QASC: A Dataset for Question Answering via Sentence Composition

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen,[+] Ashish Sabharwal
Allen Institute for AI, Seattle, WA, U.S.A.
[+]University of Arizona, Tucson, AZ, U.S.A.
{tushark, peterc, michalg, ashishs}@allenai.org, pajansen@email.arizona.edu

Abstract

Composing knowledge from multiple pieces of texts is a key challenge in multi-hop question answering. We present a multi-hop reasoning dataset, **Q**uestion **A**nswering via **S**entence **C**omposition (QASC), that requires retrieving facts from a large corpus and composing them to answer a multiple-choice question. QASC is the first dataset to offer two desirable properties: (a) the facts to be composed are annotated in a large corpus, and (b) the decomposition into these facts is not evident from the question itself. The latter makes retrieval challenging as the system must introduce new concepts or relations in order to discover potential decompositions. Further, the reasoning model must then learn to identify valid compositions of these retrieved facts using commonsense reasoning. To help address these challenges, we provide annotation for supporting facts as well as their composition. Guided by these annotations, we present a two-step approach to mitigate the retrieval challenges. We use other multiple-choice datasets as additional training data to strengthen the reasoning model. Our proposed approach improves over current state-of-the-art language models by 11% (absolute). The reasoning and retrieval problems, however, remain unsolved as this model still lags by 20% behind human performance.

Question: Differential heating of air can be harnessed for what?
(A) electricity production (D) reduce acidity of food
(B) erosion prevention ...
(C) transfer of electrons ...

Annotated facts:
f_S: Differential heating of air produces wind.
f_L: Wind is used for producing electricity.
Composed fact f_C: Differential heating of air can be harnessed for electricity production.

Figure 1: A sample 8-way multiple choice QASC question. Training data includes the associated facts f_S and f_L shown above, as well as their composition f_C. The term wind connects f_S and f_L, but appears neither in f_C nor in the question. Further, decomposing the question relation *"harnessed for"* into f_S and f_L requires introducing the new relation *"produces"* in f_S. The question can be answered by using broad knowledge to compose these facts together and infer f_C.

1 Introduction

Several multi-hop question-answering (QA) datasets have been proposed to promote research on multi-sentence machine comprehension. On one hand, many of these datasets (Mihaylov et al. 2018; Clark et al. 2018; Welbl, Stenetorp, and Riedel 2018; Talmor and Berant 2018) do not come annotated with sentences or documents that can be combined to produce an answer. Models must thus learn to reason without direct supervision. On the other hand, datasets that come with such annotations involve either single-document questions (Khashabi et al. 2018a) leading to a strong focus on coreference resolution and entity tracking, or multi-document questions (Yang et al. 2018) whose decomposition into simpler single-hop queries is often evident from the question itself.

We propose a novel dataset, **Q**uestion **A**nswering via **S**entence **C**omposition (QASC; pronounced kask) of 9,980 multi-hop multiple-choice questions (MCQs) where simple syntactic cues are insufficient to determine how to decompose the question into simpler queries. Fig. 1 gives an example, where the question is answered by decomposing its main relation *"harnessed for"* (in f_C) into a similar relation *"used for"* (in f_L) and a newly introduced relation *"produces"* (in f_S), and then composing these back to infer f_C.

While the question in Figure 1 can be answered by composing the two facts f_S and f_L, that this is the case is unclear based solely on the question. This property of relation decomposition not being evident from reading the question pushes reasoning models towards focusing on learning to compose new pieces of knowledge, a key challenge in language understanding. Further, f_L has no overlap with the question, making it difficult to retrieve it in the first place.

Let's contrast this with an alternative question formulation: "What can something *produced by* differential heating of air be *used for*?" Although awkwardly phrased, this variation is easy to syntactically decompose into two simpler

Property	CompWebQ	DROP	HotPotQA	MultiRC	OpenBookQA	WikiHop	**QASC**
Supporting facts are available	N	Y	Y	Y	N	N	Y
Supporting facts are annotated	N	N	Y	Y	N	N	Y
Decomposition is not evident	N	–	N	Y	Y	Y	Y
Multi-document inference	Y	N	N	N	Y	N	Y
Requires knowledge retrieval	Y	N	Y	N	Y	N	Y

Table 1: QASC has several desirable properties not simultaneously present in any single existing multihop QA dataset. Here "available" indicates that the dataset comes with a corpus that is guaranteed to contain supporting facts, while "annotated" indicates that these supporting facts are additionally annotated.

queries, as well as to identify what knowledge to retrieve. In fact, multi-hop questions in many existing datasets (Yang et al. 2018; Talmor and Berant 2018) often follow this syntactically decomposable pattern, with questions such as: "Which government position was held by the lead actress of X?"

All questions in QASC are human-authored, obtained via a multi-step crowdsourcing process (Section 3). To better enable development of both the reasoning and retrieval models, we also provide the pair of facts that were composed to create the question.[1] We use these annotations to develop a novel *two-step retrieval* technique that uses question-relevant facts to guide a second retrieval step. To make the dataset difficult for fine-tuned language models using our proposed retrieval (Section 5), we further augment the answer choices in our dataset via a *multi-adversary distractor choice selection* method (Section 6) that does not rely on computationally expensive multiple iterations of adversarial filtering (Zellers et al. 2018).

Even 2-hop reasoning for questions with implicit decomposition requires new approaches for retrieval and reasoning not captured by current datasets. Similar to other recent multi-hop reasoning tasks (Yang et al. 2018; Talmor and Berant 2018), we also focus on 2-hop reasoning, solving which will go a long way towards more general N-hop solutions.

