langle e_s, r, e_t\rangle \,|e_s, e_t \in E, r \in R\}$ where E and R represent the entity set and the relation set, respectively. Each triple $\langle e_s, r, e_t\rangle \in G$ shows the existence of the relation r between the source entity e_s and the target entity e_t. Given a natural language question q represented by a sequence of word tokens, the task of KGQA aims to find an answer a to q using triples from G. The answer a can either be a natural language sequence or an entity in E.

3.2. Modules

The proposed pipeline works in three steps as shown in Figure 2: 1) **retriever** takes a user-given question as the input and retrieves N relatively relevant documents from the pre-defined KB, 2) **re-ranker** selects the most informative triples from the above documents, and 3) **generator** leverages both the question and top-ranked triples from the previous step to generate final answers.

3.2.1. Retriever

In line with prior works (Yu et al., 2023; Oguz et al., 2022; Guu et al., 2020), we employ both BM25 (Robertson and Zaragoza, 2009) (i.e. sparse) retrieval and DPR (Karpukhin et al., 2020) (i.e., dense) retrieval methods to obtain relevant documents from our indexed KG. BM25 (Robertson and Zaragoza, 2009) leverages TF-IDF (Ramos, 2003) scores for word matches between a query and our indexed KG. DPR (Karpukhin et al., 2020) consists of two BERT-base (Devlin et al., 2019) models to encode questions and documents into

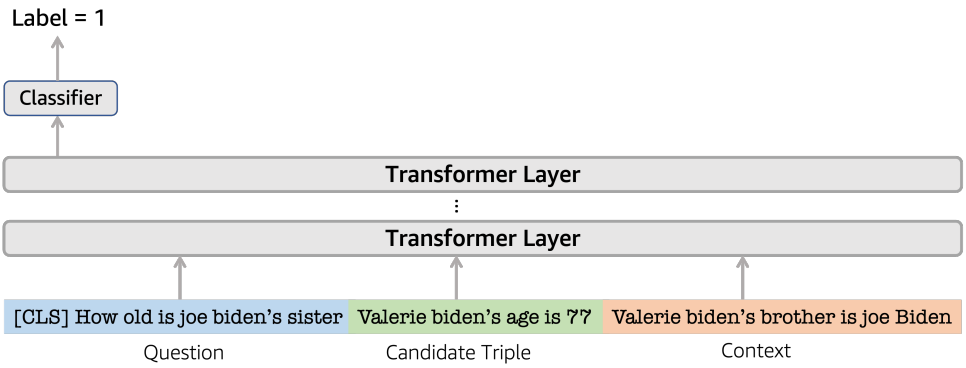


Figure 3: The architecture of the contextual re-ranker. Each input has: **question**, **candidate triple**, and **context** (other triples from the same document and share the same subject entity as the candidate triple). The output is either 0 (irrelevant) or 1 (relevant).

low-dimensional embedding spaces. The two models are trained with a contrastive objective such that the similarities between the encodings of a question and its relevant documents are maximized.

The "verbalization" of KG triples to text is a well-studied problem that can be addressed with both templates (Oguz et al., 2022) and generative models (Agarwal et al., 2021). For example, the triple *"<Will Smith, Age, 54>"* can easily be verbalized with a template to produce *"Will Smith's Age is 54"*. We use this template strategy to verbalize KG triples as single factoid sentences.

However, grouping verbalized triples into retrieval documents with meaningful structure remains an open problem. Prior work either treats each relation as its retrieval document, such as in Unik-QA (Oguz et al., 2022) or applies length-based splitting of randomly grouped triples of the same subject entities, such as in DecAF (Yu et al., 2023). The prior option results in many retrieval documents that lack significant contextual information, such as other relations about the same entity, while the latter potentially split a triple into separate documents. We employ an alternative approach to document generation, where relations sharing the same subject are consolidated into a single document, presented randomly. The condition is that each triple must be contained within a single document. Our document generation method decreases the number of documents in our index, enhancing retrieval efficiency. Simultaneously, it ensures that each relation is situated within the context of other relations featuring the same subject entity and placed within the same document.

3.2.2. Contextual Re-ranker

Recent studies for answer re-ranking on text documents have demonstrated the significant benefits of contextual information, such as sentences within the same paragraph of a target sentence (Lauriola and Moschitti, 2021; Han et al., 2021). Inspired by this prior work, we hypothesize that such contextual information can also be helpful for triple re-ranking within the KGQA setting. The notion of context used in these prior works assumes a natural ordering of information; that is, it assumes that sentences within the same paragraph as a target sentence contain helpful contextual information. The document generation strategy is designed to fulfill this assumption by ensuring that relations within the same document share the same subject and provide useful contextual information.

