Dense Retrieval of Knowledge Graphs for Question Answering

Sharmila Reddy Nangi, Michihiro Yasunaga, Hongyu Ren, Qian Huang, Percy Liang, Jure Leskovec
Stanford University
Stanford, USA
{srnangi,myasu,hyren,qhwang,pliang,jure}@cs.stanford.edu

ABSTRACT

Recent works in commonsense question answering are leveraging the unstructured knowledge from powerful language models and structured knowledge from Knowledge Graphs. QA-GNN [26] is one such method giving state-of-the-art performances, but is limited by its reliance on the extraction of contextual subgraph for every QA pair through entity linking and heuristics. To address this limitation, there is a growing need for more generalizable approaches to sub-graph retrieval. There has been an increasing effort of dense retrieval in the language domain [6, 11, 12] which focus on retrieving relevant information from large knowledge sources like Wikipedia through learning better data and query representations. In this work, we extend this approach to the context of graphs and build **DrKG**, a dense retrieval framework for Knowledge graphs in the task of question answering. Our experiments with empirical and qualitative results suggest that our framework extracts subgraphs that show improved performance on multiple datasets for commonsense QA.

KEYWORDS

knowledge graph, information retrieval, question answering

ACM Reference Format:
Sharmila Reddy Nangi, Michihiro Yasunaga, Hongyu Ren, Qian Huang, Percy Liang, Jure Leskovec. 2023. Dense Retrieval of Knowledge Graphs for Question Answering. In *Proceedings of ACM Conference (KDD'23).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Commonsense Question Answering (QA) task refers to the challenge of answering questions that require a deep understanding of everyday knowledge and reasoning abilities. Many state-of-the-art solutions incorporate this commonsense knowledge using large pre-trained language models like BERT and using information from Knowledge Graphs like ConceptNet. These two knowledge sources are *complementary* in nature, as pre-trained language models have information from unstructured documents and KGs have information in structured form, enabling logical reasoning. Prior works in this domain use a combination of Graph Neural Networks and Language Models.

The QA-GNN model [26] is one such work that leverages the power of a pre-trained Large Language Models (LLMs) and Knowledge Graph for end-to-end question answering in commonsense reasoning task. At its core, this model involves several steps. Initially, a vector representation of the question-and-answer (QA) context is procured by utilizing a pre-trained Language Model. It then retrieves a sub-graph from the ConceptNet Knowledge Graph, a process that's carried out heuristically via entity linking. Subsequently, a Graph Attention Network (GAT) is trained on a joint graph that is assembled from the QA context and the heuristically-extracted KG sub-graph, with an ultimate goal to predict the score of the correct answer.

Despite the effectiveness of the QA-GNN model, the method employed for extracting the KG subgraph hinges on a heuristic approach, where all the 2-hop paths emanating from the question and answer entities are included. This mechanism, while useful, has two significant drawbacks. Firstly, it potentially restricts the extraction of relevant nodes that aren't located within the 2-hop network. In other words, there may be valuable information residing in nodes outside the 2-hop network which will be missed by this approach. Secondly, the heuristic nature of the KG extraction procedure is not trainable, meaning it doesn't have the capacity to learn from its mistakes or enhance the quality of subgraph retrieval over time. Thus, despite the initial promise of the QA-GNN model, its potential could be further realized if these limitations were addressed. To address this, there is a growing need for more generalizable approaches to sub-graph retrieval. Subgraph retrieval is crucial to the overall QA performance, as a small subgraph is highly likely to exclude the answer but a large one might introduce noises that affect the QA performance.

Recent advancements in the field of information retrieval from expansive knowledge sources have led to the development of models designed to enhance the accuracy of relevant data extraction. For instance, in the context of open domain question answering, Dense Passage Retrieval framework [11] performs efficient passage retrieval with a simple dual encoder framework that learns dense representations. REALM [6] presents a differential knowledge retriever which allows the model to retrieve and attend over documents from a large corpus such as Wikipedia and pre-train with MLM or fine-tune it for the task of Question Answering. Building on these developments, the Retrieval Augmented Generation (RAG) [12] follows REALM, DPR to retrieve the right passages, with a particular focus on the generation of questions, answers, and tags for evidence-based problem. Drawing inspiration from these works, we believe that expanding this framework to the context of knowledge graphs would be simple and powerful, especially in the tasks of question answering.

Accepted to Second Workshop on Knowledge Augmented Methods for Natural Language Processing, in conjunction with KDD 2023.

KDD'23, August 2023, Long Beach, CA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

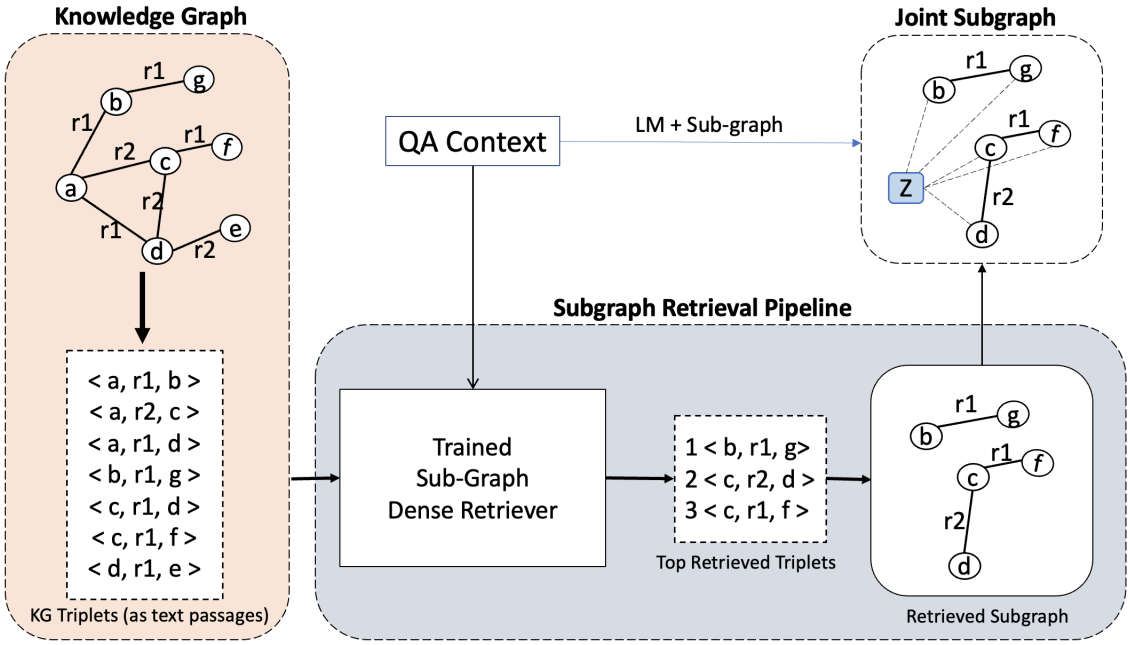


Figure 1: Overview of our approach, Dense Retrieval of Knowledge Graphs for Question Answering (DrKG). The framework include (a) Representing Knowledge Graph triplets as text passages (b) Training a Dense Retriever to extract the relevant triplets in the question-answering context and (c) Construction of the relevant sub-graph from the retrieved triplets.

