Knowledge Fusion and Semantic Knowledge Ranking for Open Domain Question Answering

Pratyay Banerjee and **Chitta Baral**
Department of Computer Science, Arizona State University
`pbanerj6,chitta@asu.edu`

Abstract

Open Domain Question Answering requires systems to retrieve external knowledge and perform multi-hop reasoning by composing knowledge spread over multiple sentences. In the recently introduced open domain question answering challenge datasets, QASC and OpenBookQA, we need to perform retrieval of facts and compose facts to correctly answer questions. In our work, we learn a semantic knowledge ranking model to re-rank knowledge retrieved through Lucene based information retrieval systems. We further propose a "knowledge fusion model" which leverages knowledge in BERT-based language models with externally retrieved knowledge and improves the knowledge understanding of the BERT-based language models. On both OpenBookQA and QASC datasets, the knowledge fusion model with semantically re-ranked knowledge outperforms previous attempts.

1 Introduction

Open Domain Question Answering is one of the challenging Natural Language Question Answering tasks where systems need to retrieve external knowledge and perform multi-hop reasoning by understanding knowledge spread over multiple sentences. In recent years, several open domain natural language question answering datasets and challenges have been proposed. These challenges try to replicate the human question answering setting where humans are asked to answer questions and are allowed to refer to books or other information sources available to them. Datasets such as HotPotQA (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019), MultiRC (Khashabi et al., 2018b), ComplexWebQuestions (Talmor and Berant, 2018) and WikiHop (Welbl et al., 2018) require finding relevant knowledge and reasoning over multiple sentences. In these tasks, the systems

Figure 1: An example from the QASC Dataset. The highlighted words are the keywords present in the Question, Annotated Fact 1 and Annotated Fact 2. Annotated Facts 1 and 2 are present in the associated knowledge corpus. These are the gold knowledge facts that need to be retrieved and composed to answer the question. The correct answer is highlighted in blue.

are not constrained to any pre-determined knowledge bases. Both, the task of finding knowledge and reasoning over multiple sentences requires deep natural language understanding. The goal of these tasks is not to memorize the texts and facts, but to be able to understand and apply the knowledge to new and different situations (Jenkins, 1995; Mihaylov et al., 2018).

In our work, we focus on the datasets, OpenBookQA and QASC (Mihaylov et al., 2018; Khot et al., 2019a). They differ from the above mentioned datasets in the following aspects. Special care is taken in OpenBookQA and QASC to avoid simple syntactic cues in questions, that allow decomposition into simpler queries (Khot et al., 2019a). It has been shown through human verification that answering questions in both of these datasets requires a composition of two or more facts. OpenBookQA is accompanied by an open-book of facts, which contains partial knowledge to answer the questions. QASC provides a knowledge

corpus of around 19 million science facts.

In OpenBookQA, to answer questions systems need to combine a seed core fact from the provided open-book with an unknown number of additional facts and an unspecified source for retrieving such additional facts. Questions in QASC require retrieval of exactly two facts from the provided knowledge corpus. Moreover, it provides annotation for which facts are required, allowing the building of a supervised knowledge retrieval model.

There are several challenges we need to address while solving these datasets. The first challenge is the task of relevant knowledge retrieval. We use a simple Lucene based information retrieval system Elasticsearch (Gormley and Tong, 2015) to retrieve knowledge from an appropriate knowledge corpus. The task of retrieval is done for each answer options. Moreover, we require multiple steps during retrieval as it has been mentioned above, the questions need composition of knowledge in multiple sentences to correctly answer the questions. A multi-step information retrieval introduces a significant amount of noise. We address this challenge by learning a BERT-based Semantic Knowledge Ranking model.

The second challenge we address is the creation of the dataset for Semantic Knowledge Ranking. In order to train our Semantic Knowledge Ranking model, we need knowledge facts with both positive and negative labels. For this task, QASC, OpenBookQA and SciTail (Khot et al., 2018) provide positive labels, i.e, the knowledge facts which are relevant to a given question-answer pair. We address the challenge of negative label creation using our automatic dataset preparation techniques. The design choices for the dataset preparation are demonstrably impactful on the downstream question answering task.