Contextual Re-ranking Model We extend the ELECTRA-large (Clark et al., 2020) contextual re-ranker proposed by Lauriola and Moschitti (2021) as our re-ranker backbone. Given a question q, the retriever retrieves a list of documents $[D_1, D_2, \cdots, D_m]$ where each document $D_i = [T_1, T_2, \cdots T_n]$ contains a list of triples. For each triple of each document $T_j \in D_i$, the context C_j of T_j is the concatenation of other triples in the same document $T_{1\cdots j-1} + T_{j+1\cdots n}$. The input to the re-ranker is the concatenation q, T_j, and C_j, where these three types of input have different token type embeddings. Figure 3 shows an example of the input and output of the model. In line with prior work, the re-ranker is trained with a classification objective such that the positive triple has label 1 while the negative triple has label 0.

Labeling Strategy The process of assigning high-quality labels is essential for re-ranker training. Prior studies (Glass et al., 2022) use document-level labels for re-ranking documents, in which all documents containing the gold answer are given positive labels. In our framework, this approach degrades label quality by effectively teaching the model that every triple from the correct document is relevant to the question. For example, consider the question "**who is Justin Bieber's brother**" and its

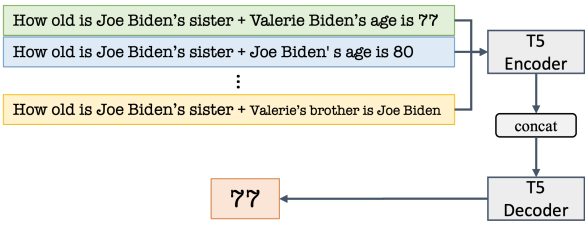


Figure 4: The illustration of the input and output of the FiD. Each re-ranked triple is concatenated with the question and encoded individually. More details of FiD can be found in the original paper (Izacard and Grave, 2020).

answer, "**Jaxon Bieber**". Suppose we retrieve two documents: *"Justin Bieber sibling Jaxon Bieber. Justin Bieber people person ethnicity Canadian"* and *"Jaxon Bieber sibling Justin Bieber"*. Using the "document-level" labeling strategy, the irrelevant triple *"Justin Bieber ethnicity Canadian"* receives the same label as the highly relevant triple *"Jaxon Bieber sibling Justin Bieber* only because they are from the same document. We hypothesize that the "false positive" cases introduced by this labeling strategy impede the ability of the re-ranker to differentiate between highly relevant and largely irrelevant triples.

We propose to mitigate this shortcoming with a fine-grained "triple-level" labeling strategy; this strategy gives a positive label to triples that contain the gold answer and a topic entity. If no such triple exists, it removes the topic entity constraint and checks only for the gold answer. This fallback check is introduced because topic entities are not always provided in dataset annotations, and the answer entity and topic entity are not always within a single hop.

Before training the re-ranker, we first infer gold documents from the KB. For each training question, we traverse over the one-hop subgraphs from both the topic and answer entities. If both entities are involved in a triple, the corresponding document is identified as gold. During training, we process both the inferred gold documents and retrieved documents to create positive and negative samples for the re-ranker using either the "document-level" or "triple-level" strategy.

3.2.3. Generator

Given a question q, the retriever retrieves N documents $D_1, D_2, \cdots D_N$, and the re-ranker produces a confidence score for each triple $T_j \in D_i$ for any i. The triples are re-ranked based on their confidence scores. Only top-K triples are selected and used for answer generation. We use Fusion-in-decoder (FiD) (Izacard and Grave, 2020) as the generator for better information aggregation. FiD uses sequence-to-sequence Transformer-based models T5 (Raffel et al., 2020) as the backbone, encodes documents individually, and aggregates information by incorporating all encoded documents during decoding. In detail, for each selected triple T_j, FiD encodes it with

$$\mathbf{P}_j = Encoder(q; T_j)$$

where $\mathbf{P}_j$ is the hidden representation of T_j. The token embeddings of all passages in the last layer of the encoder are concatenated and passed to the decoder. The decoder generates answers following

$$\mathbf{a} = Decoder(\mathbf{P}_1; \cdots, \mathbf{P}_K)$$

where $\mathbf{a}$ is the generated answer represented as a sequence of word tokens.Figure 4 shows an example input and output of the generator.