In this work, we introduce **DrKG**, a new approach to training a dense retriever specifically designed for knowledge graphs, with a focus on the task of commonsense question answering. Our approach aims to harness the benefits of these recent advancements in information retrieval and apply them to the unique challenges presented by knowledge graphs. Additionally, we thoroughly explore the benefits of different training approaches, model designs, pretraining techniques, and perform experiments on the end-to-end question-answering task to assess its efficacy.

2 RELATED WORK

Question Answering with LM+KG: Prior works in question answering [1, 13, 16, 25] employ the representations from LLMs and GNNs to model interaction from both modalities. Recent works like QA-GNN [26], GreaseLM [29] and JointLK[23] adopt joint learning by integrating the language and graph modalities into a joint graph representation through GNNs. However, they rely on retrieval methods for subgraph extraction which are predominantly heuristic, resulting in limited reasoning capabilities, which is addressable through our proposed method.

Trainable Retriever for QA: Dense retrieval for open-domain QA has been initially explored to retrieve relevant passages iteratively using reformulated question vectors [3]. This was further extended by REALM (Retrieval-Augmented Language Model Pretraining)[6] which includes tuning the passage encoder asynchronously by reindexing the passages during training. Dense Passage Retrieval (DPR) [11], presented novel retrieval method for open-domain QA that uses dense vector representations of questions and documents, rather than traditional sparse retrieval techniques. These methods was later extended in the work of Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [12], which uses a seq2seq generator, conditioned on the input and retrieved documents, to produce detailed responses. UniK-QA[18] uses DPR to retrieve relevant information from structured and flattened unstructured knowledge sources. KG-FiD [27] and GRAPE [10] follow a reader-retriever method to retrieve relevant passages with DPR and use graph based methods to capture the relation between them to improve reader performance for QA. However, these methods are designed specifically for open-domain QA and do not extend directly to Commonsense QA, where we need structured knowledge graph information. We draw inspiration from these retrieval based methods in QA and extend it into the task of sub-graph retrieval, which is very crucial in Commonsense QA.

Sub-Graph Retrieval: Extracting subgraphs is a challenging and important problem. Some recent emerging works such as PullNet [22], SRN [19] conducted retrieval by training the retriever, but the retrieving and the reasoning are intertwined, causing the reasoning on partially retrieved subgraphs. Recently, a toolkit designed for Semantic-relevant Subgraph Retrieval, SRTK [20] was developed to facilitate entity linking, retrieval, and visualization on large knowledge sources like Wikidata and Freebase using advanced retrieval algorithms . Another closely related work [28], trainable subgraph retrieval for multi-hop question answering on knowledge bases. Their approach involves training a retriever separately from the

reasoning process, using weakly supervised or unsupervised pre-training and end-to-end fine-tuning with a reasoner. The model employs a dual-encoder method to expand paths and generate subgraphs, with the ability to automatically terminate expansion. However, while there are similarities between their approach and ours, their multi-hop KBQA systems rely on direct answers from the knowledge base and cannot be extended to the more comprehensive CSQA task. In CSQA, a more comprehensive extraction of relevant information is required, and it necessitates a combination of language model knowledge to tackle challenging questions.

3 METHOD

In this work, we propose DrKG, which is a trainable dense retrieval framework for contextual extraction of relevant subgraphs from vast structured Knowledge Graphs. Inspired by the Dense Passage Retrieval approach [11] developed for open-domain question answering, we utilize a dual-encoder framework to expand into relevant sub-graph retrieval. The retrieval pipeline, as depicted in Figure 1, illustrates the steps involved in our approach.

3.1 Knowledge Graph as Text

Our approach begins with structured Knowledge Graphs, such as ConceptNet[21], wherein entities are depicted as nodes and the relations between them as edges. We extract all the triplets from the Knowledge Graph, each comprising two entities connected by a relation, and treat each triplet as a passage or document in a text corpus. This process effectively deconstructs the original problem by transforming the graph structure into a collection of text passages. These newly formed passages serve as an open-domain knowledge source, similar to the corpus used in traditional text-based retrieval methods like Dense Passage Retrieval (DPR). Thus, we conduct neural retrieval on this transformed corpus, maintaining a direct parallel with the original DPR methodology.

3.2 Dense Sub-Graph Retriever

Dense Subgraph Retriever (DrKG) uses a dense encoder $E_P(.)$ which maps any text passages to a d-dimensional real-valued vectors and builds an index for all the M passages that we will use for retrieval. At run-time, DrKG applies a different encoder $E_Q(.)$ that maps the input question to a d-dimensional vector, and retrieves k passages of which vectors are the closest to the question vector. We use a inner-product between the embeddings to measure the similarity/closeness of the query-passage pair.

3.2.1 ***Encoder Models.*** : We use BERT [4] and RoBERTa [15] based models as our encoders. In BERT models, we use the representation of $[CLS]$ token, while in RoBERTa, we use the average representation of all the tokens in the sentence as the as the text/query embedding. We also tried out the more recent Contriever [8] model, which employed contrastive learning for unsupervised dense retrievers which showed significant improvement in performance. More details about the model variants and their performance are presented in the experiments section. During inference time, we index all the passages and query embeddings through FAISS [9], which provides better clustering and similarity search on dense vector representations.

3.2.2 ***Retriever Training.*** : Our goal during the training phase is to fine-tune the encoders in a manner that allows the similarity between the query and passage vectors to effectively function as a reliable ranking metric for retrieval. Essentially, we aim to train the encoder to learn representations that ensure relevant passages (or triplets, in our context) are nearer to the query in the embedding space, while the non-relevant ones are kept at a distance. Similar to DPR, we achieve this by minimizing a negative log-likelihood loss on the positive and negative passage examples for a given query.

