Can a Suit of Armor Conduct Electricity?
A New Dataset for Open Book Question Answering

Todor Mihaylov[‡] and **Peter Clark**[†] and **Tushar Khot**[†] and **Ashish Sabharwal**[†]

[†] Allen Institute for Artificial Intelligence, Seattle, WA, U.S.A.
[‡] Research Training Group AIPHES & Heidelberg University, Heidelberg, Germany
{peterc,tushark,ashishs}@allenai.org, mihaylov@cl.uni-heidelberg.de

Abstract

We present a new kind of question answering dataset, OpenBookQA, modeled after open book exams for assessing human understanding of a subject. The open book that comes with our questions is a set of 1326 elementary level science facts. Roughly 6000 questions probe an understanding of these facts and their application to novel situations. This requires combining an open book fact (e.g., metals conduct electricity) with broad common knowledge (e.g., a suit of armor is made of metal) obtained from other sources. While existing QA datasets over documents or knowledge bases, being generally self-contained, focus on linguistic understanding, OpenBookQA probes a deeper understanding of both the topic—in the context of common knowledge—and the language it is expressed in. Human performance on OpenBookQA is close to 92%, but many state-of-the-art pre-trained QA methods perform surprisingly poorly, worse than several simple neural baselines we develop. Our oracle experiments designed to circumvent the knowledge retrieval bottleneck demonstrate the value of both the open book and additional facts. We leave it as a challenge to solve the retrieval problem in this multi-hop setting and to close the large gap to human performance.

1 Introduction

Open book exams are a common mechanism for assessing human understanding of a subject, where test takers are allowed free access to a relevant book, study guide, or class notes when answering questions. In this context, the goal is not to evaluate memorization but a deeper understanding of the material and its application to new situations (Jenkins, 1995; Landsberger, 1996). The application, in turn, often requires combining a fact in the book (e.g., *metals conduct electricity*) with additional common knowledge the test taker is ex-

Figure 1: An example for a question with a given set of choices and supporting facts.

pected to have acquired by this stage (e.g., *a suit of armor is made of metal*).

Motivated by this setting, we present a new kind of question answering dataset, OpenBookQA,[1] that consists of two parts: $\mathcal{Q}$, a set of 5957 multiple-choice questions, and $\mathcal{F}$, a set of 1326 diverse facts about elementary level science. $\mathcal{F}$ has three key characteristics of an 'open book': (a) it forms the basis for generating $\mathcal{Q}$; (b) it has been deemed central to scientific explanations (Jansen et al., 2018); and (c) by itself, $\mathcal{F}$ is generally insufficient to answer questions in $\mathcal{Q}$. Faced with a question $q \in \mathcal{Q}$, a student or system S is expected retrieve a relevant fact $f \in \mathcal{F}$, and appeal to their own common knowledge, $\mathcal{K}_{\mathcal{S}}$, when applying f to answer q.

Figure 1 provides an example. Here, *metals are thermal conductors* is a core scientific fact available in $\mathcal{F}$. One way to apply this fact to decide whether *a steel spoon* would let the *most heat travel through* is to appeal to common knowledge that steel is metallic and heat travels through thermal conductors. In general, the expected common knowledge is relatively simple (taxonomic facts,

---

[1] The dataset and the code for the models are available at http://data.allenai.org/OpenBookQA.

definitions, object properties, etc.); the difficulty lies in identifying it and meaningfully combining it with a core fact from $\mathcal{F}$ to answer the question.

OpenBookQA questions are challenging as they require *multi-hop reasoning with partial context* provided by $\mathcal{F}$. Specifically, unlike existing datasets for reading comprehension (RC), answering questions on the back of a textbook (TQA),[2] as well as question answering over structured knowledge-bases (KBQA), the open book $\mathcal{F}$ that comes with OpenBookQA is not self-contained. A successful system must therefore go beyond the typical challenges such as paraphrase matching and coreference resolution, without benefiting from the canonicalized and complete information in KBQA.

Generating interesting open book questions is a difficult task. We used a multi-stage process starting with $\mathcal{F}$, using crowd-sourcing to generate (noisy) questions based on $\mathcal{F}$ that probe novel situations, using an automatic filter to ensure hardness for retrieval and association based systems, using a crowd filter to ensure answerability by a lay person, and further using an expert filter to ensure higher quality in Dev and Test sets.

We evaluate a number of existing QA systems for science (without retraining) on OpenBookQA, finding that they perform surprisingly close to the random guessing baseline of 25%. Human performance, on the other hand, is close to 92%.[3]

Motivated by recent findings of gameability of NLP datasets (Gururangan et al., 2018), we also develop and evaluate simple, attention-based, neural baselines including a *plausible answer detector* (which ignores the question text completely) and an *odd-one-out solver*. These highlight inevitable human bias in any crowdsourced dataset, increasing performance on OpenBookQA to 48%.