The third challenge we address is about knowledge composition and understanding. BERT-based language models possess considerable knowledge learned through their pre-trained language modelling tasks. The challenge of incorporating external domain specific knowledge, discriminative enough to answer questions is addressed by our knowledge fusion module over BERT-based language models. We identify the drawbacks in the current way systems use BERT-based language models, and augment them with our new knowledge fusion module.

Finally, we analyse both our knowledge rank-

ing and question answering models to identify how different components contribute. We also analyze what noise our knowledge ranking model introduces, where our question answering model fails and why is it unable to answer such questions. Our analysis shows some of the drawbacks of using Attention-based language models and the necessity of improvements in specific components.

Our contributions are summarized below:

- We prepare a Semantic Knowledge Ranking dataset by using annotations from three different sources and with our dataset preparation techniques. We also evaluate two BERT-based ranking model on this task.
- We propose a new model to perform better knowledge composition and question answering with external knowledge.
- We analyze our knowledge ranking and knowledge composition models to better understand the failures to enable future improvements.
- We achieve new state-of-the-art results in both OpenBookQA (+2.2%) and QASC (+7.28%).

2 Related Work

Text based QA: Multiple datasets have been proposed in recent years for natural language question answering such as (Rajpurkar et al., 2016; Joshi et al., 2017; Khashabi et al., 2018a; Richardson et al., 2013; Lai et al., 2017; Reddy et al., 2018; Choi et al., 2018; Tafjord et al., 2018; Mitra et al., 2019b; Sap et al., 2019; Bisk et al., 2019). There are several attempts to solve these challenges such as (Devlin et al., 2019; Vaswani et al., 2017; Seo et al., 2016; Liu et al., 2019) etc. In our work we focus on QASC (Khot et al., 2019a) and OpenBookQA (Mihaylov et al., 2018) datasets, which are multiple-choice question answering datasets which need multi-hop reasoning and knowledge retrieval from provided knowledge corpus.

External Knowledge: Closest to our work are the models which use external knowledge, such as in the work of (Khot et al., 2019b,a; Pirtoaca et al., 2019; Banerjee et al., 2019; Mitra et al., 2019a) where they use sentences as knowledge. We differ to these with our knowledge ranking and infusion models. Other models such as in the work of (Mihaylov and Frank, 2018; Chen et al., 2018; Yang et al., 2019; Wang and Jiang, 2019) embed syntactic or semantic knowledge into knowledge enriched embeddings. They do not use additional sen-

Figure 2: Examples of questions present in OpenBookQA and QASC. The source of facts for QASC is the available knowledge corpus. For OpenBookQA, the Fact 1 is present in the open-book, but the Fact 2 needs to be retrieved from an external source. Correct answers are highlighted in Blue.

tences as knowledge. Sentences lack a well-defined structure and possess a larger number of variables. Semantic knowledge retrieval is also present in a different form in the above mentioned work. The task of neural explanation retrieval or ranking such as in (Jansen and Ustalov, 2019; Banerjee, 2019) is similar to semantic knowledge retrieval. We differ from them in the way we model our tasks and our dataset preparation techniques.

3 Datasets

Open Book Question Answering: Open-BookQA is a multiple-choice school level science question answering dataset from AllenAI. There are four answer options for each question. The dataset provides an accompanying open-book of 1,324 science facts. There are a total of 4,957 questions in train and 500 questions in each validation and test set. The nuance of this dataset is that it needs knowledge retrieval and knowledge composition to answer the questions. It is specified that one of the facts is present in the accompanying open-book and we need additional knowledge sources to retrieve the other facts.

Question Answering via Sentence Completion: QASC is another multiple-choice science question answering dataset from AllenAI. There are 9,980 8-way multiple-choice questions from elementary and middle school level science. There are 900 questions in the validation and 920 questions in the test set. It is different from OpenBookQA as it

is accompanied by a knowledge corpus of 19 million science facts. Moreover to answer each question we need to combine exactly two facts, both of which are present in the knowledge corpus. QASC provides additional annotations of which two facts are required to answer the questions. These annotations enable us to create a semantic knowledge ranking dataset. Figure 2 shows an example from both of the datasets.

4 Approach Overview

Our approach includes the following modules: Multi-step Knowledge Retrieval, Semantic Knowledge Ranking Dataset Preparation, Semantic Knowledge Ranking and Knowledge Fusion based Question Answering.