3.2.3 ***Training Data.*** : The most challenging part in this setting is to decide what constitutes the *positive* and *negative* passages for a given query, especially when we do not have a ground truth optimal subgraph. In prior works within the text domain, a notion of relatedness to a positive context is usually present, which can be deduced through text representations or by comparing with the ground truth context. However, when representing a Knowledge Graph as text, applying this notion becomes complicated due to the difficulty in designating a single triplet as the ground truth. This has led us to design for multiple positive contexts for each query. In the Commonsense QA datasets, we have question-answer pairs in a multiple choice setting. We thus choose the query and the correct answer pairs for training the retriever. We extract entities from the query and correct answer, and all the triplets that include one of these entities and that form a path from query-answer entities are chosen as the positive context for a question. This approach is guided by the intuition that during the inference time, when a query is input, we want the retriever models to predict the set of triplets that are closer to the query and the correct answer. This methodology is an attempt to create a robust system that can predict the most relevant triplets, thus enhancing the effectiveness and accuracy of the retriever in question answering tasks.

Now, for choosing the negative context, we use **In-batch Negatives** following prior work [2, 5, 7, 11]. In this approach, within a mini-batch consisting of N instances, when provided with a positive context for a question pair, we consider the positive contexts from the remaining N-1 data points as negative samples. This technique has been proven effective for training the encoders efficiently.

3.3 Optimal Subgraph Construction

Once the training of the DrKG encoder model is complete, we proceed to generate and index dense representations for the triplet passages. Subsequently, when presented with a query from our QA dataset, we pass it through the encoder and extract the top 200 triplet passages that exhibit closer proximity to the query in the embedding space. Using this retrieved data, we can construct the graph by incorporating the entities and relationships present in the triplets. This constructed graph is considered as the retrieved optimal subgraph, which is employed during QA-GNN training instead of the sub-graph obtained through heuristic methods.

This approach is better than the heuristics-based sub-graph generation as it surpasses the limitations imposed by potential 2-hop paths involving the extracted QA entities. Furthermore, this method has the capability to retrieve components that are not connected, an outcome unattainable through heuristics alone. Moreover, the KG retrieval process in this framework is trainable, enabling us to enhance the sub-graph generation by training the retriever.

4 EXPERIMENTAL SETUP

To assess the performance of our proposed method, we conducted experiments with QA-GNN to compare its performance with the proposed sub-graph retrieval method, in contrast to the heuristics-based method.

4.1 Datasets

We majorly evaluate on 2 datasets - *CommonsenseQA* [24] and *OpenBookQA* [17]. CommonsenseQA is a 5-way multiple choice QA task that requires reasoning with commonsense knowledge, containing 12,102 questions. OpenBookQA is a 4-way multiple choice QA task that requires reasoning with elementary science knowledge, containing 5,957 questions. We use the data splits similar to QA-GNN. Additionally, we also make use of the RiddleSense [14] dataset with 5,715 examples during the retriever training as it is complementary to the CSQA dataset and introduces novel challenges for the commonsense reasoning community. We use ConceptNet [21], a general- domain knowledge graph, as our structured knowledge source for all of the above tasks.

4.2 Implementation and Training Details

We use the official DPR[2] implementation for training dense graph retriever. For the inference stage, we conducted experiments by varying the number of triplets, namely 50, 100, and 200.Additionally, we explored different combinations of datasets, such as CSQA+OBQA, CSQA+RS, during the retriever's training process. During the construction of optimal sub-graph, we noticed that the model might not extract all the relevant entities in the form of triplets. Therefore, we opted to include the question and answer triplets in the construction of the optimal sub-graph.

Furthermore, we introduced a new relation between the context node and the additional nodes (excluding QA nodes) present in the extracted subgraph. This addition ensures graph connectivity, facilitating effective message passing within the GNN. Consequently, the total number of relations in the QAGNN code increased to 40, compared to the original 38. Apart from this modification, we maintained the same set of hyperparameter and optimizer settings while training QAGNN on both CSQA and OBQA datasets.

4.3 Multiple Choice v/s Open QA

By training the retriever to extract pertinent information from a vast knowledge graph (KG), we gain the flexibility to approach the commonsense QA task as an open domain QA. This implies that we do not rely on any information derived from multiple answer choices in the input data. Instead, we utilize only the query statement for the retriever and include only the question entities during sub-graph construction. Alternatively, in the multiple-choice setting, we employ the (query+answer choice) statement as input to the encoder and utilize both the question and answer choice entities in the construction of the sub-graph.

4.4 Evaluation Metrics

In the case of multiple-choice questions, our objective is to predict the correct option from the given answer choices. Therefore, we assess the performance of the QA-GNN models based on their accuracy in correctly predicting the ground truth answer. We select the model that achieves the highest performance on the dev split and report the accuracies for both the dev and test sets.

[2]https://github.com/facebookresearch/DPR

4.5 Baselines

As baselines for our experiments, we utilize the QA-GNN models trained on CSQA and OBQA datasets respectively. However, we introduce an extension to the baseline models by increasing the number of relations to 40, as previously mentioned. This expansion allows us to investigate the impact of the additional relations on the model performance. Consequently, we repeat the experiments on both the original setting with 38 relations and the modified setting with 40 relations. Additionally, to compare the effectiveness of the retrieval process, we also used a random-retrieved baseline, where we randomly sample the triplets (which are connected with the question and answer entities) to create an optimal sub-graph, as against using a trained retriever.

5 RESULTS AND ANALYSIS

5.1 Quantitative Results

Table 1 and 2 present the QAGNN model performance results with heuristic graph baselines and sub-graphs extracted through different graph retriever modules on CSQA and OBQA datasets.