Building upon a recent neural model for incorporating external knowledge in the story cloze setting (Mihaylov and Frank, 2018), we propose a knowledge-aware neural baseline that can utilize both the open book $\mathcal{F}$ and common knowledge retrieved from sources such as ConceptNet (Speer et al., 2017). While retrieving the most useful pieces of knowledge remains an open challenge, our 'oracle' experiments with the fact f used while generating a question q and an interpretation (by

the question author) of the additional knowledge k needed for q, provides valuable insight into the nature of this dataset: Facts from the open book $\mathcal{F}$ are valuable (5% improvement) but not sufficient. Using both f and k increases the accuracy to 76%, but is still far from human level performance, suggesting the need for non-trivial reasoning to combine these facts.

To encourage further research on this new task, for each Train and Dev question q, OpenBookQA also includes f as intermediate supervision signal, which may be viewed as a partial *explanation* for q. We leave closing the large gap to human performance as a challenge for the NLP community.

2 Related Work

By construction, answering OpenBookQA questions requires (i) some base science facts from a provided 'open book', (ii) broader understanding about the world (common or commonsense knowledge), and (iii) an ability to combine these facts (reasoning). This setup differs from several existing QA tasks, as summarized below.

Reading Comprehension (RC) datasets have been proposed as benchmarks to evaluate the ability of systems to understand a document by answering factoid-style questions over this document. These datasets have taken various forms: multiple-choice (Richardson et al., 2013), cloze-style (Hermann et al., 2015; Onishi et al., 2016; Hill et al., 2016), and span prediction (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017) However, analysis (Chen et al., 2016; Sugawara et al., 2017) of these datasets has shown that many of the questions can be solved with context token matching (Chen et al., 2017a; Weissenborn et al., 2017) or relatively simple paraphrasing.

To focus on the more challenging problem of reasoning across sentences, new datasets have been proposed for multi-step RC. **QAngaroo** (Welbl et al., 2018) have used a knowledge-base to identify entity pairs (s, o) with a known relation, r, which is also supported by a multi-hop path in a set of documents. They use structured tuple queries (s, r, ?) and use all the documents along the path as the input passage. **NarrativeQA** (Kociský et al., 2017) is an RC dataset that has been shown to require an iterative reasoning about the narrative of a story. Similar to OpenBookQA, the questions were generated to ensure that the answer is not a direct match or paraphrase

---

[2]Only ∼5% of the TQA questions of Kembhavi et al. (2017) require additional common knowledge.

[3]To avoid ambiguity in the term 'human performance', Section 3.2 describes the specific randomized model we use.

that can be retrieved with an IR approach. Most recently, Khashabi et al. (2018) proposed **MultiRC**, a multiple-choice RC dataset that is designed to require multi-sentence reasoning and can have multiple correct answers. Again, like most RC datasets, it is self-contained.

Tasks with external knowledge. While many of the RC datasets could benefit from commonsense or background knowledge, they are designed to be self-contained, i.e., solvable by the document context alone. Datasets such as the **Story Cloze Test** (Mostafazadeh et al., 2016), **MCScript**,[4] and **ProPara** (Mishra et al., 2018) do require additional domain knowledge about everyday events, scripts, and processes, respectively. However, these datasets need domain-specific modeling of events, whereas OpenBookQA appeals to broad common knowledge cutting across a variety of types and topics.

Stasaski and Hearst (2017) explore the creation of multi-hop questions and propose generating stronger distractors for the multiple-choice setting. Their work, however, starts with structured knowledge, specifically a Biology ontology.

Lastly, many **Science Question Answering** datasets (e.g. Clark et al., 2016, 2018) have been released that need broad external knowledge to answer the questions. However, these questions are not associated with a core set of facts, i.e., an "open book" used to define these questions. As a result, the questions vary widely in style and complexity (Clark et al., 2018). In contrast, OpenBookQA focuses on a more well-defined subset of science QA, appealing to one core fact from the open book and one (or few) relatively simple commonly known supporting facts.

3 OpenBookQA Dataset

The OpenBookQA dataset consists of about 6,000 4-way multiple-choice questions, each associated with one core fact from a "book" $\mathcal{F}$ of 1326 such facts, and an auxiliary set $\mathcal{K}$ of about 6000 additional facts. The questions were created via a multi-stage crowdsourcing and partial expert filtering process, discussed in Section 3.1.

The small "book" $\mathcal{F}$ consists of recurring science themes and principles, each of which can be (and here is) instantiated into multiple questions.

For $\mathcal{F}$, we use a subset of the WorldTree corpus which Jansen et al. (2018) have analyzed for sufficiency for elementary level science. The subset we use is taken from the 2287 WorldTree facts that were marked as "central" by the original authors in at least one explanation. We further filter them down to 1326 that appear general enough to be applicable to multiple situations.

OpenBookQA additionally requires broad common knowledge, which is expected to come from large corpora, such as ConceptNet, Wikipedia, or a corpus with 14M science-related sentences used by some existing baselines. The crowdsourcing process below also asks workers to mark a second fact, k, needed for each question q, in addition to f. These second facts, unfortunately, were often incomplete, over-complete, or only distantly related to q. We thus include in OpenBookQA the set $\mathcal{K}$ of such second facts only as *auxiliary data* for optional use. We emphasize that $\mathcal{K}$ should not be viewed as 'gold' additional facts, or as a substitute for broad common knowledge.