In **Multi-step Knowledge Retrieval** we want to extract relevant facts from the knowledge corpus. We use heuristic-based query creation techniques. In the first step of the retrieval, we use the question Q and the answer option A_i to generate queries. In the second step, we use question Q, the answer option A_i and F_1, the top ten facts retrieved and ranked from step one, to generate queries.

In **Semantic Knowledge Ranking**, we re-rank our retrieved knowledge sentences using BERT-based language models. To train our models, we do **Semantic Knowledge Ranking Dataset Preparation** by using the provided annotations present in the QASC, OpenBookQA and SciTail datasets. But the annotations only provide positive labels, so we use our dataset preparation techniques to

automatically generate negative labels.

We train a BERT-based language model for question answering using external knowledge and identify the gaps in knowledge understanding. We analyze the results and define a **Knowledge Fusion** module. The module is based on a simple intuition that the model should have the ability to compare against different answer options and the input should clearly distinguish between the knowledge which is ranked higher across all answers.

5   Multi-Step Knowledge Retrieval

Knowledge Sources: We use the open-book from OpenBookQA, the knowledge corpus from QASC and the ARC knowledge Corpus (Clark et al., 2018) as our knowledge sources. There are 1324 knowledge facts in the open-book from Open-BookQA. There are 19 million knowledge facts in the knowledge corpus of QASC. There are 1.7 billion knowledge facts in ARC knowledge corpus. We keep the knowledge facts as is for the open-book and QASC knowledge corpus. We preprocess the knowledge facts from ARC by removing non-English characters and remove punctuations that do not have an impact on sentence structure. We index the factual sentences into Elasticsearch, a Lucene based search engine.

Step-1:   In the first step of knowledge retrieval, we have a question Q and the ith answer options A_i. We generate the query using a simple heuristic of concatenating the question and answer, and removing stop-words but keeping the order of the words intact. We use stop-words from NLTK and standard python "stop-words" library. We query the Elasticsearch IR system and retrieve top-50 sentences. These sentences are denoted as F_1.

Step-2:   The second step of knowledge retrieval is similar to the work of heuristic-based natural language abduction in (Banerjee et al., 2019) and the 2-step retrieval mentioned in (Khot et al., 2019a). For each question Q, answer option A_i and top ten knowledge retrieved and semantically ranked F_{1ij} we find the set of unique words present in Q, A_i and F_1ij using the following heuristics:

$$Qu_{ij} = ((Q \cup A_i) \cup F_{1ij}) \setminus ((Q \cup A_i) \cap F_{1ij})$$

where Qu is the generated query, ith answer option, and jth retrieved sentence from F_{1i}. These sentences are denoted as F_2.

6   Semantic Knowledge Ranking

6.1   Task Definition

We model the semantic knowledge ranking task as a binary sentence-pair classification task, i.e, given the question Q and answer option A_i, we classify the corresponding retrieved knowledge F_{kij} into two classes, irrelevant and relevant, where k is the IR step, i is the answer option number and j is the retrieved sentence number. We rank the sentences using the class probabilities for the relevant class.

Formally, we learn the following probability:

$$Rel(F_{kij}, Q, A_i) = P(F_{kij} \in G | Q, A_i) \quad (1)$$

where G is the set of relevant facts.

We compare multiple task settings for Knowledge Ranking, such as a regression task similar to Semantic Textual Similarity. Though this task is more appropriate as a ranking task, it is harder to get correct and noiseless annotations for such a task using automatic techniques.

6.2   Dataset Preparation

For the classification task, we require both relevant and irrelevant facts given a question and answer pair, which is simpler to achieve automatically.

Positive Labels: Questions in QASC are accompanied by two human-annotated knowledge facts, which when composed can be used to answer the questions. These annotated facts provide us the positive labels. We gather more positive labels from the OpenBookQA datasets. The OpenBookQA also has an additional resource where they provide which fact from the open-book is the most relevant. The final source of positive labels is the SciTail dataset (Khot et al., 2018). SciTail contains questions, the correct answer, and a sentence pair. The task in SciTail is natural language inference, i.e, does the fact entails or is neutral to the hypothesis (question-answer). The hypothesis is not a concatenated version of the question and an answer but a well-structured sentence. Unfortunately in our target application, we do not possess a well-structured sentence equivalent to our question-answer pairs. For positive labels, we select question-answer pair and the corresponding premise as the relevant facts from the samples annotated as "entails".