In summary, we make the following contributions: (1) a dataset QASC of 9,980 8-way multiple-choice questions from elementary and middle school level science, with a focus on fact composition; (2) a pair of facts f_S,f_L from associated corpora annotated for each question, along with a composed fact f_C entailed by f_S and f_L, which can be viewed as a form of multi-sentence entailment dataset; (3) a novel two-step information retrieval approach designed for multi-hop QA that improves the recall of gold facts (by 43 pts) and QA accuracy (by 14 pts); and (4) an efficient multi-model adversarial answer choice selection approach.

QASC is challenging for current large pre-trained language models (Peters et al. 2018; Devlin et al. 2019), which exhibit a gap of 20% (absolute) to a human baseline of 93%, even when massively fine-tuned on 100K external QA examples in addition to QASC and provided with relevant knowledge using our proposed two-step retrieval.

[1]Questions, annotated facts, and corpora are available at https://github.com/allenai/qasc. Supplementary details are provided in a longer version of this paper at https://arxiv.org/abs/1910.11473.

2 Comparison With Existing Datasets

Table 1 summarizes how QASC compares with several existing datasets along five key dimensions (discussed below), which we believe are necessary for effectively developing retrieval and reasoning models for knowledge composition.

Existing datasets for the science domain require different reasoning techniques for each question (Clark et al. 2016; 2018). The dataset most similar to our work is OpenBookQA (Mihaylov et al. 2018), which comes with multiple-choice questions and a book of core science facts used as the seed for question generation. Each question requires combining the seed core fact with additional knowledge. However, it is unclear how many additional facts are needed, or whether these facts can even be retrieved from any existing knowledge sources. QASC, on the other hand, explicitly identifies two facts deemed (by crowd workers) to be sufficient to answer a question. These facts exist in an associated corpus and are provided for model development.

MultiRC (Khashabi et al. 2018a) uses passages to create multi-hop questions. However, MultiRC and other single-passage datasets (Mishra et al. 2018; Weston et al. 2015) have a stronger emphasis on passage discourse and entity tracking, rather than relation composition.

Multi-hop datasets from the Web domain use complex questions that bridge multiple sentences. We discuss 4 such datasets. (a) WikiHop (Welbl, Stenetorp, and Riedel 2018) contains questions in the tuple form $(e, r, ?)$ based on edges in a knowledge graph. However, WikiHop lacks questions with natural text or annotations on the passages that could be used to answer these questions. (b) ComplexWebQuestions (Talmor and Berant 2018) was derived by converting multi-hop paths in a knowledge-base into a text query. By construction, the questions can be decomposed into simpler queries corresponding to knowledge graph edges in the path. (c) HotPotQA (Yang et al. 2018) contains a mix of multi-hop questions authored by crowd workers using a pair of Wikipedia pages. While these questions were authored in a similar way, due to their domain and task setup, they also end up being more amenable to decomposition. (d) A recent dataset, DROP (Dua et al. 2019), requires discrete reasoning over text (such as counting or addition). Its focus is on performing discrete (e.g., mathematical) operations on extracted pieces of information, unlike our proposed sentence composition task.

Many systems answer science questions by composing multiple facts from semi-structured and unstructured knowl-

edge sources (Khashabi et al. 2016; Khot, Sabharwal, and Clark 2017; Jansen et al. 2017; Khashabi et al. 2018b). However, these often require careful manual tuning due to the large variety of reasoning techniques needed for these questions (Boratko et al. 2018) and the large number of facts that often must be composed together (Jansen 2018; Jansen et al. 2016). By limiting QASC to require exactly 2 hops (thereby avoiding semantic drift issues with longer paths (Fried et al. 2015; Khashabi et al. 2019)) and explicitly annotating these hops, we hope to constrain the problem enough so as to enable the development of supervised models for identifying and composing relevant knowledge.

2.1 Implicit Relation Decomposition

As mentioned earlier, a key challenge in QASC is that syntactic cues in the question are insufficient to determine how one should decompose the question relation, r_Q, into two sub-relations, r_S and r_L, corresponding to the associated facts f_S and f_L. At an abstract level, 2-hop questions in QASC generally exhibit the following form:

$$Q \triangleq r_Q(x_q, z_a^?)$$
$$r_S^?(x_q, y^?) \wedge r_L^?(y^?, z_a^?) \Rightarrow r_Q(x_q, z_a^?)$$

where terms with a '?' superscript represent unknowns: the decomposed relations r_S and r_L as well as the bridge concept y. (The answer to the question, $z_a^?$, is an obvious unknown.) To assess whether relation r_Q holds between some concept x_q in the question and some concept z_a in an answer candidate, one must come up with the missing or implicit relations and bridge concept. In our previous example, r_Q ="*harnessed for*", x_q ="*Differential heating of air*", y ="*wind*", r_S ="*produces*", and r_L ="*used for*".

In contrast, syntactically decomposable questions in many existing datasets often spell out both r_S and r_L: $Q \triangleq r_S(x_q, y^?) \wedge r_L(y^?, z_a^?)$. The example from the introduction, "Which government position was held by the lead actress of X?", could be stated in this notation as: *lead-actress*$(\mathrm{X}, y^?) \wedge$ *held-govt-posn*$(y^?, z_a^?)$.