Encoder	Training Data	Relations	Dev Accuracy	Test Accuracy
QAGNN Heuristic Graphs (Baselines)				
-	-	38	0.785	0.724
-	-	40	0.764	0.732
Random Retrieval Graphs (Baselines)				
-	-	38	0.760	0.686
-	-	40	0.752	0.708
Graph Retriever - Open-Domain QA				
BERT-Base	CSQA	38	0.745	0.720
BERT-Base	No pretraining	40	0.754	0.719
BERT-Base	CSQA	40	0.756	**0.744**
RoBERTa-Base	No pretraining	40	0.761	0.707
RoBERTa-Base	CSQA	40	0.747	0.725
RoBERTa-Base	CSQA+RS	40	0.761	0.708
RoBERTa-Base	CSQA+OBQA	40	0.754	0.704
RoBERTa-Base	CSQA+OBQA+RS	40	0.743	0.707
RoBERTa-Large	CSQA	40	0.756	0.698
Contriever	CSQA	40	**0.763**	0.709
Graph Retriever - Multiple Choice QA				
BERT-Base	CSQA	38	0.731	0.728
BERT-Base	No pretraining	40	0.759	0.694
BERT-Base	CSQA	40	0.756	**0.732**
RoBERTa-Base	No pretraining	40	0.756	0.696
RoBERTa-Base	CSQA	40	0.753	0.724
RoBERTa-Base	CSQA+RS	40	0.745	0.704
RoBERTa-Base	CSQA+OBQA	40	0.750	0.701
RoBERTa-Base	CSQA+OBQA+RS	40	0.745	0.712
RoBERTa-Large	CSQA	40	**0.763**	0.703
Contriever	CSQA	40	0.756	0.713

Table 1: Performance of DrKG modules on CSQA dataset

5.1.1 Number of Relations: Firstly, when comparing the performance of baselines with 38 and 40 relations, we note that the model with 40 relations perform much better in both the datasets. This indicates that the additional relation to connect context node to extra nodes is helpful in passing useful information along the GNN

Encoder	Training Data	Relations	Dev Accuracy	Test Accuracy
QAGNN Heuristic Graphs (Baselines)				
-	-	38	0.616	0.594
-	-	40	0.712	0.658
Random Retrieval Graphs (Baselines)				
-	-	38	0.644	0.624
-	-	40	0.698	0.670
Graph Retriever - Open-Domain QA				
RoBERTa-Base	No pretraining	40	0.690	0.670
RoBERTa-Base	CSQA	40	0.678	0.644
RoBERTa-Base	CSQA+OBQA	40	0.672	0.664
RoBERTa-Large	CSQA	40	0.692	0.672
Contriever	CSQA	40	**0.712**	**0.682**
Graph Retriever - Multiple Choice QA				
RoBERTa-Base	No pretraining	40	0.660	0.622
RoBERTa-Base	CSQA	40	0.678	0.652
RoBERTa-Base	CSQA+OBQA	40	0.690	0.658
RoBERTa-Large	CSQA	40	**0.720**	**0.688**
Contriever	CSQA	40	0.680	0.662

Table 2: Performance of DrKG modules on OBQA dataset

network. This was a strong indication for us to pursue other graph retrieval modules experiments with 40 relations.

5.1.2 Graph Retriever vs Baselines: In our evaluation, we compare the results obtained from different settings of the graph retriever modules, including variations in encoder models, training combinations, and the distinction between multi-domain and open-domain QA, with the baseline models with heuristic graphs. The overall findings reveal that the graph retriever modules outperform the heuristic graph extractions when integrated into the end-to-end QAGNN model training. For the CSQA task, the BERT-base model trained on the CSQA test dataset, along with the RoBERTA-Large model trained on CSQA data, yielded the highest performance in the OBQA task. Additionally, to assess the accuracy of the retrieval process, we contrast the performance of the retrieved graphs with subgraphs constructed through random retrieval. Remarkably, we observe that the randomly sampled triplets (from all the triplets related to the question and answer entities), perform inadequately on both datasets. This outcome provides further evidence that the neural graph retrieval mechanism is capable of learning meaningful information and extracting relevant triplets.

5.1.3 Base v/s Trained Retriever: It is evident from the results that the trained DrKG encoder provides a substantial performance improvement compared to the base model across all tasks. This observation highlights the effectiveness of our chosen training data and the selection of positive and in-batch negative contexts in enhancing the encoder.

5.1.4 Impact of Training Data: To analyze whether the inclusion of additional training data would enhance the retriever's performance, we conducted pre-training of RoBERTa models using various combinations of training data. In the case of the CSQA task, augmenting the CSQA dataset with more data did not result in significant performance improvement for both the multi-domain and open-domain QA settings. This outcome could be attributed to the fact that the CSQA dataset already contains a relatively large amount of data compared to the RS and OBQA datasets. As our ultimate task focuses on CSQA, incorporating training data from OBQA/RS, which potentially originates from slightly different distributions, did not contribute significantly to the training module.

However, for the OBQA task, we observed that the performance was higher when using the CSQA+OBQA combination in the multi-choice setting, compared to the base model or the model trained solely on CSQA. This improvement may be attributed to the limited amount of data available in the OBQA dataset. The introduction of data from the richer CSQA dataset facilitates knowledge transfer, allowing the encoder parameters to learn more effectively.

5.1.5 RoBERTa Base v/s Large Models: In case of OBQA task, we notice that having a RoBERTa large model is very helpful in boosting the performance in all cases. However, in case of CSQA task, while having a large model improves the performance on dev set, there is no significant improvement in the test accuracy. We know that the test data is anonymous and we only us the in-house dataset split for test. So, it would be better to use the model for the original test set to appropriately conclude the impact of large models on CSQA.

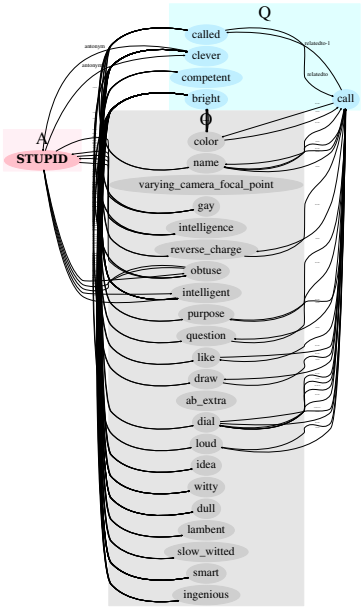
5.1.6 Impact of Contriever: The Contriever encoder proves to be a highly advantageous choice for integration within the graph retriever framework. We observed a remarkable performance enhancement in the OBQA task compared to the baselines. Additionally, in the CSQA tasks, although the Contriever did not surpass the baselines, its performance was notably superior to training the RoBERTa-Large model alone. The Contriever's superiority can be attributed to its pre-training on extensive datasets using contrastive loss. This training methodology enables the model to discern the attributes of similar objects while distinguishing dissimilar objects. Consequently, the Contriever excels in learning meaningful deep representations for both positive and negative contexts, ultimately leading to performance improvement in the tasks at hand.