3.1 Crowdsourcing Process

The overall question generation and filtering pipeline is summarized in Figure 2. Given the "book" $\mathcal{F}$ of core facts, the process proceeds as follows, starting with an empty question set Qs and an empty 'second facts' set $\mathcal{K}$:

1. A crowd-worker[5] w is shown a random science fact f from the set $\mathcal{F}$.

2. w is asked to think of a second common fact, k, that may be combined with f to derive a new, valid assertion s.

3. w then converts s into a question-answer pair and extends this into a 4-way multiple choice question by adding 3 incorrect answer choices, $q_{\mathrm{mc}} = (q, \{c_1, c_2, c_3, c_4\})$, where one of the c_i's is the unique correct answer.

4. The system verifies q_{mc} passes basic checks such as uniformity of answer choices.[6]

5. w then feeds the multiple-choice question q_{mc} to an information retrieval solver (Clark et al.,
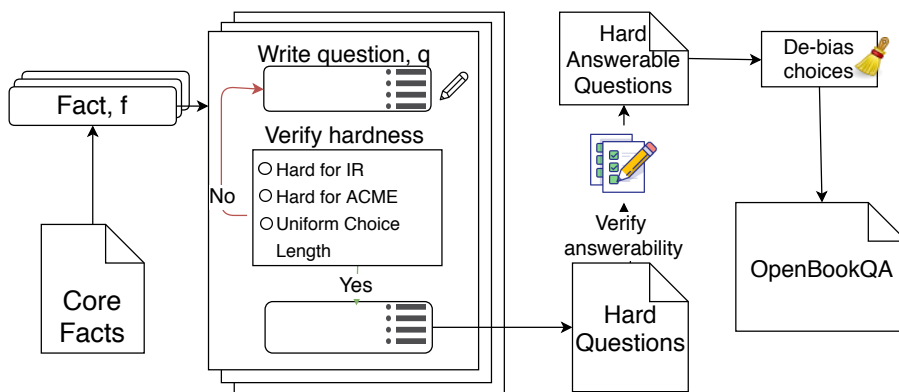

Figure 2: OpenBookQA question generation pipeline

2016) and a word association based solver (Turney, 2017), and verifies that (a) neither of them answers q_{mc} correctly and (b) the top 3 IR retrieved sentences are insufficient to answer q_{mc}; if not, the question is edited and re-tried.

6. Question q_{mc} is then shown to 5 new crowd-workers, who are asked to answer it.

7. If at least 4 out of 5 workers answer q_{mc} correctly, it is deemed answerable and the process continues. If not, q_{mc} is discarded.

8. The answer choices of q_{mc} are randomly shuffled to avoid unintended bias.[7]

9. q_{mc} is associated with f as the core science fact and added to the question set $\mathcal{Q}$. k is added to the set $\mathcal{K}$ of additional (noisy) facts.

The Dev and Test splits were further filtered by an in-house expert to ensure higher quality.

3.2 Human Performance

To assess human accuracy on this dataset, we consider the following model: Each question $q \in \mathcal{Q}$ has some (unknown) human accuracy p_q, defined as the probability that a random human subject, chosen uniformly from a large pool $\mathcal{H}$, would answer q correctly. Thus, we can think of this as defining a Bernoulli random variable, $X_q \sim B(p_q)$, whose mean is (unknown) p_q. The average human accuracy on $\mathcal{Q}$ under this model is:

$$H(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} p_q$$

where $\{p_q \mid q \in \mathcal{Q}\}$ are unknown.

With $\mathcal{H}$ as the set of crowd-workers (cf. Footnote 5), step 6 of the above question generation

process is equivalent to obtaining 5 independent samples, $X_{q,i}, i \in I, |I| = 5$, from $B(p_q)$. We must, however, be careful when using this data to estimate p_q, as the same 5 samples were used to decide whether q makes it into the question set $\mathcal{Q}$ or not. For instance, if we had kept only those questions that all 5 workers answered correctly, it would clearly be inaccurate to claim that the human accuracy on $\mathcal{Q}$ is 100%. Nevertheless, it is possible to re-use the judgments from Step 6 to approximate $H(\mathcal{Q})$ with high confidence, without posing the questions to new workers.

Intuitively, if all questions in $\mathcal{Q}$ were difficult to answer (i.e., all p_q were small), it would be unlikely that all $|\mathcal{Q}|$ questions would pass the test in Step 6. We can use the contrapositive of this observation to conclude that p_q, on average, must have been high for $q \in \mathcal{Q}$.