Negative Labels: From SciTail we take all samples marked as "neutral" as an initial set of irrelevant facts. We gather further negative samples using the following simple heuristics. For

all question-answer pairs from QASC and Open-BookQA we do an initial knowledge retrieval using the query generation as mentioned in step one of multi-step knowledge retrieval. From this set of retrieved facts we identify similar facts to the gold relevant facts using Spacy (Honnibal and Montani, 2017) document similarity and mark them as irrelevant, i.e, sentences retrieved using wrong answers similar to gold whose similarity is less than a threshold of T. T is identified using manual evaluation of 100 samples. Spacy uses word vectors to compute document similarity. The similarity function used is cosine similarity. We also mark the knowledge retrieved using wrong answer options as irrelevant. Note that we never mark the gold annotated relevant facts as irrelevant for the wrong answer options, as we want our Semantic Knowledge Ranking model to rank these facts better even for the wrong answer options. Train and validation sets are balanced for both classes, with each having 145,200 and 16,134 samples respectively.

6.3 Model Description

We evaluate two BERT-based language models, BERT-large-cased (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) for the sentence-pair classification task.

We provide the concatenated question and answer pair as sentence A and the fact as sentence B. Let S_A denote tokens from sentence A and S_B denote tokens from sentence B, then the input to the BERT model is defined as $\{[CLS]S_{A_i}[s]S_{B_j}[s]\}$. Here $[s]$ is the separator token. $[CLS]$ is the class token. We take the encoding of $[CLS]$ token from the final layer of BERT, which we pass through a feed-forward layer of dimension $H_{dim} \times 2$, where H_{dim} is the dimensions of the encoding and a final softmax layer for getting probabilities for the two classes. We use cross-entropy loss between the predicted scores and the gold relevance labels. [1]

$$logits = FF(\beta(CLS, S_A, S_B))$$

$$score(F_{kij}, Q, A_i) = softmax(logits)$$

β is the BERT model, FF is the feedforward layer.

7 Knowledge Fusion Question Answering Model

Overview of BERT-based QA Models: Let Q define the set of tokens in the question, A_i the set

---

[1]More training details is in Supplementary Materials.

of tokens in the ith answer option and K_i, the retrieved knowledge. The current systems (Devlin et al., 2019; Liu et al., 2019; Banerjee et al., 2019; Khot et al., 2019a) use the following way of defining the input to the BERT-based models. For an answer, they append the question tokens Q to the knowledge tokens K_i. The total input is defined as $\{[CLS] K_i Q [SEP] A_i [SEP]\}$. This can be interpreted as an entailment model where, given the knowledge and question, we predict the entailment score of each answer. Each answer has a separate input. This way of creating input and modeling question answering has certain drawbacks.

Firstly, each knowledge retrieved is unique to the corresponding answers. BERT has multiple layers of stacked attention neural units, and attention with the corresponding knowledge enables the system to perform the question answering task. But the knowledge retrieved uses the question and answer tokens; consequently, there is a lot of similarity between the knowledge, question and answer tokens. This though helpful in answering also introduces noise and confusion.

Secondly, the input does not enable the comparison of different answers using attention layers. Cross-answer attention is needed to answer some questions which are comparative in nature. For example, a sample question from OpenBookQA:

> **Question:** *Owls are likely to hunt at?*
> a) 3:00 PM b) **2:00 AM** c) 6:00 PM d) 7:00 AM

Entailment models may falter in such comparative questions, as they do not compare different answer options, and are only aware of one answer option at a time.

Finally, the set and order of facts retrieved for each answer is unique for an answer, but there may be sentences retrieved which are common to all the answer options. These sentences are relevant to the question, and attention with these facts with all the answer options together should enable the model to be able to discriminate between the correct and incorrect answer options. We use the above insights to develop our input and knowledge fusion modules.