4. Experimental Evaluation

4.1. Knowledge Graph

As most of the public benchmark KGQA datasets are created based on Freebase (Bollacker et al., 2008), in line with prior work (Yu et al., 2023), we use the full Freebase pre-processed by Wang et al. (2021) as the KG in our experiments. The total number of entities, relations, and triplets are about 88 million, 20k, and 472 million, respectively. Applying our document generation strategy as described above with at most 10 triples per document results in an index of 107 million documents.

Retriever We evaluate 3 retrieval techniques: BM25 (Robertson and Zaragoza, 2009), DPR (Karpukhin et al., 2020), and a balanced hybrid that sums BM25 and DPR scores, all implemented using Pyserini (Lin et al., 2021). To fine-tune DPR, we label all documents with a least one triple containing both the topic entities and answer entities as positive and leverage BM25 to retrieve hard negatives. Yu et al. (2023) showed that while BM25 and DPR produce comparable results on FreebaseQA, DPR works better on WebQSP. We thus use BM25 to retrieve 500 documents per FreebaseQA question and a Hybrid retriever to retrieve 1000 documents per question on WebQSP.

Re-ranker For each dataset, we extend the ELECTRA-large architecture proposed by Clark et al. (2020) and adapted to the contextual sentence selection task by Lauriola and Moschitti (2021). The re-ranker is trained with a maximum sequence length of 256 and a learning rate 1e-5 for at most 5 epochs. The best training checkpoint is selected using the answer Hit@5 on the dev split of each dataset.

Model	FreebaseQA	WebQSP	
	Hit@1	LF?	Hit@1
FILM (Verga et al., 2021)	63.3	-	-
CBR-SUBG (Das et al., 2022)	52.1	Yes	72.1
PullNet (Sun et al., 2019)	-	Yes	67.8
ReTrack (Chen et al., 2021)	-	Yes	<u>74.7</u>
DecAF (large, 100) (Yu et al., 2023)	79.0	Yes	**80.7**
Unik-QA (Oguz et al., 2022) (base)	-	No	76.7
Unik-QA (Oguz et al., 2022) (large)	-	No	**79.1**
DecAF - Answer only (large, 100)	79.0	No	74.7
Ours (base, 50)	80.9	No	71.8
Ours (large, 50)	**84.3**	No	76.9
Ours (base, 100)	80.2	No	77.2
Ours (large, 100)	<u>84.2</u>	No	<u>77.8</u>

Table 1: The overall performance of our framework. The columns LF? indicate whether the model uses logic forms. FreebaseQA does not have logic from annotations. DecAF - Answer only is a variant of DecAF that does not leverage logic forms. For each category (use or ignore LF), results with the best performance are highlighted in bold font, while those with the second-best performance are underlined.

Generator We fine-tune FiD as our generator using the original FiD implementation (Izacard and Grave, 2020). We report our performance with T5-base and T5-large (Raffel et al., 2020) as the base model. The max sequence length to 400, and the max answer length to 128 for both training and evaluation. We report our results using 50 and 100 triples as input to the generator.

4.2. Datasets

We study our approach on two popular KGQA benchmarks: WebQSP and FreebaseQA.

FreebaseQA (Jiang et al., 2019) consists of questions whose answers are Freebase entities. The train, dev, and test splits contain 20358, 3994, and 3996 questions, respectively. The dataset only contains annotations on topic and answer entities but not logical forms. We use Hit@1 / Exact Match as the evaluation metric.

WebQSP (Yih et al., 2016) is also based on Freebase. The original data only contains train and test splits. For training, we further split the training data into train and dev splits with a ratio of 9:1. The final train, dev, and test splits contain 2789, 309, and 1639 questions, respectively. Besides topic and answer entities, the data also contain logical form annotations. We do not use logical forms in our pipeline. We report Hit@1 as calculated by the official WebQSP evaluation script.