This difference in how the question is presented in QASC makes it challenging to both retrieve relevant facts and reason with them via knowledge composition. This difficulty is further compounded by the property that a single relation r_Q can often be decomposed in *multiple ways* into r_S and r_L. We defer a discussion of this aspect to later, when describing QASC examples in Table 3.

3 Multihop Question Collection

Figure 2 gives an overall view of the crowdsourcing process. The process is designed such that each question in QASC is produced by composing two facts from an existing text corpus. Rather than creating compositional questions from scratch or using a specific pair of facts, we provide workers with only one seed fact f_S as the starting point. They are then given the creative freedom to find other relevant facts from a large corpus, F_L that could be composed with this seed fact. This allows workers to find other facts compose naturally with f_S and thereby prevent complex questions that describe the composition explicitly.

Once crowd-workers identify a relevant fact $f_L \in F_L$ that can be composed with f_S, they create a new composed fact f_C and use it to create a multiple-choice question. To ensure that the composed facts and questions are consistent with our instructions, we introduce automated checks to catch any inadvertent mistakes. E.g., we require that at least one intermediate entity (marked in black in subsequent sections) must be dropped to create f_C. We also ensure that the intermediate entity wasn't re-introduced in the question.

These questions are next evaluated against baseline systems to ensure hardness, i.e., at least one of the incorrect answer choices had to be be preferred over the correct choice by one of two QA systems (IR or BERT; described next), with a bonus incentive if both systems were distracted.

3.1 Input Facts

Seed Facts, F_S: We noticed that the quality of the seed facts can have a strong correlation with the quality of the question. So we created a small set of 928 good quality seed facts F_S from clean knowledge resources. We start with two medium size corpora of grade school level science facts: the WorldTree corpus (Jansen et al. 2018) and a collection of facts from the CK-12 Foundation.[2] Since the WorldTree corpus contains only facts covering elementary science, we used their annotation protocol to expand it to middle-school science. We then manually selected facts from these three sources that are amenable to creating 2-hop questions.[3] The resulting corpus F_S contains a total of 928 facts: 356 facts from WorldTree, 123 from our middle-school extension, and 449 from CK-12.

Large Text Corpus, F_L: To ensure that the workers are able to find any potentially composable fact, we used a large web corpus of 17M cleaned up facts F_L. We processed and filtered a corpus of 73M web documents (281GB) from Clark et al. (2016) to produce this clean corpus of 17M sentences (1GB). The procedure to process this corpus involved using *spaCy*[4] to segment documents into sentences, a Python implementation of Google's *langdetect*[5] to identify English-language sentences, *ftfy*[6] to correct Unicode encoding problems, and custom heuristics to exclude sentences with artifacts of web scraping like HTML, CSS and JavaScript markup, runs of numbers originating from tables, email addresses, URLs, page navigation fragments, etc.

3.2 Baseline QA Systems

Our first baseline is the **IR** system (Clark et al. 2016) designed for science QA with its associated corpora of web and science text (henceforth referred as the Aristo corpora). It retrieves sentences for each question and answer choice from the associated corpora, and returns the answer choice with the highest scoring sentence (based on the retrieval score).

[2] https://www.ck12.org

[3] While this is a subjective decision, it served our main goal of identifying a reasonable set of seed facts for this task.

[4] https://spacy.io/

[5] https://pypi.org/project/spacy-langdetect/

[6] https://github.com/LuminosoInsight/python-ftfy

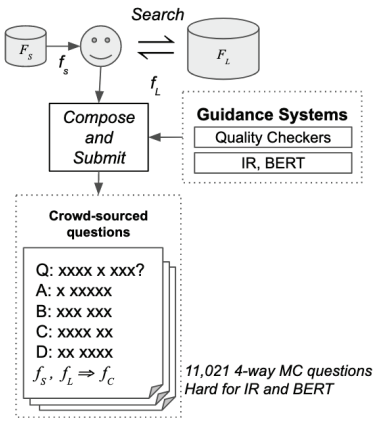

Figure 2: Crowd-sourcing questions using the seed corpus F_S and the full corpus F_L.

Our second baseline uses the language model **BERT** of Devlin et al. (2019). We follow their QA approach for the multiple-choice situation inference task SWAG (Zellers et al. 2018). Given question q and an answer choice c_i, we create `[CLS]` q `[SEP]` c_i `[SEP]` as the input to the model, with q being assigned to segment 0 and c_i to segment 1.[7] The model learns a linear layer to project the representation of the `[CLS]` token to a score for each choice c_i. We normalize the scores across all answer choices using softmax and train the model using the cross-entropy loss. When context/passage is available, we append the passage to segment 0, i.e., given a retrieved passage p_i, we provide `[CLS]` p_i q `[SEP]` c_i `[SEP]` as the input. We refer to this model as BERT-MCQ in subsequent sections.

For the crowdsourcing step, we use the `bert-large-uncased` model and fine-tuned it sequentially on two datasets: (1) RACE (Lai et al. 2017) with context; (2) SCI questions (ARC-Challenge+Easy (Clark et al. 2018) + OpenBookQA (Mihaylov et al. 2018) + Regents 12th Grade Exams[8]).

3.3 Question Validation

We validated these questions by having 5 crowdworkers answer them. Any question answered incorrectly or considered unanswerable by at least 2 workers was dropped, reducing the collection to 7,660 questions. The accuracy of the IR and BERT models used in Step 4 was 32.25% and 38.73%, resp., on this reduced subset.[9] By design, every question has the desirable property of being annotated with two sentences from F_{QASC} that can be composed to answer it. The low score of the IR model also suggests that these questions can not be answered using a single fact from the corpus.