Input Description: We first categorize the retrieved facts into two classes. Let C denote the set of facts which are present in the knowledge retrieved for each answer option. Let U_i denote the set of facts which is unique to an answer option. While creating the sets we maintain the order of
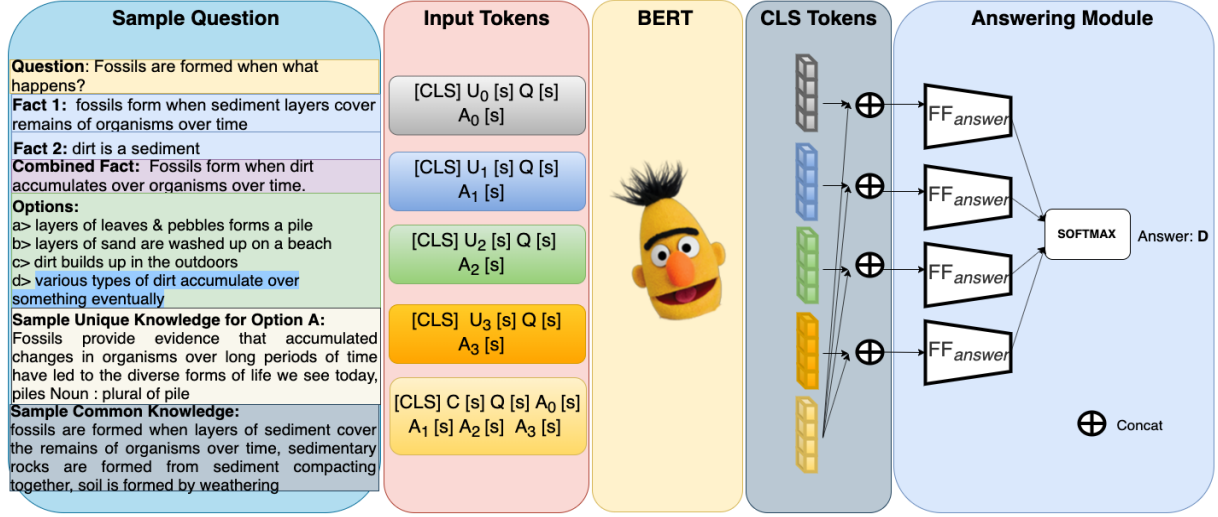
Figure 3: Our end-to-end model view with sample question, input description and knowledge fusion module. The correct answer is highlighted in blue. The sample knowledge are retrieved from OpenBookQA and our knowledge corpus, consisting of ARC Corpus, OpenBookQA open-book and QASC knowledge corpus. The sample knowledge are semantically ranked.

the facts, after retrieval and semantic knowledge ranking. For creating C, we count the appearance of each sentence across different U_i and multiply the max score for this sentence from the ranking model. We sort the sentences in decreasing order of this score. We build our inputs as follows.

We concat the unique knowledge U_i to the question similar to the above mentioned input to the entailment model . This input is the "per answer option input" defined as $\{[CLS_{A_i}] \, U_i \, Q \, [s] \, A_i \, [s]\}$.

We create another single input in which we concat the question, all the answer options and the common knowledge C. The "common input" is defined as $\{[CLS_C] \, Q \, [s] \, A_1 \ldots [s] \, A_4 \, [s] \, C \, [s]\}$.

For a four-way multiple choice question, we give five total inputs to BERT. Similarly for an eight-way multiple choice question, we give nine inputs.

Model Description: We feed each "per answer input" to BERT-based language models. We also feed the "common input" to the BERT-based language models. We take the corresponding embeddings of the class token, $[CLS_{A_i}]$, for each answer specific input, and concatenate the class token, $[CLS_C]$ from the "common input" . Let V be the final concatenated vector.

$$CLS_{A_i} = \beta(CLS, U_i, Q, A_i)$$
$$CLS_C = \beta(CLS, C, Q, A_{1..4})$$
$$V = [CLS_{A_i} : CLS_C]$$

We feed this knowledge enriched vector V through a two-layer feedforward network with an intermediate GeLU activation (Hendrycks and Gimpel, 2016) and BertLayerNorm (Devlin et al., 2019). We have a final softmax layer to get probabilities.

$$logits_i = FF2(BNorm(GeLU(FF1(V))))$$
$$score(Q, A_i, U_i, C) = softmax(logits_i)$$

where, $FF1$ and $FF2$ are feedforward networks, $BNorm$ is the BertLayerNorm and $GeLU$ is the GeLU unit. We train the model using cross-entropy loss between predictions and gold answer labels. [2]

8 Experimental Results and Analysis

For Semantic Knowledge Ranking, we compare with a baseline of simple Information Retrieval using Elasticsearch. For Question Answering, we compare with a baseline of RoBERTa with and without semantically ranked knowledge. Each model is trained with a hyperparameter budget of ten runs (Dodge et al., 2019), and the mean of the accuracies are reported.