Formally, aggregating across all questions gives the following empirical estimate of $H(\mathcal{Q})$:

$$\tilde{H}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|I|} \sum_{i \in I} X_{q,i}$$
$$= \frac{1}{|\mathcal{Q}||I|} \sum_{q \in \mathcal{Q}, i \in |I|} X_{q,i}$$

For analysis, we assume all samples $X_{q,i}$ are independent, i.e., every answer is obtained independently.[8] An application of Hoeffding's Inequality (Hoeffding, 1963) shows that $\tilde{H}(\mathcal{Q})$ converges to $H(\mathcal{Q})$ very rapidly as $n = |\mathcal{Q}||I|$ grows; specifically, $\tilde{H}(\mathcal{Q}) \leq H(\mathcal{Q}) + t$ with probability at least $1 - \exp(-2nt^2)$; similarly for $\tilde{H}(\mathcal{Q}) \geq H(\mathcal{Q}) - t$. In our Dev and Test sets, where $|\mathcal{Q}| = 500$ and $|I| = 5$, this translates into $H(Q)$ being at least

---

[7]Choice 'A' was the correct answer in 69% of the questions at the end of Step 4.

[8]Realistically, there is some dependence across questions as a single worker may answer multiple questions. We leave a formal analysis of this setting as future work.

OpenBookQA Statistics	
# of questions	5957
# of choices per question	4
Avg. question sentences	1.08 (6)
Avg. question tokens	11.46 (76)
Avg. choice tokens	2.89 (23)
Avg. science fact tokens	9.38 (28)
Vocabulary size (q+c)	11855
Vocabulary size (q+c+f)	12839
Answer is the longest choice	1108 (18.6%)
Answer is the shortest choice	216 (3.6%)

Table 1: Statistics for full OpenBookQA dataset. Parenthetical numbers next to each average are the *max*.

Fact Type	% Questions	% Facts
PROPERTY	29.11%	25.81%
ISA	20.25%	17.20%
BASIC	17.72%	19.35%
DEFINITION	17.72%	15.05%
CAUSAL	11.39%	9.68%
OTHERS	13.92%	12.90%

Table 2: Percentage of questions and facts for the five most common type of additional facts. Note that % Questions does not add up to 100% since we count the percentage of questions where at least one such fact is needed.

$\tilde{H}(\mathcal{Q}) - 3\%$ with probability over 98.8% and at least $\tilde{H}(\mathcal{Q}) - 2.5\%$ with prob 95.6%; we report the former as our conservative estimate on human performance.

3.3 Question Set Analysis

OpenBookQA consists of 5957 questions, with 4957/500/500 in the Train/Dev/Test splits.[9] Table 1 summarizes some statistics about the full dataset. Each question has exactly four answer choices and one associated fact used in the creation process. We report the average length of questions, candidate choices, and associated facts, as well as how often is the longest/shortest choice the correct one.

We analyzed 100 questions in the Train set to capture the kind of common knowledge and reasoning needed. For each, we wrote down the additional common knowledge needed to answer this question *in addition* to the original science fact. In 21% of the cases, the crowdsourced question actually tests for a fact that doesn't necessarily need the original science fact. For example, the question: "On a rainy day the clouds are (A) low (B) white (C) small (D) gray" was written based on the science fact "clouds produce rain" but doesn't need this fact to answer it. We ignore such questions in our analysis. For the remaining questions, we categorized the additional facts into five high-level categories (and collapsed the remaining facts into a catch-all OTHERS category) based on previous approaches on similar science questions (Clark et al., 2018; Jansen et al., 2016):

1. ISA: Basic taxonomic facts such as isa(tree,

living thing), isa(granite, rock).

2. PROPERTY: Properties of objects such as madeof(belt buckle, metal), has(mammals, four legs), contains(lemon juice, citric acid).

3. DEFINITION: Definitions of objects that may be based on their appearance (tape is a plastic with markings), working mechanism (telescope is a device that uses mirrors to view objects), etc.

4. CAUSAL: Causal facts such as causes(adding lemon juice to milk, milk to break down).

5. BASIC: General scientific fact that did not fit above, e.g. squirrels eat nuts for food.

Table 2 presents the proportions of these facts in our analyzed question set. For each type of fact, we calculate the percentage of questions that need at least one such fact (shown as % Questions). We also calculate the overall percentage of each fact type across all the common knowledge facts (shown as % Facts). Most of our questions need simple facts such as *isa* knowledge and properties of objects, further confirming the need for simple reasoning with common knowledge. Apart from these five major categories of facts, the catch-all OTHERS category contains commonsense facts (e.g., it is dark at night), world knowledge (e.g., Japan is often hit by earthquakes) and lexical rewrites[10] (e.g., *ad infinitum* means over and over).

Most of our questions need simple facts that should be easily retrievable from any knowledge-base/textual corpora. On an average, each question needed 1.16 additional facts ignoring any linguistic variations. Despite the simplicity of the knowledge needed for these questions, as we show

---

empirically, most baseline approaches achieve a relatively low score on this dataset (even when the core fact is provided). We claim that this is due to the fact that the reasoning needed to answer these questions is non-trivial. Table 3 shows few questions with the associated facts and high-level reasoning needed to answer these questions. Assuming a model can extract the described relations (e.g. defn, contains), the QA system still needs to be able to chain these facts together, identify the resulting relation and verify its expression for each choice. In the extreme case (as shown in the last example), even though only one additional fact is needed to answer the question, it needs a system to *apply* the core "general" science fact to a "specific" situation.