4.3. Experiments

End-to-end KGQA Performance To demonstrate the efficacy of our proposed pipeline, we compare the end-to-end KGQA performance with prior works in Table 1. Our approach outperforms all prior work on FreebaseQA, including DecAF, the prior state-of-the-art, by 5.56% Hit@1. We also note that our (base, 100) and (base, 50) models outperform the prior SOTA while using 550M fewer parameters. On WebQSP, our approach exceeds the performance of PullNet (Sun et al., 2019), ReTrack (Chen et al., 2021), and CBR-SUBG (Das et al., 2022) but falls short of exceeding the results of Unik-QA and DecAF. We argue that DecAF maintains a slight advantage on WebQSP by exploiting the significant volume of logical form training data not used in our approach; we note a significant 3.1% improvement above the directly comparable DecAF Answer-Only setting, which does not exploit this additional data. While Unik-QA outperforms our approach when using FiD-large as the generator, it relies on an entity-linking module. It only creates an index for retrieval of question-specific subgraphs. As explored in prior studies (Soliman et al., 2022), entity linking modules tend to be dataset-dependent, making retrieval-based approaches more practical. Furthermore, rather than generating an index for each question or entity, our method establishes a single index that can be conveniently re-used for any question, potentially enhancing efficiency. Additionally, compared to Unik-QA with FiD-base, the marginally superior performance of our approach suggests that the re-ranking process offers greater advantages for smaller-sized generation modules.

Effect of Context The term "context" is an important component of the input in our contextual re-ranker. To better understand the effect of con-

Ranking Method	Retriever	Re-ranker				Generator
	Hit@500	Hit@1	Hit@10	Hit@100	GT Triple Hit@100	Hit@1
No ranker	98.2	37.8	68.1	90.1	33.6	45.5
Doc-level Label	98.2	39.7	67.4	81.2	77.3	75.4
Triple-level Label	98.2	54.0	84.0	95.2	78.3	80.0

(a) Results on FreebaseQA

Ranking Method	Retriever	Re-ranker				Generator
	Hit@500	Hit@1	Hit@10	Hit@100	GT Triple Hit@100	Hit@1
No ranker	98.1	30.0	53.9	83.7	44.4	57.4
Doc-level Label	98.1	50.3	70.4	79.2	68.9	67.6
Triple-level Label	98.1	73.0	86.0	91.0	74.0	70.5

(b) Results on WebQSP.

Table 2: Re-ranker ablation studies on FreebaseQA and WebQSP. We show the retriever, re-ranker, and generator performance regarding Hit@K. We additionally report the GT Triple hit rate following the abovementioned triple-level labeling strategy. We report the generator performance using FiD-base with 20 passages per question to reduce the computation required.

text, we conduct a **no context** ablation, wherein the re-ranker excludes contextual triples and only considers a candidate triple and the question as input. As with the default setting, the no-context re-ranker is trained to classify the encoding of the concatenated string. We utilize the same training data as in our default setting and fine-tune the model based on a pre-trained AS2 model (Lauriola and Moschitti, 2021). We evaluated the default setting and the no-context ablation using the same retrieved documents and the same generator (FiD-base), focusing solely on the top-20 re-ranked triples during the final answer generation. The results demonstrate that the contextual re-ranker consistently outperforms its no-context counterpart, achieving 82.4 vs. 81.3 on FreebaseQA and 76.8 vs. 75.9 on WebQSP. The results suggest that the "context" offers valuable insights that enable models to distinguish between candidates more effectively, allowing for a more precise identification of the most relevant triples.

Effect of the Labeling Strategy To demonstrate the significant benefits of our re-ranker within our overall pipeline, we perform a **no ranker** ablation and **doc-level labeling** ablation using document-level labeling instead of our proposed triple-level labeling. Table 2 reports the performance of different components in the resulting ablated pipelines in terms of hit rates. The retriever Hit@500 shows that at least one correct document is retrieved within the top 500 for nearly all questions in both datasets. As shown Table 2, both re-ranker approaches substantially improve over the no-ranker setting in terms of end-to-end KGQA performance on both datasets, validating the benefit of incorporating a re-ranker in this setting. However, including a re-ranker trained with "doc-level" labels only improves the Hit@K performance for K values less than 100; this trend does not hold at K>=100, and the performance degrades below the no-ranker setting. This result supports the claim that document-level labeling produces noisy labels. A higher-quality labeling strategy (e.g., our triple-level labels) creates a much more powerful ranker. In other words, both re-ranker and end-to-end KGQA performance are highly influenced by the choice of the labeling strategy: our "triple-level" strategy guides the model to differentiate relevant and irrelevant triples better and thus provide the most relevant information to the generator. We believe that extending and further improving re-ranker label quality is a promising future direction.