Semantic Knowledge Ranking: Table 1 shows the accuracy of the BERT-based semantic knowledge ranking model. Table 1 shows the impact of knowledge ranking on the precision of retrieval of gold annotated knowledge facts for OpenBookQA and QASC validation set. We compare the impact of knowledge ranking in both the steps of Multi-Step Knowledge Retrieval. We can observe that Semantic Knowledge Ranking considerably

---

[2]More training details is in Supplementary Materials.

Model	Accuracy % ↑	Dataset	Step 1 of Retrieval (%) ↑			Step 2 of Retrieval (%) ↑		
			F1 F2 : R@5	F1 F2 : R@10	F1 & F2 : R@10	F1 F2 : R@5	F1 F2 : R@10	F1 & F2 : R@10
Simple IR	N/A	OpenBook	33.50 N/A	42.60 N/A	N/A	N/A N/A	N/A N/A	N/A
		QASC	34.32 11.45	47.54 14.78	08.12	33.18 18.78	47.78 24.56	11.34
BERT	88.32	OpenBook	54.60 N/A	65.80 N/A	N/A	N/A N/A	N/A N/A	N/A
		QASC	46.80, 22.50	51.80 29.67	14.44	48.60 27.85	50.30 29.33	15.88
RoBERTa	91.56	OpenBook	**59.62** N/A	**79.60** N/A	N/A	N/A N/A	N/A N/A	N/A
		QASC	49.32 28.38	55.80 <u>31.35</u>	15.56	**51.40 32.56**	**57.68 35.35**	**19.75**

Table 1: Results for Semantic Knowledge Ranking. F_1 and F_2 represent the two core knowledge facts. Accuracy is the classification accuracy of the classifiers on validation set. Recall@N (R@N) is the measure of the fact being present in the top N retrieved sentences. F_1 & F_2 represent both the facts are present in the top 10. For OpenBookQA we do not have annotations for gold F_2. Best scores are marked in Bold.

Base Model: RoBERTa + RACE	
Module	Dev Acc (Δ) % ↑
No Knowledge	59.40
Step 1	62.40 (+3.0)
Step 1 + SKR	66.70 (+4.3)
Step 1 + KF	70.50 (+3.8)
Step 1 + KF + SKR	74.60 (+4.1)
Step 2	82.50 (+7.9)
Step 2 + SKR	83.90 (+1.4)
Step 2 + KF	84.20 (+0.3)
Step 2 + KF + SKR	85.20 (+1.0)

Table 2: Ablation studies on the QASC dataset. Step 1 and 2 correspond to different steps of Multi-step Knowledge Retrieval. SKR is semantic knowledge ranking. KF is our knowledge fusion model. 900 samples in the Validation set.

Model	Acc % ↑
Human	91.7
Sun et al.	56.0
Microsoft BERT MT	64.0
Pan et al.	70.0
AristoBERTv7	72.0
Banerjee et al.	72.0
AllenAI	77.8
RoBERTa + Step2	76.4
Ours: Step2+SKR+KF	**80.0**

Table 3: OpenBookQA test set comparison of different models. The current state-of-the-art is AllenAI's AristoRoBERTaV7, which is a no knowledge RoBERTA model pre-trained on multiple datasets. Our model is with semantic knowledge ranking & knowledge fusion.

improves the precision, notably P@10 of F_2 facts for single-step retrieval.