We next analyze the retrieval and reasoning challenges associated with these questions. Based on these analyses, we will propose a new baseline model for multi-hop questions that substantially outperforms existing models on this task. We use this improved model to adversarially select additional distractor choices to produce the final QASC dataset.

4 Challenges

Table 2 shows sample crowd-sourced questions along with the associated facts. Consider the first question: *"What can trigger immune response?"*. One way to answer it is to first retrieve the two annotated facts (or similar facts) from the corpus. But the first fact, like many other facts in the corpus, overlaps only with the words in the answer *"transplanted organs"* and not with the question, making retrieval challenging. Even if the right facts are retrieved, the QA model would have to know how to compose the *"found on"* relation in the first fact with the *"trigger"* relation in the second fact. Unlike previous datasets (Yang et al. 2018; Talmor and Berant 2018), the relations to be composed are not explicitly mentioned in the question, making reasoning also challenging. We next discuss these two issues in detail.

4.1 Retrieval Challenges

We analyze the retrieval challenges associated with finding the two supporting facts associated with each question. Note that, unlike OpenBookQA, we consider the more general setting of retrieving relevant facts from a single large corpus $F_{QASC} = F_S \cup F_L$ instead of assuming the availability of a separate small book of facts (i.e., F_S).

Standard IR approaches for QA retrieve facts using question + answer as their IR query (Clark et al. 2016; Khot, Sabharwal, and Clark 2017; Khashabi et al. 2018b). While this can be effective for lookup questions, it is likely to miss important facts needed for multi-hop questions. In 96% of our crowd-sourced questions, at least one of the two annotated facts had an overlap of fewer than 3 tokens (ignoring stop words) with this question + answer query, making it difficult to retrieve such facts.[10] Note that our annotated facts form one possible pair that could be used to answer the question. While retrieving these specific facts isn't necessary, these crowd-authored questions are generally expected to have a similar overlap level to other relevant facts in our corpus.

Neural retrieval methods that use distributional representations can help mitigate the brittleness of word overlap measures, but also vastly open up the space of possibly relevant sentences. We hope that our annotated facts will be useful for training better neural retrieval approaches for multi-hop reasoning in future work. In this work, we focused on a modified non-neural IR approach that exploits the intermediate concepts not mentioned in the question (black words in our examples), which is explained in Section 5.1.

4.2 Reasoning Challenges

As described earlier, we collected these questions to require compositional reasoning where the relations to be composed are not obvious from the question. To verify this, we analyzed 50 questions from our final dataset and identified

[7] We assume familiarity with BERT's notation such as `[CLS]`, `[SEP]`, uncased models, and masking (Devlin et al. 2019).

[8] http://www.nysedregents.org/livingenvironment

[9] The scores are not 0% as crowdworkers were not required to distract both systems for every question.

[10] See Table 9 in Appendix E (provided in the longer version at https://arxiv.org/abs/1910.11473) for more details.

Question	Choices	Annotated Facts
What can *trigger immune response*?	(A) **Transplanted organs** (B) Desire (C) Pain (D) Death	f_S: Antigens are found on cancer cells and the cells of **transplanted organs**. f_L: Anything that can *trigger an immune response* is called an antigen.
What *forms caverns by seeping through rock and dissolving limestone*?	(A) **carbon dioxide in groundwater** (B) oxygen in groundwater (C) pure oxygen (D) magma in groundwater	f_S: *a cavern is formed by* carbonic acid in **groundwater** *seeping through rock and dissolving limestone*. f_L: When **carbon dioxide** is in water, it creates carbonic acid.

Table 2: Examples of questions generated via the crowd-sourcing process along with the facts used to create each question.

Fact 1	r_S	Fact 2	r_L	Composed Fact	r_Q
Antigens are found on cancer cells and the cells of transplanted organs.	located	Anything that can trigger an immune response is called an antigen.	causes	transplanted organs can trigger an immune response	causes
a cavern is formed by carbonic acid in groundwater seeping through rock and dissolving limestone	causes	Any time water and carbon dioxide mix, carbonic acid is the result.	causes	carbon dioxide in groundwater creates caverns	causes

Table 3: These examples of sentence compositions result in the same composed relation, `causes`, but via two different composition rules: located + causes ⇒ causes and causes + causes ⇒ causes These rules are not evident from the composed fact, requiring a model reasoning about the composed fact to learn the various possible decompositions of `causes`.

the key relations in f_S, f_L, and the question, referred to as r_S, r_L, and r_Q, respectively (see examples in Table 3). 7 of the 50 questions could be answered using only one fact and 4 of them didn't use either of the two facts. We analyzed the remaining 39 questions to categorize the associated reasoning challenges. In only 2 questions, the two relations needed to answer the question were explicitly mentioned in the question itself. In comparison, the composition questions in HotpotQA had both the relations mentioned in 47 out of 50 dev questions in our analysis.

Since there are a large number of lexical relations, we focus on 16 semantic relations in our analysis such as `causes`, `performs`, etc. These relations were defined based on previous analyses on science datasets (Clark et al. 2014; Jansen et al. 2016; Khashabi et al. 2016). We found 25 unique relation composition rules (i.e., $r_S(X,Y), r_L(Y,Z) \Rightarrow r_Q(X,Z)$). On average, we found every query relation r_Q had 1.6 unique relation compositions. Table 3 illustrates two different relation compositions that lead to the same `causes` query relation. As a result, models for QASC have a strong incentive to learn various possible compositions that lead to the same semantic relation, as well as extract them from text.