4 Baseline Models

We evaluate the performance of several baselines systems on the Dev and Test subsets of Open-BookQA. For each question, a solver receives 1 point towards this score if it chooses the correct answer, and $1/k$ if it reports a k-way tie that includes the correct answer. The "Guess All" baseline, which always outputs a 4-way tie, thus achieves a score of 25%, same as the expected performance of a uniform random baseline.

4.1 No Training, External Knowledge Only

Since OpenBookQA is a set of elementary level science questions, one natural baseline category is existing systems that have proven to be effective on elementary- and middle-school level science exams. These pre-trained systems, however, rely only on their background knowledge and do not take the set $\mathcal{F}$ of core facts into account. Further, their knowledge sources and retrieval mechanism are close to those used by the IR solver that, by design, is guaranteed to fail on OpenBookQA. These two aspects place a natural limit on the effectiveness of these solvers on OpenBookQA, despite their excellent fit for the domain of multiple-choice science questions. We consider four such solvers.

PMI (Clark et al., 2016) uses pointwise mutual information (PMI) to score each answer choice using statistics based on a corpus of 280 GB of plain text. It extracts unigrams, bigrams, trigrams, and skip-bigrams from the question q and each answer choice c_i. Each answer choice is scored based on the average PMI across all pairs of question and answer n-grams.

TableILP (Khashabi et al., 2016) is an Integer Linear Programming (ILP) based reasoning system designed for science questions. It operates over semi-structured relational tables of knowledge. It scores each answer choice based on the optimal (as defined by the ILP objective) "support graph" connecting the question to that answer through table rows. The small set of these knowledge tables, however, often results in missing knowledge, making TableILP not answer 24% of the OpenBookQA questions at all.

TupleInference (Khot et al., 2017), also an ILP-based QA system, uses Open IE tuples (Banko et al., 2007) as its semi-structured representation. It builds these subject-verb-object tuples *on-the-fly* by retrieving text for each question from a large corpus. It then defines an ILP program to combine evidence from multiple tuples.

DGEM (Khot et al., 2018) is a neural entailment model that also uses Open IE to produce a semi-structured representation. We use the adaptation of this model to multiple-choice question answering proposed by Clark et al. (2018), which works as follows: (1) convert q and each c_i into a hypothesis, h_i, and each retrieved fact into a premise p_j; and (2) return the answer choice with the highest entailment score, $\arg\max_i e(p_j, h_i)$.

4.2 No Training; $\mathcal{F}$ and Extr. Knowledge

We also consider providing the set $\mathcal{F}$ of core facts to two existing solvers: the **IR** solver of Clark et al. (2016) (to assess how far simple word-overlap can get), and the **TupleInference** solver.

4.3 Trained Models, No Knowledge

We consider several neural baseline models that are trained using Train set of OpenBookQA. For ease of explanation, we first define the notation used in our models. For a given question $q_{mc} = (q, \{c_1, c_2, c_3, c_4\})$, we define the set of token sequences , $\mathcal{S} = \{q, c_1, c_2, c_3, c_4\}$. For each token sequence $s \in \mathcal{S}$, w_j^s is the jth and $e_j^s = \mathrm{Emb}(w_j^s)$ is the embedding for this token. We use n_s to indicate the number of tokens in s and d for the dimensionality of the embeddings.[11] We model multiple-choice QA as multi-class classification: Given q_{mc}, predict one of four class labels $L =$

---

[11] For all experiments we use $d = 300$ *GloVe* (Pennington et al., 2014) embeddings pre-trained on 840B tokens from *Common Crawl* (https://nlp.stanford.edu/projects/glove/).

Question	Science Fact	Common Knowledge (Type)	Reasoning Challenge
What is the most likely to be an effect of acid rain on an aquatic environment? (A) increase in plant growth (B) increase in fish population **(C) decrease in plant life** (D) cleaner and clearer water	acid rain has a negative impact on water quality	decrease in water quality leads to a decrease in aquatic life (CAUSAL)	causes(x, y) $\wedge$ causes(y, z) $\Rightarrow$ causes(x, z)
The moon's surface (A) is smooth on the entire surface (B) contains an internal core of cheese (C) is filled with lakes **(D) contains large cavities cause by explosions**	the moon's surface contains many craters	Craters are large cavities caused by explosions (DEFINITION)	contains(x, y) $\wedge$ defn(y, z) $\Rightarrow$ contains(x, z)
As a car approaches you in the night (A) the headlights remain at a constant (B) the headlights turn off **(C) the headlights become more intense** (D) the headlights recede into the dark	as a source of light becomes closer, that source will appear brighter	Headlights of a car are source of light (PROPERTY)	[lhs $\Rightarrow$ rhs] $\Rightarrow$ [ground(lhs) $\Rightarrow$ ground(rhs)]

Table 3: Example training questions (with their correct choices **marked**) along with the facts and reasoning needed. In the last example, the science fact states that lhs="source of light becomes closer" implies rhs="source will appear brighter". Grounding this rule based on the common-knowledge fact, produces a new rule: "As headlights of the car come closer, headlights will appear brighter"

$\{1, 2, 3, 4\}$, where the true label is the correct answer index.