Ablation Studies: Table 2 shows the impact of each of our components on the accuracy in the QASC dataset. We add our modules over the base model of RoBERTA pre-trained on RACE (Lai et al., 2017). We observe that the task of QASC needs external knowledge, as the accuracy of no knowledge model is relatively low. Each of our modules, contribute to the overall increase in accuracy. Semantic knowledge ranking improves the accuracy of Step 1 by a large margin. So does the multi-step knowledge retrieval. Both the techniques have the same effect of increase in P@10 of F_1 and F_2, bringing more relevant facts for the model to answer correctly. Our Knowledge Fusion Model further improves accuracy, showing we can utilize knowledge more effectively. The increase in accuracy from knowledge fusion is especially significant using the facts retrieved from Step 1.

OpenBookQA and QASC Results: Table 3 and 4 compares our best model to the previous work on OpenBookQA and QASC. For QASC, we compare with our stronger RoBERTa baselines. Figure 4,
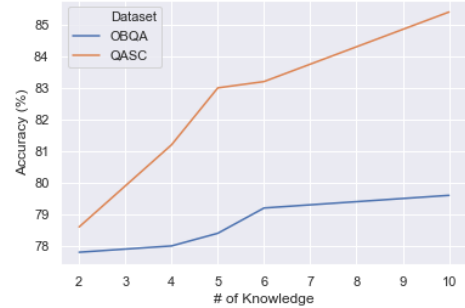

Figure 4: Impact of knowledge on the question answering task on the respective validation sets. More than 10 is limited by BERT max token length.

shows the validation accuracy versus the number of facts retrieved. As we can see, increasing the facts increases the accuracy as more appropriate facts are seen by the model.

Model Analysis: Figure 5 shows our model is more confident when it predicts the correct answer compared to when it predicts the wrong answer. QASC has eight answer options, which increases the presence of distracting facts, and confusing answer options. Since our models use attention, and use statistical correlation, even though we re-

	Model	Dev Accuracy	Test Accuracy
	Human	-	93.00
Khot et al.	Random	12.50	12.50
	BERT+ Step 1	69.50	62.60
	BERT+ Step 2	72.90	68.48
	BERTwwm + Step 2	78.00	<u>73.15</u>
Ours	RoBERTA+ Step 1 + KF + SKR	74.60	68.80
	RoBERTA+ Step 2	82.50	79.28
	RoBERTA+ Step 2 + KF	84.20	80.11
	RoBERTA+ Step 2 + SKR + KF	85.20	**80.43**

Table 4: QASC validation and test set comparison between the baselines, the proposed models in Khot et al. and our models. We compare our models with a stronger baseline of RoBERTA with knowledge. The best score is marked in bold. Previous best is underlined. BERT : BERT Large, BERTwwm : BERT Large whole word masked, RoBERTA : RoBERTA large.

Figure 5: Distribution of prediction confidence of the Best Model for QASC Validation set.

trieve relevant facts, the model predicts the answers which have the highest correlation with the facts. Our approach is to improve semantic ranking and knowledge composition to push the quality of knowledge in each step, which leads improving accuracy. Although using attention-based models for ranking brings facts that attend to all question-answer pairs, our knowledge fusion model is able to understand the appropriate knowledge. Our analysis shows that, the input creation algorithm for knowledge fusion model acts as another source for knowledge ranking, and the "common input" enables the model to distinguish between answers.

Errors in Semantic Knowledge Retrieval: The semantic knowledge ranking model has comparatively high accuracy (91.56%). The errors are mostly false positives, i.e, classification of question-wrong answer and corresponding retrieved facts using the wrong answer. These are the sources of noise for the downstream question answering task. The alternative of only using the question to rank knowledge retrieved performs even worse. This indicates semantic knowledge ranking needs question-answer pair, and is a challenging task.

Errors in Question Answering: We analyzed the 100 errors made in OpenBookQA and the 137 errors made in QASC. We can broadly classify the errors made in question answering into four categories: Answering needs Complex Reasoning, Confusing fact is Retrieved, Knowledge Retrieval Failure, and Knowledge Composition Failure. There are few examples in OpenBookQA where more complex reasoning such as Temporal, Qualitative, Conjunctive and Negation is required. Following

is an example of Conjunctive Reasoning:[3]

> **Question:** Which pair don't reproduce the same?
> (A) rabbit and hare (B) *mule and hinny* (C) **cat and catfish** (D) caterpillar and butterfly

> **Question:** Astronomy can be used for what?
> (A) *Communication* (B) safe operation (C) vision (D) homeostasis (E) gardening (F) cooking (G) **navigation** (H) architecture
> **Fact Retrieved:** what is radio astronomy. a radio is used for communication.