5 Question Answering Model

We now discuss our proposed two-step retrieval method and how it substantially boosts the performance of BERT-based QA models on crowd-sourced questions. This will motivate the need for adversarial choice generation.

5.1 Retrieval: Two-step IR

Consider the first question in Table 2. An IR approach that uses the standard $q + a$ query is unlikely to find the first fact since many irrelevant facts would also have the same overlapping words – "transplanted organs". However, it is likely to retrieve facts similar to the second fact, i.e., *"Antigens trigger immune response"*. If we could recognize antigen as an important intermediate entity that would lead to the answer, we can then query for sentences connecting this intermediate entity (*"antigens"*) to the answer (*"transplanted organs"*) which is then likely to find the first fact (*"antigens are found on transplanted organs"*). One potential way to identify such an intermediate concept is to consider the new entities introduced in the first retrieved fact that are absent from the question, i.e., $f_1 \setminus q$.

Based on this intuition, we present a simple but effective two-step IR baseline for multi-hop QA: **(1)** Retrieve K (=20 for efficiency) facts F_1 based on the query Q=q + a; **(2)** For each $f_1 \in F_1$, retrieve L (=4 to promote diversity) facts F_2 each of which contains at least one word from $Q \setminus f_1$ and from $f_1 \setminus Q$; **(3)** Filter $\{f_1, f_2\}$ pairs that do not contain any word from q or a; **(4)** Select top M unique facts from $\{f_1, f_2\}$ pairs sorted by the sum of their individual IR score.

Each retrieval query is run against an ElasticSearch[11] index built over F_{QASC} with retrieved sentences filtered to reduce noise (Clark et al. 2018). We use the set-difference between the stemmed, non-stopword tokens in $q + a$ and f_1 to identify the intermediate entity. Generally, we are interested in finding facts that connect new concepts introduced in the first fact (i.e., $f_1 \setminus Q$) to concepts not yet covered in question+answer (i.e., $Q \setminus f_1$).

Training a model on our annotations or essential terms (Khashabi et al. 2017) could help better identify these concepts. Recently, Khot, Sabharwal, and Clark (2019) proposed a span-prediction model to identify such intermediate entities for OpenBookQA questions. Their approach, however, assumes that one of the gold facts is provided as input to the model. Our approach, while specifically designed for 2-hop questions, can serve as a stepping stone towards developing retrieval methods for N-hop questions.

[11] https://www.elastic.co

The single step retrieval approach (using only f_1 but still requiring overlap with q and a) has an overall recall of only 2.9% (i.e., both f_S and f_L were in the top 10 sentences for 2.9% of the questions). The two-step approach, on the other hand, has a recall of 44.4%—a **15X improvement** (also limited to top M=10 sentences). Even if we relax the recall metric to finding f_S or f_L, the single step approach underperforms by 28% compared to the two-step retrieval (42.0 vs 69.9%). We will show in the next section that this improved recall also translates to improved QA scores. This shows the value of our two-step approach as well as the associated annotations: progress on the retrieval sub-task enabled by our fact-level annotations can lead to progress on the QA task.

5.2 Reasoning: BERT Models

We primarily use BERT-models fine-tuned on other QA datasets and with retrieved sentences as context, similar to prior state-of-the-art models on MCQ datasets (Sun et al. 2018; Pan et al. 2019).[12] There is a large space of possible configurations to build such a QA model (e.g., fine-tuning datasets, corpora) which we will explore later in our experimental comparisons. For simplicity, the next few sections will focus on one particular model: the `bert-large-cased` model fine-tuned on the RACE + SCI questions (with retrieved context[13]) and then fine-tuned on our dataset with single-step/two-step retrieval. For consistency, we use the same hyper-parameter sweep in all fine-tuning experiments (cf. Appendix D).

5.3 Results on Crowd-Sourced Questions

To enable fine-tuning models, we split the questions them into 5962/825/873 questions in train/dev/test folds, resp. To limit memorization, any two questions using the same seed fact, f_S, were always put in the same fold. Since multiple facts can cover similar topics, we further ensure that similar facts are also in the same fold. (See Appendix B for details.)

While these crowd-sourced questions were challenging for the baseline QA models (by design), models fine-tuned on this dataset perform much better. The BERT baseline that scored 38.7% on the crowd-sourced questions now scores 63.3% on the dev set after fine-tuning. Even the basic single-step retrieval context can improve over this baseline score by 14.9% (score: 78.2%) and our proposed two-step retrieval improves it even further by 8.2% (score: 86.4%). This shows that the distractor choices selected by the crowdsource workers were not as challenging once the model is provided with the right context. This can be also seen in the incorrect answer choices selected by them in Table 2 where they used words such as "Pain" that are associated with words in the question but may not have a plausible reasoning chain. To make this dataset more challenging for these models, we next introduce adversarial distractor choices.

[12]Experiments section contains numbers for other QA models.

[13]We use the same single-step retrieval over the large Aristo corpus as used by other BERT-based systems on ARC and OpenBookQA leaderboards.