In addition to the answer hit rate, we report the gold/ground-truth (GT) triple hit rate, which measures the hit rate of "triple-level labels" in the ranker output. We note that while the answer hit rate@100 varies by only 15%, the GT triple hit@100 and end-to-end KGQA performance vary by a far more significant amount. We additionally note that end-to-end KGQA performance is within 2-4% of the GT Triple hit rate, indicating the utility of our inferred labels as a highly correlated indicator of the overall KGQA performance.

Error Analysis To gain deeper insights into our processing pipeline, we performed a randomized selection of examples where our model produced incorrect predictions. As is shown in Figure 5, these examples were subsequently divided into five primary categories:

1) **Confusing Triples**: In certain instances, the selected triples, while relevant, were incorrect. These confusing triples have the potential to mis-

Question:	where did clay matthews go to school?
Gold Answers:	University of Southern California ; Agoura High School
Predicted Answers:	Georgia Institute of Technology
Error Type:	Confusing Triples
Rationale:	*There are multiple people named Clay Matthews, and one of them graduated from Georgia Tech*
Question:	what all does google now do?
Gold Answers:	Google Maps
Predicted Answers:	Google Maps Engine
Error Type:	Strict Evaluation
Rationale:	*The predicted answer is correct but is marked wrong due to strict evaluation for exact match*
Question:	what shows are shot in new york?
Gold Answers:	Both Sides ; The Stand ; Flight of the Conchords ; Trial Heat
Predicted Answers:	The Big Short
Error Type:	Incomplete Labels
Rationale:	*The show 'The Big Short' was filmed in New York, but it's not listed in the gold answers*
Question:	What 1976-9 UK TV series, written by David Nobbs, frequently featured brief footage of a hippopotamus?
Gold Answers:	The Fall and Rise of Reginald Perrin
Predicted Answers:	Nightmare in the Park
Error Type:	Complex Constraints
Rationale:	*The question imposes multiple constraints that the model fails to meet correctly*
Question:	Who starred alongside Polly James in the first series of The Liver Birds?
Gold Answers:	Pauline Collins
Predicted Answers:	Nerys Hughes
Error Type:	Relative Information
Rationale:	*Nerys Hughes starred in 'The Liver Birds,' but she joined from the 2nd series. The model requires relative or temporal information to answer the question accurately*

Figure 5: Examples for error analysis were sampled from both the FreebaseQA and WebQSP datasets. Each example includes the raw question, the gold answers, the predicted answers from the best-performing model, the error type, and a detailed rationale for the error.

direct the model, therefore degrading the overall system performance. This emphasizes the necessity for designing an effective labeling strategy. A precise strategy with the proper granularity is important for improving the performance.

2) **Strict Evaluation**: In some scenarios, predictions semantically align with the gold answers but are still treated as wrong. It would be beneficial to incorporate auxiliary metrics that evaluate semantic equivalence, ensuring a more comprehensive assessment of our task.

3) **Incomplete Labels**: There exist questions for which the predictions are accurate but are not included in the gold answer set.

4) **Complex Constraints**: Certain questions need the answer to satisfy all specified constraints. However, our model does not consistently meet these requirements. This can potentially be addressed by enriching the training dataset.

5) **Relative Information**: A subset of questions needs to understand sequential or temporal information. Our current system mainly focuses on answer extraction rather than complex reasoning. Future systems could integrate an enhanced generation module, focusing on complex reasoning over multiple triples/documents.

Based on our analysis, we believe integrating a ranking module for the KGQA task is beneficial. The key to this approach is the choice of an appropriate ranking model and developing a precise labeling strategy to train the model, enhancing its capability to select salient information while avoiding ambiguities. Besides the research direction of improving information selection, there is also potential in exploring advanced generation modules that reason within diverse constraints.

5. Conclusions and Future Work

In this paper, we introduce a retriever-re-ranker-generator pipeline for KGQA and illustrate how to design a high-performance re-ranker for this task. Specifically, we improve the relevance of the information provided to the generator by applying a contextual re-ranker to the retrieved triples. In order to mitigate a lack of high-quality triple-level labels in existing KBQA benchmarks, we also introduce a novel triple-level labeling strategy to provide high-quality labels for re-ranker training. We empirically demonstrate that our contextual re-ranker is able to identify salient information from a large KB. As a result, we additionally show that our proposed

retriever-re-ranker-generator pipeline increases the KGQA state of the art on FreebaseQA by 5.56%. We additionally note that the same pipeline improves by 3.1% over the nearest comparable system on WebQSP.