Embeddings + Similarities as Features. We first experiment with a simple logistic regression model (Mihaylov and Nakov, 2016; Mihaylov and Frank, 2016, 2017) that uses centroid vectors r_s^{emb} of the word embeddings of tokens in s, and then computes the cosine similarities between the question and each answer choice, r_{q,c_i}^{cos}:

$$r_s^{\text{emb}} = \frac{1}{n_s} \sum_{j=1}^{n_s} e_{s_j} \in \mathbb{R}^d$$

$$r_{q,c_i}^{\text{cos}} = \cos(r_q^{\text{emb}}, r_{c_i}^{\text{emb}}) \in \mathbb{R}^1$$

For each training instance, we build a feature representations $\vec{f}$ by concatenating these vectors and train an L2 logistic regression classifier:

$$\vec{f} = [r_q^{\text{emb}}; r_{c_{1..4}}^{\text{emb}}; r_{q,c_{1..4}}^{\text{cos}}] \in \mathbb{R}^{5d+4}$$

BiLSTM Max-Out Baselines. As a simple neural baseline, we adapt *BiLSTM max-out* model (Conneau et al., 2017) to our QA task. That is, we first encode the question tokens and choice tokens $w_{1...n_s}^s$, independently with a bi-directional context encoder (LSTM) to obtain a context (ctx) representation $h_{s_{1...n_s}}^{\text{ctx}} = \text{BiLSTM}(e_{1...n_s}^s) \in \mathbb{R}^{n_s \times 2h}$ Next, we perform an element-wise aggregation operation $\max$ on the encoded representations $h_{s_{1..n_s}}^{\text{ctx}}$ to construct a single vector:

$$r_s^{\text{ctx}} = \max(h_{s_{1..n_s}}^{\text{ctx}}) \in \mathbb{R}^{2h}. \quad (1)$$

Given the contextual representations for each token sequence, we experiment with three configurations for using these representations for QA:

(a) Plausible Answer Detector. This baseline goes to the extreme of completely ignoring q and trying to learn how plausible it is for c_i to be the correct answer to *some* question in this domain. This captures the fact that certain choices like 'a magical place' or 'flying cats' are highly unlikely to be the correct answer to a science question without negation (which is the case for OpenBookQA).

We implement a plausible answer detector using a *choice-only* model for predicting the answer by obtaining a score α_{c_i} as: $\alpha_{c_i} = W_c^T r_{c_i}^{\text{ctx}} \in \mathbb{R}^1$, where $W_c^T \in \mathbb{R}^{2h}$ is a weights vector optimized during training, $i = \{1..4\}$ is the index of the choice. To obtain the answer choice from the set of choice scores $\alpha_{c_{1..4}}$ using $\arg\max(\text{softmax}(\alpha_{c_{1..4}}))$, where $\text{softmax}(\alpha_{c_i}) = \frac{\exp(\alpha_{c_i})}{\sum_{j=1}^{4} \exp(\alpha_{c_j})}$ as usual.

(b) Odd-One-Out Solver. It considers all 4 answer options jointly and selects the one that is least similar to the others. This captures bias in human authored questions arising from the fact that creating good quality incorrect answers is difficult. Workers generally start with the correct answer, and then come up with three incorrect ones. The latter often tend to be homogeneous or share other common properties (e.g., non-scientific terms) uncharacteristic of the correct answer.

We implement this using a *choice-to-choices* attention model. For each choice c_i, we calculate the attention to the other choices as α_{c_i,c_j}. We then sum these attention values to compute the attention for c_i to the rest of the choices, $\alpha_{c_i 2 c_{r(est)}}$, and return the choice with the lowest sum. The atten-

tion is computed as $\alpha_{c_i,c_j} = \text{Att}(r_{c_i}^{\text{ctx}}, r_{c_j}^{\text{ctx}})$ where

$$\text{Att}(u, v) = W^T([u; v; u \cdot v; |u - v|]) \in \mathbb{R}^1$$

is a linear attention function and $W \in \mathbb{R}^{8h}$ is a weight vector. We then compute $\alpha_{c_i 2 c_{r(est)}} = \sum_{j=1}^{4} \alpha_{c_i,c_j}$ ($j \neq i$) and select the answer with the index $a_{c2c_r} = \arg\min(\text{softmax}(\alpha_{c_{1..4}2c_r}))$.

(c) Question Match. This solver tries to predict which choice best matches the question (Nakov et al., 2016), without relying on external knowledge. To achieve that, we compute an attention score α_{q,c_i} between q and each of the choices q_i as $\alpha_{q,c_i} = \text{Att}(r_q^{\text{ctx}}, r_{c_i}^{\text{ctx}})$, and select the one with the highest score. We also experiment with a model where r_q^{ctx} and $r_{c_i}^{\text{ctx}}$ are obtained using token-wise interaction proposed in ESIM (Chen et al., 2017b).

4.4 Trained Model with External Knowledge

Lastly, we implement a two stage model for incorporating external common knowledge, K. The first module performs information retrieval on K to select a fixed size subset of potentially relevant facts $K_{Q,C}$ for each instance in the dataset (see Appendix A). The second module is a neural network that takes $(Q, C, K_{Q,C})$ as input to predict the answer $a_{q,c}$ to a question Q from the set of choices C.