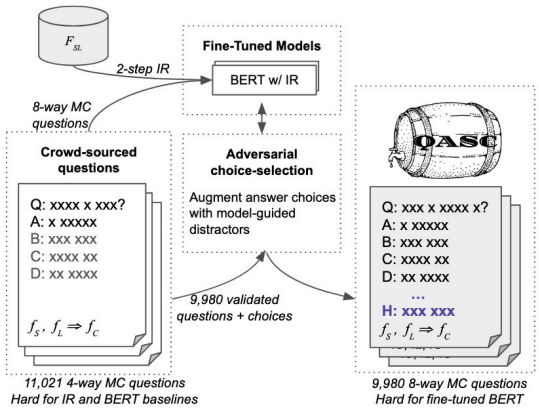

Figure 3: Generating QASC questions using adversarial choice selection.

6 Adversarial Choice Generation

To make the crowdsourced dataset challenging for fine-tuned language models, we use model-guided *adversarial choice generation* to expand each crowdsourced question into an 8-way question. Importantly, the human-authored body of the question is left intact (only the choices are augmented), to avoid a system mechanically reverse-engineering how a question was generated.

Previous approaches to adversarially create a hard dataset have focused on iteratively making a dataset harder by sampling harder choices and training stronger models (Zellers et al. 2018; 2019a). While this strategy has been effective, it involves multiple iterations of model training that can be prohibitively expensive with large LMs. In some cases (Zellers et al. 2018; 2019b), they need a generative model such as GPT-2 (Radford et al. 2019) to produce the distractor choices. We, on the other hand, have a simpler setup where we train only a few models and do not require a model to generate the distractor choices.

6.1 Distractor Options

To create the space of distractors, we follow Zellers et al. (2019a) and use correct answer choices from other questions. This ensures that a model won't be able to predict the correct answer purely based on the answer choices (one of the issues with OpenBookQA). To reduce the chances of a correct answer being added to the set of distractors, we pick them from the most dissimilar questions. We further filter these choices down to $\sim$30 distractor choices per question by removing the easy distractors based on the fine-tuned BERT baseline model. Further implementation details are provided in Appendix C.

This approach of generating distractors has an additional benefit: we can recover the questions that were rejected earlier for having multiple valid answers (in § 3.3). We add back 2,774 of the 3,361 rejected questions that (a) had at least one worker select the right answer, and (b) were deemed unanswerable by at most two workers. We, however, ignore all crowdsourced distractors for these questions since they were considered potentially correct answers in the validation task.

	Dev Accuracy	
	Single-step retr.	Two-step retr.
Original Dataset (4-way)	78.2	86.4
Random Distractors (8-way)	74.9	83.3
Adversarial Distractors (8-way)	61.7	72.9

Table 4: Results of the BERT-MCQ model on the adversarial dataset using `bert-large-cased` model and pre-trained on RACE + SCI questions.

We use the adversarial distractor selection process (to be described shortly) to add the remaining 7 answer choices.

To ensure a clean evaluation set, we use **another crowdsourcing task** where we ask 3 annotators to identify *all possible valid answers* from the candidate distractors for the dev and test sets. We filter out answer choices in the distractor set that were considered valid by at least one turker. Additionally, we filter out low-quality questions where more than four distractor choices were marked valid or the correct answer was not included in the selection. This dropped 20% of the dev and test set questions and finally resulted in train/dev/test sets of size 8134/926/920 questions with an average of 30/26.9/26.1 answer choices (including the correct one) per question.

6.2 Multi-Adversary Choice Selection

We first explain our approach, assuming access to K models for multiple-choice QA. Given the number of datasets and models proposed for this task, this is not an unreasonable assumption. In this work, we use K BERT models, but the approach is applicable to any QA model.

Our approach aims to select a diverse set of answers that are challenging for different models. As described above, we first create $\sim$30 distractor options, D for each question. We then sort these distractor options based on their relative difficulty for these models, defined as the number of models fooled by this distractor: $\sum_k \mathbf{I}\big[m_k(q, d_i) > m_k(q, a)\big]$ where $m_k(q, c_i)$ is the k-th model's score for the question q and choice c_i. In case of ties, we then sort these distractors based on the difference between the scores of the distractor choice and the correct answer: $\sum_k \big(m_k(q, d_i) - m_k(q, a)\big)$.[14]

We used BERT-MCQ models that were fine-tuned on the RACE +SCI dataset as described in the previous section. We additionally fine-tune these models on the training questions with random answer choices added from the the space of distractors to make each question an 8-way multiple-choice question. This ensures that our models have seen answer choices from both the human-authored and algorithmically selected space of distractors. Drawing inspiration from bootstrapping (Breiman 1996), we create two such datasets with randomly selected distractors from D and use the models fine-tuned on these datasets as m_k (i.e, $K = 2$). There is a large space of possible models and scoring functions that may be explored,[15] but we found this simple approach to be effective at identifying good distractors. This process of generating the adversarial dataset is depicted in Figure 3.

6.3 Evaluating Dataset Difficulty

We select the top scoring distractors using the two BERT-MCQ models such that each question is converted into an 8-way MCQ (including the correct answer and human-authored valid distractors). To verify that this results in challenging questions, we again evaluate using the BERT-MCQ models with two different kinds of retrieval. Table 4 compares the difficulty of the adversarial dataset to the original dataset and the dataset with random distractors (used for fine-tuning BERT-MCQ models).

The original 4-way MCQ dataset was almost solved by the two-step retrieval approach and increasing it to 8-way with random distractors had almost no impact on the scores. But our adversarial choices drop the scores of the BERT model given context from either of the retrieval approaches.