5.2 Qualitative Analysis

Based on the empirical results of the QA-GNN end-to-end task using the retrieved graphs, it is evident that the retrieved graphs exhibit favorable performance. However, since we lack a ground truth optimal subgraph for measuring the correctness of the extracted subgraph, we conducted a thorough analysis of the graphs obtained from training different retriever models such as BERT, RoBERTa, and Contriever. This analysis aimed to gain insights into the similarities and differences introduced by these models in comparison to the baseline heuristic graphs. More details for each model are discussed below:

5.2.1 Heuristic Graphs. In Figure 2, we present heuristic graphs obtained through a two-hop search in ConceptNet. The graph includes Question (Q) entities (blue), Answer (A) entities (red), and Other (O) entities (grey) within the two-hop connections. The graph is limited to a maximum of 200 nodes, resulting in a high density with around 4,000 edges. To maintain clarity, the figure excludes the display of O-O edges. The selected two-hop nodes reveal a path connecting the question and answer entities.

5.2.2 Graph Retrieval with BERT Encoder. In Figure 3, we present the re-constructed graphs obtained after applying the graph retrieval pipeline with the BERT encoder. In this case, the BERT

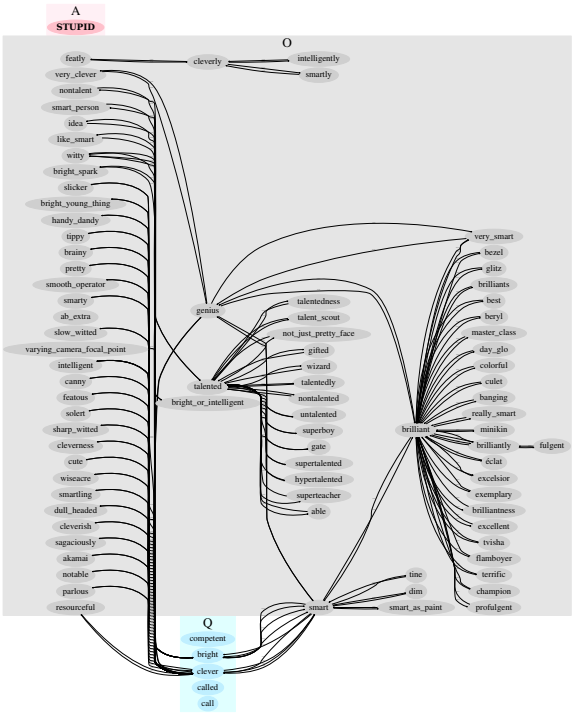


What is someone who isn't clever, bright, or competent called?

Answer: stupid Prediction: dull

(G): QA-GNN

Figure 2: Retrieved graph with two-hop heuristics



What is someone who isn't clever, bright, or competent called?

Answer: stupid Prediction: stupid

(G): QA-GNN

Figure 3: Retrieved graph with BERT encoder in DrKG

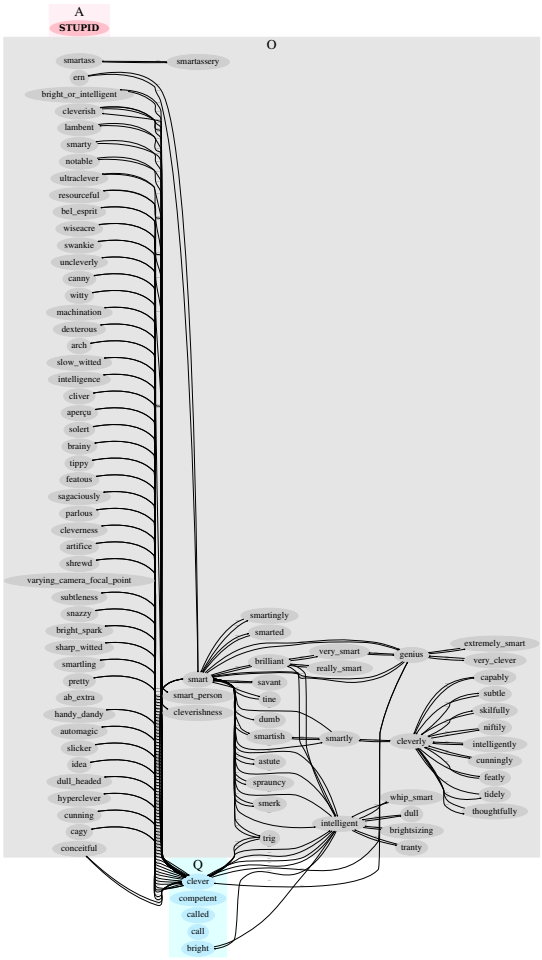
encoder model [4] was pre-trained on the CSQA dataset. As we limit the number of triplets to be retrieved, the resulting graph is not as densely connected as the heuristic graphs. However, we observe that the extracted triplets are closer to the question-answer choice and exhibit greater relevance compared to the heuristic graph. For example, the top 5 retrieved triplets for the Question ,"What is someone who isn't clever, bright, or competent called?" are - (bright, relatedto, bright_young_thing), (bright, relatedto, smart), (bright, relatedto, witty), (clever, relatedto, bright_spark), (clever, relatedto, dull_headed). We notice that these are very closely related to the key entities in the question like "bright" and "clever" and ignores the unrelated entities like "called". In the previous heuristic graph, since we were taking one hop neighbourhood, there are edges from all the question entities even though unrelated. Additionally, there are some very unrelated entities extracted in the heuristic graph like "varying_camera_focal_point", "lambent" which is focussed on the entity "bright" in context of light, but does not make sense in the context of the question. Additionally, despite the extensive extraction of the dense subgraph in the heuristic graph, it failed to correctly identify the answer choice. On the other hand, the graph extracted by the BERT retriever accurately identifies the correct answer. Note that the BERT-retrieved graph is considerably smaller in comparison to the heuristic graph.

5.2.3 Graph Retrieval with RoBERTa Encoder. In Figure 4, we showcase the retrieved graphs obtained using the RoBERTa [15] encoder in the graph retriever. These graphs exhibit similarities to those obtained with the BERT retriever since both models are trained in a similar manner. However, due to the RoBERTa model being trained on a larger corpus and having a better understanding of context, the retrieved graphs demonstrate nodes that are closer in proximity to the contextual understanding of the question.