Knowledge-Enhanced Reader. As a base knowledge-aware model, we use a variant of the model of Mihaylov and Frank (2018), implemented by extending our **BiLSTM max-out question-match baseline (c)**. For each instance the model reads the question q and answers $c_{1..4}$ independently and attends to the set of retrieved external knowledge facts $K_{Q,C}$. We encode each fact k_j from $K_{Q,C} = k_{1..N_k}$ (N_k is the number of facts) with same BiLSTM as used for q and $c_{1..4}$ and construct a single vector $r_{k_j}^{\text{ctx}} \in \mathbb{R}^{2h}$ using Eq. 1. Having such representations for each k_j results in knowledge memory matrix $M_k = r_{k_{1..N_k}}^{\text{ctx}} \in \mathbb{R}^{N_k \times 2h}$. Note that M_k is dynamic memory, specific for each instance in the batch and is encoded in each step during training. This memory is used to calculate a knowledge-aware representation, $r_s^{\text{kn}} = \sum((M_k^T r_s^{\text{ctx}}).M_k) \in \mathbb{R}^{2h}$. Each context (ctx) representation r_s^{ctx} ($s \in \mathcal{S}$) is combined with r_s^{kn} to obtain a knowledge-enhanced representation $r_s^{\text{ctx+kn}} = (r_s^{\text{ctx}} + r_s^{\text{kn}})/2$. We then model the knowledge-enhanced attention $\alpha_{q,c_j}^{\text{kn}}$ between

Solver	Dev	Test
Human solver	89.3*	91.7*
Guess All ("random")	25.0	25.0
NO TRAINING, KB ONLY (§4.1)		
TupleInference	15.9	17.9
PMI (Waterloo corpus)	19.7	21.2
TableILP	20.0	23.4
DGEM	27.4	**24.4**
NO TRAINING, KB + $\mathcal{F}$ (§4.2)		
IR with $\mathcal{F}$	25.5	24.8
TupleInference with $\mathcal{F}$	23.6	**26.6**
DGEM with $\mathcal{F}$	28.2	24.6
TRAINED MODELS, NO $\mathcal{F}$ OR KB (§4.3)		
Embedd+Sim	44.6	41.8
ESIM	53.9±0.4	48.9±1.1
Plausible Answer Detector	54.4±0.7	49.6±0.7
Odd-one-out Solver	56.9±0.5	50.2±1.6
Question Match	54.6±1.2	**50.2±0.9**
ORACLE MODELS, $\mathcal{F}$ AND/OR KB (§4.4)		
f	63.0±2.3	55.8±2.3
f + WordNet	57.6±1.4	56.3±1.3
f + ConceptNet	57.0±1.6	53.7±1.5
f + k	80.2±1.1	76.9±0.7

Table 4: Scores obtained by various solvers on Open-BookQA, reported as a percentage ± the standard deviation across 5 runs with different random seeds. Other baselines are described in the corresponding referenced section. For oracle evaluation, we use the gold science fact f associated with each question, and optionally the additional fact k provided by the question author. Bold denotes the best Test score in each category.

q and c_i as a linear combination of the ctx, kn and ctx + kn representations as

$$\alpha_{q,c_i} = W^T[\text{Att}(r_s^{\text{ctx}}, r_{c_i}^{\text{ctx}}); \text{Att}(r_s^{\text{kn}}, r_{c_i}^{\text{kn}});$$
$$\text{Att}(r_s^{\text{ctx+kn}}, r_{c_i}^{\text{ctx}}); \text{Att}(r_s^{\text{ctx}}, r_{c_i}^{\text{ctx+kn}});$$
$$\text{Att}(r_s^{\text{ctx}}, r_{c_i}^{\text{kn}}); \text{Att}(r_s^{\text{kn}}, r_{c_i}^{\text{ctx}});$$
$$\text{Att}(r_s^{\text{ctx+kn}}, r_{c_i}^{\text{kn}}); \text{Att}(r_s^{\text{kn}}, r_{c_i}^{\text{ctx+kn}});$$
$$\text{Att}(r_s^{\text{ctx+kn}}, r_{c_i}^{\text{ctx+kn}})],$$

where $W \in \mathbb{R}^9$ is a weight vector initialized with the *ones* vector and optimized during training. We then select the answer c_i with the highest score.

5 Baseline Performance

The results for various baseline models are summarized in Table 4, grouped by method category. We make a few observations:

First, the task is **largely solvable by a layperson**, as evidenced by the 92% score of crowdworkers. This is measured as described in Sec-

tion 3.2. We use annotations from Step 6 of the question generation process and report $\tilde{H}(Q)-3\%$ as a conservative lower estimate. As an additional assessment, we also obtained 5 *new* annotations for 100 randomly chosen questions from each of Train, Dev, and Test sets. The performance remained similar at 88.6%, 90.2%, and 91.6%, resp.

The **second group** shows that pre-trained state-of-the-art solvers for multiple-choice science questions perform poorly. One explanation is their correlation with the the IR method used for question filtering, as mentioned in Section 4.1.