6.4 QASC Dataset

The final dataset contains **9,980** questions split into **[8134|926|920]** questions in the [train|dev|test] folds. Each question is annotated with two facts that can be used to answer the question. These facts are present in a corpus of 17M sentences (also provided). The questions are similar to the examples in Table 2 but expanded to an 8-way MCQ and with shuffled answer choices. E.g., the second example there was changed to "What forms caverns by seeping through rock and dissolving limestone? (A) pure oxygen (B) Something with a head, thorax, and abdomen (C) basic building blocks of life (D) **carbon dioxide in groundwater** (E) magma in groundwater (F) oxygen in groundwater (G) At the peak of a mountain (H) underground systems".

Table 6 gives a summary of QASC statistics, and Table 7 in the Appendix provides additional examples.

7 Experiments

While we used large pre-trained language models first fine-tuned on other QA datasets ($\sim$100K examples) to ensure that QASC is challenging, we also evaluate BERT models without any additional fine-tuning here. All models are still fine-tuned on the QASC dataset.

To verify that our dataset is challenging also for models that do not use BERT (or any other transformer-based architecture), we evaluate Glove (Pennington, Socher, and Manning 2014) based models developed for multiple-choice science questions in OpenBookQA. Specifically, we consider these non-BERT baseline models:

- **Odd-one-out**: Answers the question based on just the choices by identifying the most dissimilar answer.
- **ESIM Q2Choice** (with and without Elmo): Uses the ESIM model (Chen et al. 2017) with Elmo embeddings (Peters et al. 2018) to compute how much does the question entail each answer choice.

[14] Since we use normalized probabilities as model scores, we do not normalize them here.

[15] For example, we evaluated the impact of increasing K, but didn't notice any change in the fine-tuned model's score.

	Model	Embedding	Retr. Corpus (#docs)	Retrieval Approach	Addnl. fine-tuning (#examples)	Dev Acc.	Test Acc.
	Human Score						**93.0**
	Random					12.5	12.5
OBQA Models	ESIM Q2Choice	Glove				21.1	17.2
	ESIM Q2Choice	Glove Elmo				17.1	15.2
	Odd-one-out	Glove				22.4	18.0
BERT Models	BERT-MCQ	BERT-LC	F_{QASC} (17M)	Single-step		59.8	53.2
	BERT-MCQ	BERT-LC	F_{QASC} + ARC (31M)	Single-step		62.3	57.0
	BERT-MCQ	BERT-LC	F_{QASC} + ARC(31M)	Two-step		66.6	58.3
	BERT-MCQ	BERT-LC	F_{QASC} (17M)	Two-step		71.0	67.0
Addnl. Fine-tuning	AristoBertV7	BERT-LC[WM]	Aristo (1.7B)	Single-step	RACE + SCI (97K)	69.5	62.6
	BERT-MCQ	BERT-LC	F_{QASC} (17M)	Two-step	RACE + SCI (97K)	72.9	68.5
	BERT-MCQ	BERT-LC[WM]	F_{QASC} (17M)	Two-step	RACE + SCI (97K)	**78.0**	**73.2**

Table 5: QASC scores for previous state-of-the-art models on multi-hop Science MCQ(OBQA), and BERT models with different corpora, retrieval approaches and additional fine-tuning. While the simpler models only show a small increase relative to random guessing, BERT can achieve upto 67% accuracy by fine-tuning on QASC and using the two-step retrieval. Using the BERT models pre-trained with whole-word masking and first fine-tuning on four relevant MCQ datasets (RACE and SCI(3)) improves the score to 73.2%, leaving a gap of over 19.8% to the human baseline of 93%. ARC refers to the corpus of 14M sentences from Clark et al. (2018), BERT-LC indicates 'bert-large-cased' and BERT-LC[WM] indicates whole-word masking.

	Train	Dev	Test
Number of questions	8,134	926	920
Number of unique f_S	722	103	103
Number of unique f_L	6,157	753	762
Average question length (chars)	46.4	45.5	44.0

Table 6: QASC dataset statistics

As shown in Table 5, OpenBookQA models, that had close to the state-of-the-art results on OpenBookQA, perform close to the random baseline on QASC. Since these mostly rely on statistical correlations between questions and across choices,[16] this shows that this dataset doesn't have any easy shortcuts that can be exploited by these models.

Second, we evaluate BERT models with different corpora and retrieval. We show that our two-step approach always out-performs the single-step retrieval, even when given a larger corpus. Interestingly, when we compare the two single-step retrieval models, the larger corpus outperforms the smaller corpus, presumably because it increases the chances of having a single fact that answers the question. On the other hand, the smaller corpus is better for the two-step retrieval approach, as larger and noisier corpora are more likely to lead a 2-step search astray.

Finally, to compute the current gap to human performance, we consider a recent state-of-the-art model on multiple leaderboards: AristoBertV7 that uses the BERT model trained with whole-word masking,[17] fine-tuned on the RACE +SCI questions and retrieves knowledge from a very large corpus. Our two-step retrieval based model outperforms this model and improves even further with more fine-tuning. Replacing the pre-trained `bert-large-cased` model with the whole-word masking based model further improves the score by 4.7%, but there is still a gap of ∼20% to the human score of 93% on this dataset.

8 Conclusion

We present QASC, the first QA dataset for multi-hop reasoning beyond a single paragraph where two facts needed to answer a question are annotated for training, but questions cannot be easily syntactically decomposed into these facts. Instead, models must learn to retrieve and compose candidate pieces of knowledge. QASC is generated via a crowdsourcing process, and further enhanced via multi-adversary distractor choice selection. State-of-the-art BERT models, even with massive fine-tuning on over 100K questions from previous relevant datasets and using our proposed two-step retrieval, leave a large margin to human performance levels, thus making QASC a new challenge for the community.