Above is an example from QASC, where a confusing fact is retrieved that is semantically related to the question, but supporting the wrong answer; leading the model to perform an incorrect multi-hop reasoning.

Knowledge Retrieval Failure corresponds to 72% of the total errors in OpenBookQA. In QASC, out of 137 errors, 52 had correct F_1, 40 had correct F_2 and 25 had both F_1 and F_2 in the top ten. These errors can be mitigated by better retrieval and composition. Improving attention to perform better context-dependent similarity should enable models to distinguish between relevant and irrelevant facts.

9 Conclusion and Future Work

In this work, we have pushed the current state-of-the-art by 2.2% on OpenBookQA and 7.28% on QASC, two tasks that need external knowledge and knowledge composition for question answering. Our Semantic Knowledge Ranking and Knowledge Fusion question answering model over the BERT-based language model demonstrably improves the performance on OpenBookQA and QASC. We also provide a dataset to learn Semantic Knowledge Ranking using the annotations present in QASC, OpenBookQA, and SciTail. We have analyzed the performance of the components in our QA system. Our analysis shows the need to further improve knowledge ranking and knowledge composition.

---

[3]Correct Answers are in Bold. Predicted Answers are in italics.

References

AllenAI. 2019. Aristorobertav7. In *AristoRoBER-TAv7*.

Pratyay Banerjee. 2019. ASU at TextGraphs 2019 shared task: Explanation ReGeneration using language models and iterative re-ranking. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 78–84, Hong Kong. Association for Computational Linguistics.

Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, Florence, Italy. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.".

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Peter Jansen and Dmitry Ustalov. 2019. TextGraphs 2019 shared task on multi-hop inference for explanation regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 63–77, Hong Kong. Association for Computational Linguistics.

Tony Jenkins. 1995. *Open Book Assessment in Computing Degree Programmes*. Citeseer.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018a. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 252–262.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018b. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *NAACL*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2019a. Qasc: A dataset for question answering via sentence composition. *arXiv preprint arXiv:1910.11473*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2019b. Whats missing: A knowledge gap guided approach for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2807–2821.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794. Association for Computational Linguistics.

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*.

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. In *EMNLP*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019a. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *arXiv preprint arXiv:1909.08855*.

Arindam Mitra, Peter Clark, Oyvind Tafjord, and Chitta Baral. 2019b. Declarative question answering over knowledge bases containing natural language text with answer set programming. *arXiv preprint arXiv:1905.00198*.

Xiaoman Pan, Kai Sun, Dian Yu, Heng Ji, and Dong Yu. 2019. Improving question answering with external knowledge. *arXiv preprint arXiv:1902.00993*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

George Sebastian Pirtoaca, Traian Rebedea, and Stefan Ruseti. 2019. Answering questions by learning to rank-learning to rank by answering questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2531–2540.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2018. Improving machine reading comprehension with general reading strategies. *CoRR*, abs/1810.13441.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2018. Quarel: A dataset and models for answering questions about qualitative relationships. *arXiv preprint arXiv:1811.08048*.

A. Talmor and J. Berant. 2018. The web as a knowledge-base for answering complex questions. In *North American Association for Computational Linguistics (NAACL)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Chao Wang and Hui Jiang. 2019. Explicit utilization of general knowledge in machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2263–2272.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

A Appendices

B Supplemental Material

B.1 Training Semantic Knowledge Ranking

We use the HuggingFace (Wolf et al., 2019) and Pytorch Deep learning framework (Paszke et al., 2019). We will make our code available at Github Code Link: xx.github.com.

Model Training We train the models with learning rates in the range [1e-6,5e-5], batch sizes of [16,32,48,64], linear weight-decay in range [0.001,0.1] and warm-up steps in range of [100,1000].

B.2 Knowledge Fusion Question Answering

Model Training We train the model with following hyperparameters, learning rates in the range [1e-6,5e-5], batch sizes of [16,32,48,64], linear weight-decay in range [0.001,0.1] and warm-up steps in range of [100,1000].