For the presented example, numerous entities are related to synonyms, related words, and antonyms of "clever" or "bright," providing a richer context for answering the question. It is worth noting that in both the BERT and RoBERTa models, the graphs are not highly connected to the answer entities but primarily connected to the question entities. This observation suggests that greater importance is placed on the context of the question during triplet extraction, rather than direct connections to the answer entities.

5.2.4 Graph Retrieval with Contriever Encoder. In Figure 5, we showcase the retrieved graphs using the Contriever[8] encoder in the graph retriever. The Contriever model, based on the transformer architecture like BERT and RoBERTa, is specifically designed for information retrieval tasks, such as searching a large document database to extract relevant information based on a query.

However, Contriever models differ from BERT and RoBERTa in key aspects. They employ an adaptive span-based approach for retrieving information from documents, instead of relying on fixed-length sequences. This adaptive approach enables more effective retrieval of relevant information from longer documents. The impact of span-based training is evident in the retrieved nodes, which

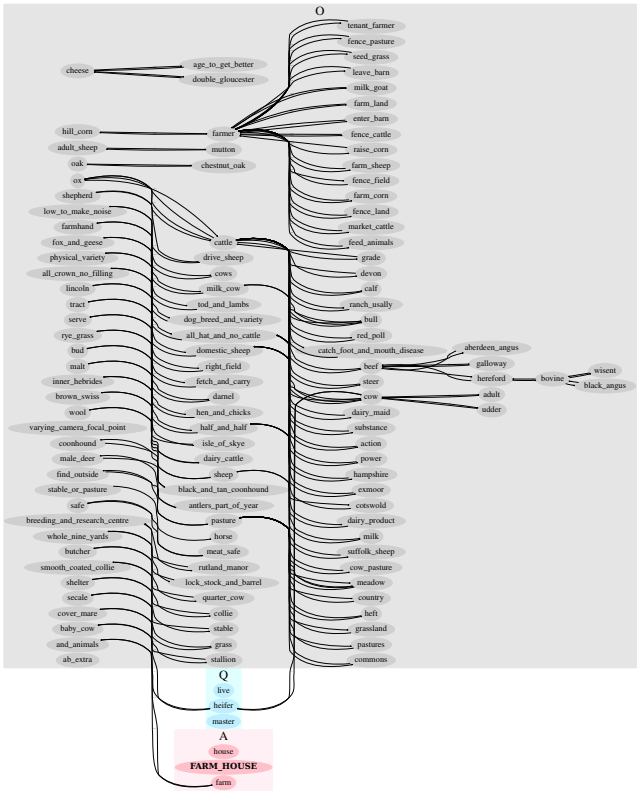


What is someone who isn't clever, bright, or competent called?

Answer: stupid Prediction: stupid

(G): QA-GNN

Figure 4: Retrieved graph with RoBERTa encoder in DrKG



Where does a heifer's master live?

Answer: farm_house Prediction: farm_house

(G): QA-GNN

Figure 5: Retrieved graph with Contriever encoder

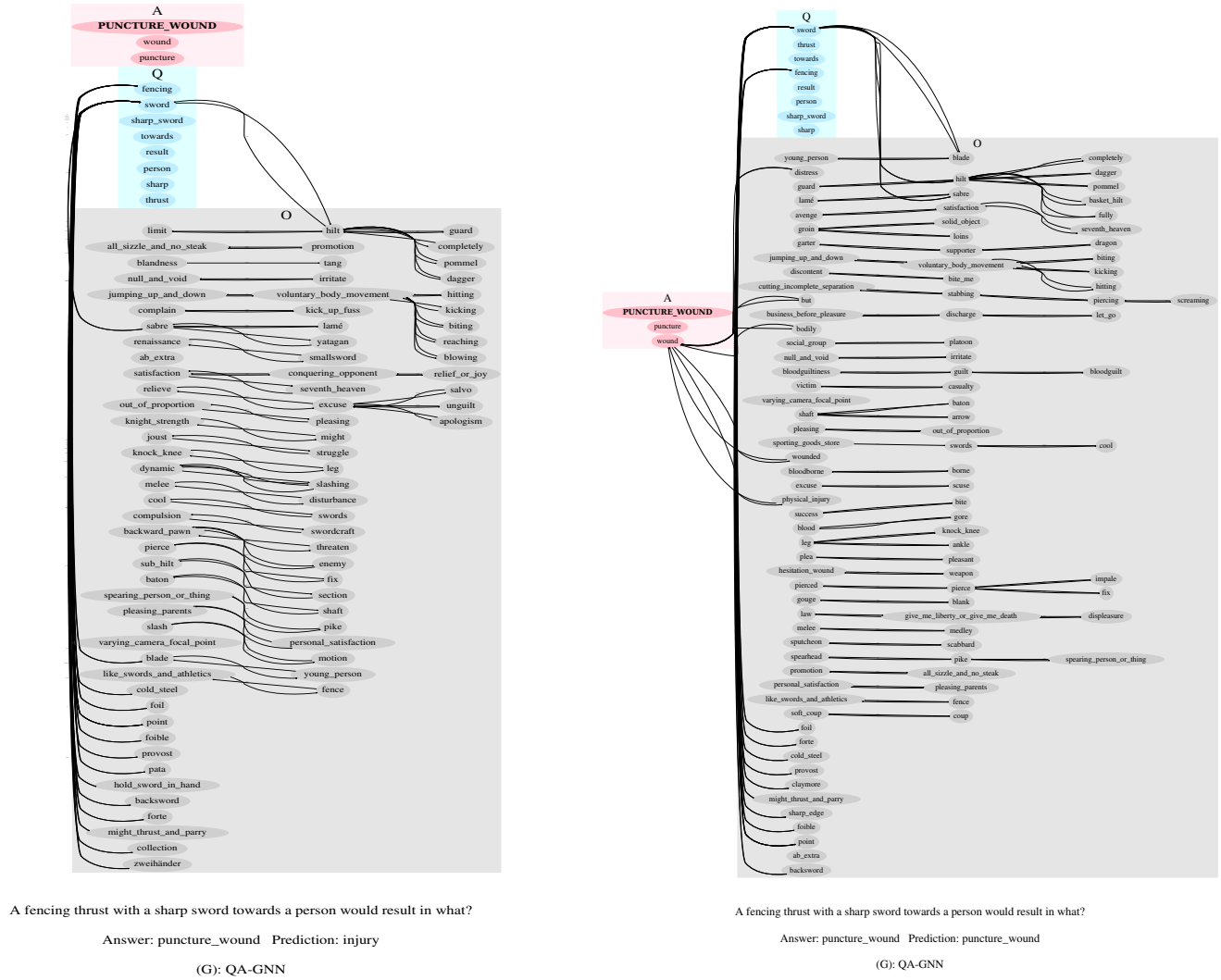
are more descriptive, and the extracted triplets demonstrate high relevance to both the question and answer context. Notably, there is a clear path connecting the question to the answer entities: (heifer, cows, dairy cattle, farmland, farm), leading to the answer "farmhouse." This path would have been missed if the graph extraction had been limited to a two-hop structure.