Acknowledgments

We thank Oyvind Tafjord for his extension to AllenNLP that was used to train our BERT models, Nicholas Lourie for his "A Mechanical Turk Interface (amti)" tool used to launch crowdsourcing tasks, Dirk Groeneveld for his help collecting seed facts, and Sumithra Bhakthavatsalam for helping generate the QASC fact corpus. We thank Sumithra Bhakthavatsalam, Kaj Bostrom, Kyle Richardson, and Madeleine van Zuylen for initial human evaluations. We also thank Jonathan Borchardt and Dustin Schwenk for inspiring dis-

[16] Their knowledge-based models do not scale to our corpus of 17M sentences.

[17] https://github.com/google-research/bert

cussions about, and guidance with, early versions of the MTurk task. We thank the Amazon Mechanical Turk workers for their effort in creating and annotating QASC questions. Computations on beaker.org were supported in part by credits from Google Cloud.

References

Boratko, M.; Padigela, H.; Mikkilineni, D.; Yuvraj, P.; Das, R.; McCallum, A.; Chang, M.; Fokoue-Nkoutche, A.; Kapanipathi, P.; Mattei, N.; Musa, R.; Talamadupula, K.; and Witbrock, M. 2018. A systematic classification of knowledge, reasoning, and context within the ARC dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*.

Breiman, L. 1996. Bagging predictors. *Machine learning* 24(2):123–140.

Chen, Q.; Zhu, X.-D.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced lstm for natural language inference. In *ACL*.

Clark, P.; Balasubramanian, N.; Bhakthavatsalam, S.; Humphreys, K.; Kinkead, J.; Sabharwal, A.; and Tafjord, O. 2014. Automatic construction of inference-supporting knowledge bases. In *4th Workshop on Automated Knowledge Base Construction (AKBC)*.

Clark, P.; Etzioni, O.; Khot, T.; Sabharwal, A.; Tafjord, O.; Turney, P. D.; and Khashabi, D. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR* abs/1803.05457.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*.

Fried, D.; Jansen, P.; Hahn-Powell, G.; Surdeanu, M.; and Clark, P. 2015. Higher-order lexical semantic models for non-factoid answer reranking. *TACL* 3:197–210.

Jansen, P.; Balasubramanian, N.; Surdeanu, M.; and Clark, P. 2016. What's in an explanation? characterizing knowledge and inference requirements for elementary science exams. In *COLING*.

Jansen, P.; Sharp, R.; Surdeanu, M.; and Clark, P. 2017. Framing qa as building and ranking intersentence answer justifications. *Computational Linguistics* 43:407–449.

Jansen, P. A.; Wainwright, E.; Marmorstein, S.; and Morrison, C. T. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of LREC*.

Jansen, P. 2018. Multi-hop inference for sentence-level textgraphs: How challenging is meaningfully combining information for science question answering? In *TextGraphs@NAACL-HLT*.

Khashabi, D.; Khot, T.; Sabharwal, A.; Clark, P.; Etzioni, O.; and Roth, D. 2016. Question answering via integer programming over semi-structured knowledge. In *IJCAI*.

Khashabi, D.; Khot, T.; Sabharwal, A.; and Roth, D. 2017. Learning what is essential in questions. In *CoNLL*.

Khashabi, D.; Chaturvedi, S.; Roth, M.; Upadhyay, S.; and Roth, D. 2018a. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL*.

Khashabi, D.; Khot, T.; Sabharwal, A.; and Roth, D. 2018b. Question answering as global reasoning over semantic abstractions. In *AAAI*.

Khashabi, D.; Azer, E. S.; Khot, T.; Sabharwal, A.; and Roth, D. 2019. On the capabilities and limitations of reasoning for natural language understanding. *CoRR* abs/1901.02522.

Khot, T.; Sabharwal, A.; and Clark, P. 2017. Answering complex questions using open information extraction. In *ACL*.

Khot, T.; Sabharwal, A.; and Clark, P. 2019. What's missing: A knowledge gap guided approach for multi-hop question answering. In *EMNLP*.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. H. 2017. RACE: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *EMNLP*.

Mishra, B. D.; Huang, L.; Tandon, N.; tau Yih, W.; and Clark, P. 2018. Tracking state changes in procedural text: A challenge dataset and models for process paragraph comprehension. In *NAACL*.

Pan, X.; Sun, K.; Yu, D.; Ji, H.; and Yu, D. 2019. Improving question answering with external knowledge. In *MRQA: Machine Reading for Question Answering Workshop at EMNLP-IJCNLP 2019*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Sun, K.; Yu, D.; Yu, D.; and Cardie, C. 2018. Improving machine reading comprehension with general reading strategies. *CoRR* abs/1810.13441.

Talmor, A., and Berant, J. 2018. The web as a knowledge-base for answering complex questions. In *NAACL-HLT*.

Welbl, J.; Stenetorp, P.; and Riedel, S. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL* 6:287–302.

Weston, J.; Bordes, A.; Chopra, S.; and Mikolov, T. 2015. Towards AI-complete question answering: A set of prerequisite toy tasks. *CoRR* abs/1502.05698.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Zellers, R.; Bisk, Y.; Schwartz, R.; and Choi, Y. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019a. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019b. HellaSwag: Can a machine really finish your sentence? In *ACL*.