We identify three key directions for future work in our research. First, we believe that improving the context available to the re-ranker beyond the 1 hop neighborhood of retrieved triples will improve its ability to surface complex information not evidenced within a single triple. For example, the context of the re-ranker could be extended to the 2-hop neighborhood of retrieved entities to provide a richer context. This also leads to potential improvements in the labeling strategy that supports multi-hop triples to train the re-ranker. Second, we discuss that augmenting the re-ranker's efficacy can be achieved through the incorporation of higher-quality labels, particularly when dealing with complex questions that necessitate the consideration of multiple significant entities. Examples of such questions include those involving entity comparisons, aggregations, and other complex KGQA scenarios. Third, we believe that extending our re-ranker to rank generated logical forms, similar to Ye et al. (2022). This expansion has the potential to facilitate the development of an efficient and accurate unified system capable of handling both simple and complex KGQA.

6. Limitations

There are a few limitations of our proposed framework. First, our GT triple-level labeling strategy is currently limited to questions whose answers involve single-hop information. In order to enable the ranker to perform well for questions that require multi-hop information may require an extension of our labeling strategy to consider the two-hop neighborhood and beyond. Second, we do not specifically address complex questions, such as those involving aggregations, and it is thus unknown whether our proposed re-ranker will be as effective at ensuring all information relevant to answering these questions is ranked highly. Third, we focus on direct answer generation and do not explore the benefits of reranking in the context of SPARQL query generation, which has recently been shown to be helpful for KGQA.

Lastly, we acknowledge that our proposed pipeline consists of 3 components and the overall parameter count ranges between 550M to 1.32B parameters, depending on the retriever and generator backbone architectures chosen. While our approach is trained module by module on commodity GPU (i.e. AWS p3.16 and p3.24dn), this presents an unfortunate barrier to entry for researchers without industry backing. We believe that this limitation may be addressed through the application of Low-Rank Estimation techniques, which is an interesting line of future work toward democratizing the benefits of our technique to a wider audience.

7. Acknowledgements

We thank colleagues at the Amazon Graphiq team for helpful suggestions at the beginning of this project. We would also like to thank LREC-COLING reviewers for spending their time reviewing this paper and providing insightful comments.

8. Bibliographical References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 2021. Retrack: A flexible and efficient framework for knowledge base question answering. pages 325–336.

Md Faisal Mahbub Chowdhury, Michael Glass, Gaetano Rossiello, Alfio Gliozzo, and Nandana Mihindukulasooriya. 2022. Kgi: An integrated framework for knowledge intensive language tasks.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Robin Jia, Manzil Zaheer, Hannaneh Hajishirzi, and Andrew McCallum. 2022. Knowledge base question answering by case-based reasoning over subgraphs. In *ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2019. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715.

Yu Gu and Yu Su. 2022. ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1718–1731.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Rujun Han, Luca Soldaini, and Alessandro Moschitti. 2021. Modeling context in answer sentence selection systems on a latency budget. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3005–3010, Online. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering.

Nic Jedema, Thuy Vu, Manish Gupta, and Alessandro Moschitti. 2022. Dp-kb: Data programming with knowledge bases improves transformer fine tuning for answer sentence selection.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. 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)*, pages 318–323.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. pages 969–974.

Ivano Lauriola and Alessandro Moschitti. 2021. Answer sentence selection using local and global context in transformer models. ECIR.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher D. Manning, and Kyoung-Gu Woo. 2022. You only need one model for open-domain question answering.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Luca Di Liello, Siddhant Garg, and Alessandro Moschitti. 2023. Context-aware transformer pre-training for answer sentence selection.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Juan Enrique Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.

Hassan Soliman, Heike Adel, Mohamed H. Gad-Elrab, Dragan Milchevski, and Jannik Strötgen. 2022. A study on entity linking across domains: Which data is best for fine-tuning? In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 184–190, Dublin, Ireland. Association for Computational Linguistics.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. *ArXiv*, abs/1904.09537.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. *EMNLP*.

Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. 2020. Sparqa: Skeleton-based semantic parsing for complex questions over knowledge bases. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8952–8959.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and interpretable neural MemoryOver symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3678–3691, Online.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2021. Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 347–357, Online. Association for Computational Linguistics.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2023. DecAF: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The Eleventh International Conference on Learning Representations*.