Overall, the retrieved graph exhibits closer contextual alignment with the question-answer choice. The empirical impact of these graphs is already demonstrated through the end-to-end training results, as shown in Table 1. It is important to note that although some of the retrieved graphs may appear disconnected, this issue is resolved in the QAGNN model by introducing a context node that connects to all retrieved nodes, allowing for effective message passing of information.

5.2.5 Open Domain v/s Multi-choice Graph Retrieval. In Figure 6, we illustrate the disparity between the graphs retrieved in the open domain and multiple-choice settings. Originally, the CSQA task is designed as a multiple-choice question answering task. However, in our proposed methodology, we also approached it as an open-domain question answering task. In the open domain setting, the retriever is provided only with the question statement and is not given any information about the answer choices. On the other hand, in the multiple-choice setting, the statement from the question-answer choice is used to extract relevant KG triplets. The difference in the extracted graphs is evident in the figure presented. Notably, in the multiple-choice setting, the triplets related to the answer entities are also retrieved in the triplets, whereas in the open-domain question answering (QnA) setting, this is not the case since there is no contextual information available regarding the answer choices.

6 CONCLUSION

In conclusion, our research unveiled a new learning framework (DrKG) designed specifically to extract optimal subgraphs from large, structured knowledge graphs, targeting the task of Commonsense Question Answering. This framework incorporated a unique method to represent knowledge graphs into textual form and then harnessing a dense retriever, trained to isolate relevant



(a) Open-Domain Retrieval **(b) Multiple Choice QnA**

Figure 6: Retrieved graph with Contriever model in Open Domain v/s Multiple choice QnA setting

subgraph components. To understand the impact of various elements in the pipeline, we conducted numerous experiments on the training of the graph dense retriever, varying the type of encoder architectures and pre-training techniques with different datasets. Empirical evidence illustrated that our retriever-extracted graphs performed notably better than heuristic graph baselines employed in the QA-GNN model. Through the consolidation of our quantitative findings and qualitative insights we can show the effectiveness of our approach in facilitating logical reasoning. Furthermore, this technique possesses generalizability and presents potential for extension to other sub-graph extraction problems within Knowledge Graph-related tasks.

Despite these promising results, we recognize the potential for further enhancement in our model's capabilities. One potential avenue for enhancement is to develop a mechanism to incorporate feedback into the graph retriever, integrating retriever training as part of the end-to-end network rather than as a separate framework. This design concept requires further investigation, particularly in terms of determining appropriate quantitative metrics for providing feedback to the retriever module. Exploring these possibilities constitutes an important aspect of our future work.

ACKNOWLEDGEMENTS

We thank members of the Stanford SNAP, P-Lambda, and NLP groups, as well as our anonymous reviewers for providing valuable feedback. This work was supported in part by DARPA under Nos. N660011924033 (MCS).

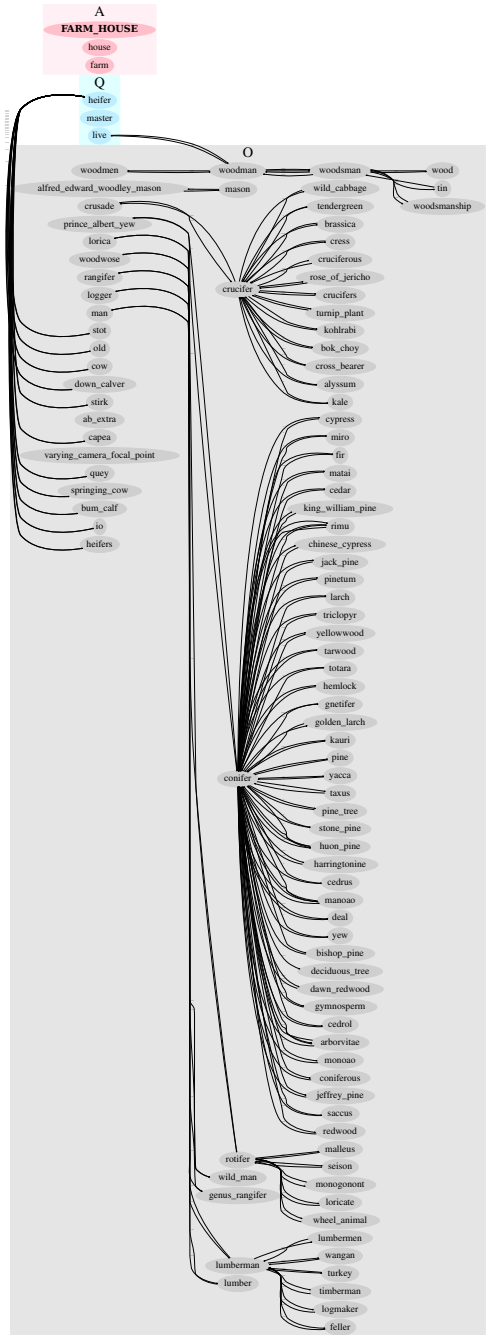
REFERENCES

[1] Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35.

12574–12582.
[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
[3] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. arXiv:1905.05733 [cs.CL]
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[5] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning Dense Representations for Entity Retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, Hong Kong, China, 528–537. https://doi.org/10.18653/v1/K19-1049
[6] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*. PMLR, 3929–3938.
[7] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652* (2017).
[8] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. https://doi.org/10.48550/ARXIV.2112.09118
[9] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.
[10] Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. 2022. GRAPE: Knowledge Graph Enhanced Passage Reader for Open-domain Question Answering. In *Findings of Empirical Methods in Natural Language Processing*.
[11] 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)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550
[12] 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 (2020), 9459–9474.
[13] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. 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)*. Association for Computational Linguistics, Hong Kong, China, 2829–2839. https://doi.org/10.18653/v1/D19-1282
[14] Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. RiddleSense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. *arXiv preprint arXiv:2101.00376* (2021).
[15] 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 preprint arXiv:1907.11692* (2019).
[16] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8449–8456.
[17] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789* (2018).
[18] 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*. Association for Computational Linguistics, Seattle, United States, 1535–1546. https://doi.org/10.18653/v1/2022.findings-naacl.115
[19] Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. 2020. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *Proceedings of the 13th international conference on web search and data mining*. 474–482.
[20] Yuanchun Shen. 2023. SRTK: A Toolkit for Semantic-relevant Subgraph Retrieval. arXiv:2305.04101 [cs.IR]
[21] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.
[22] Haitian Sun, Tania Bedrax-Weiss, and William W Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. *arXiv preprint arXiv:1904.09537* (2019).
[23] Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2021. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. *arXiv preprint arXiv:2112.02732* (2021).
[24] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937* (2018).
[25] Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021. Fusing Context Into Knowledge Graph for Commonsense Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 1201–1207. https://doi.org/10.18653/v1/2021.findings-acl.102
[26] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 535–546. https://doi.org/10.18653/v1/2021.naacl-main.45
[27] Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 4961–4974. https://doi.org/10.18653/v1/2022.acl-long.340
[28] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 5773–5784. https://doi.org/10.18653/v1/2022.acl-long.396
[29] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. GreaseLM: Graph REASoning Enhanced Language Models. In *International Conference on Learning Representations*.

A APPENDIX - QUALITATIVE ANALYSIS

Appendix with more some more qualitative examples.



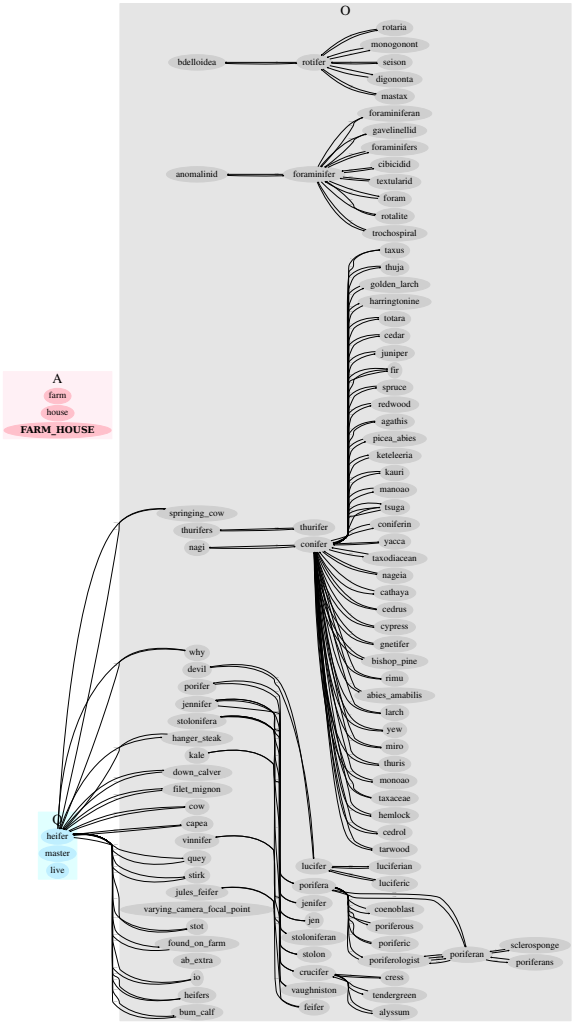
Where does a heifer's master live?

Answer: farm_house Prediction: farm_house

(G): QA-GNN

Figure 7: Retrieved graph with BERT encoder

Bert Encoder: For the question, "Where does a heifer's master live?" the top-5 relevant triplets retrieved with BERT encoder model are: (heifer, relatedto, capea), (live, relatedto, woodman), (heifer, relatedto, cow), (heifer, relatedto, quey), (woodsman, relatedto, wood), which are very relevant to the question choice. (Refer 7)



Where does a heifer's master live?

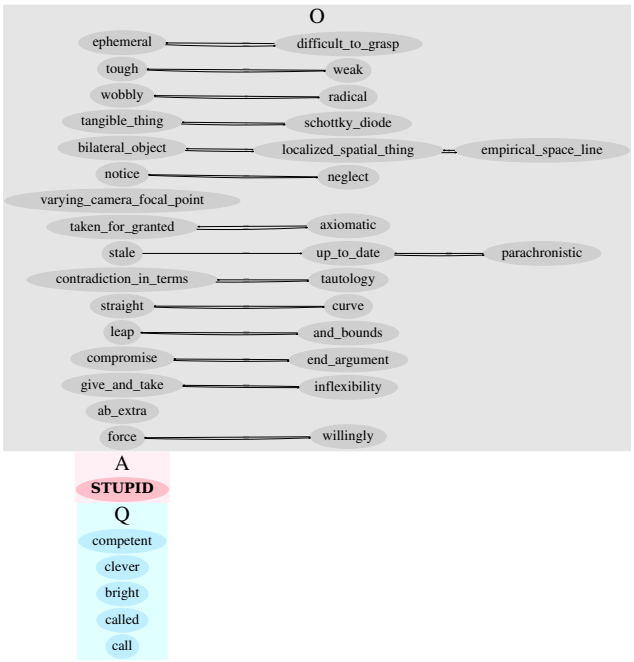
Answer: farm_house Prediction: farm_house

(G): QA-GNN

Figure 8: Retrieved graph with RoBERTa encoder

RoBERTa Encoder: For Question in figure 8, the extracted nodes appear to form contextual clusters related to "heifer," connecting to categories such as cattle, plants, and animals, which contribute to a more comprehensive context for deriving the answer "farmhouse".

Contriever Encoder: For the question in 9, in the first question, we find the entity "difficult to grasp," which is directly related to the question about bright and stupid people. These descriptive nodes are typically missing in in the BERT and RoBERTa retrieved graphs.



What is someone who isn't clever, bright, or competent called?

Answer: stupid Prediction: stupid

(G): QA-GNN

Figure 9: Retrieved graph with Contriever encoder