The **third group** of results suggests that adding $\mathcal{F}$ to pre-trained models has a mixed effect, improving TupleInference by 8.7% but not changing DGEM.[12] Unlike DGEM, TupleInference relies on brittle word-overlap similarity measures very similar to the ones used by IR. Since IR (KB) gets 0% by design, TupleInference (KB) also has poor performance and adding $\mathcal{F}$ helps it find better support despite the brittle measures.

The **fourth group** demonstrates that carefully designed trainable neural models—even if simplistic and knowledge-free—can be surprisingly powerful. For example, the "plausible answer detector" can predict the correct answer with 49.6% accuracy without even looking at the question. The "odd-one-out" solver, by considering other answer choices, raises this to 50.2%. The "question match" solver, which simply compares the BiLSTM max-out encoding of the question with that of various answer choices, also achieves 50.2%.[13] Similar findings have been reported for several recent datasets (Gururangan et al., 2018), making it imperative to perform such tests early.

Interestingly, all of these neural knowledge-free baselines simultaneously succeed on 34.4% of the Dev questions, and simultaneously fail on 23.6%. For **Question Match** and **ESIM** we also experiment with ElMo (Peters et al., 2018) which improved their score on Test with 0.4% and 1.8%.

The **final group** demonstrates the need for external knowledge and deeper reasoning. When the "oracle" science fact f used by the question author is provided to the knowledge-enhanced reader,

it improves over the knowledge-less models by about 5%. However, there is still a large gap, showing that the core fact is insufficient to answer the question. When we also include facts retrieved from WordNet (Miller et al., 1990), the score improves by about 0.5%. Unlike the WordNet gain, adding ConceptNet (Speer et al., 2017) introduces a distraction and reduces the score. This suggests that ConceptNet is either not a good source of knowledge for our task, or only a subset of its relations should be considered. Overall, external knowledge helps, although retrieving the right bits of knowledge remains difficult. In the last row of Table 4, we use the oracle core fact along with question author's interpretation of the additional fact k. This increases the scores substantially, to about 76%. This big jump shows that improved knowledge retrieval should help on this task. At the same time, we are still not close to the human performance level of 92% due to various reasons: (a) the additional fact needed can be subjective, as hinted at by our earlier analysis; (b) the authored facts $\mathcal{K}$ tend to be noisy (incomplete, over-complete, or only distantly related), also as mentioned earlier; and (b) even given the true gold facts, performing reliable "reasoning" to link them properly remains a challenge.

Sample predictions and analysis of questions from Dev are provided in Appendix D.

6 Conclusion

We present a new dataset, OpenBookQA, of about 6000 questions for open book question answering. The task focuses on the challenge of combining a corpus of provided science facts (open book) with external broad common knowledge. We show that this dataset requires simple common knowledge beyond the provided core facts, as well as multi-hop reasoning combining the two. While simple neural methods are able to achieve an accuracy of about 50%, this is still far from the human performance of 92% on this task. We leave closing this gap for future research, and illustrate, via oracle-style experiments, the potential of better retrieval and reasoning on this task.

---

[12]By design, IR with its default corpus gets 0% on Open-BookQA. Hence we don't consider the effect of adding $\mathcal{F}$, which appears artificially magnified.

[13]This model also achieves the current best score, 33.87%, on the ARC Reasoning Challenge (Clark et al., 2018). When adapted for the textual entailment task by comparing BiLSTM max-out encodings of premise and hypothesis, it achieves 85% on the SciTail dataset (Khot et al., 2018).

References

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

D. Chen, J. Bolton, and C. D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *ACL*, pages 2358–2367.

D. Chen, A. Fisch, J. Weston, and A. Bordes. 2017a. Reading wikipedia to answer open-domain questions. In *ACL*.

Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. 2017b. Enhanced lstm for natural language inference. In *ACL*, pages 1657–1668.

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

P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. D. Turney, and D. Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*, pages 2580–2586.

A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, pages 670–680.

M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*.

K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

F. Hill, A. Bordes, S. Chopra, and J. Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *ICLR*.

W. Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30.

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

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

T. Jenkins. 1995. Open book assessment in computing degree programmes 1. Technical Report 95.28, University of Leeds.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pages 1601–1611.

A. Kembhavi, M. J. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, pages 5376–5384.

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

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

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

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

D. P. Kingma and J. L. Ba. 2015. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015*, pages 1–15.

T. Kociský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette. 2017. The NarrativeQA reading comprehension challenge. *CoRR*, abs/1712.07040.

J. Landsberger. 1996. Study guides and strategies. Http://www.studygs.net/tsttak7.htm.

T. Mihaylov and A. Frank. 2016. Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. In *CoNLL-16 shared task*, pages 100–107.

T. Mihaylov and A. Frank. 2017. Story Cloze Ending Selection Baselines and Data Examination. In *LSD-Sem Shared Task*.

T. Mihaylov and A. Frank. 2018. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *ACL*, pages 821–832.

T. Mihaylov and P. Nakov. 2016. SemanticZ at SemEval-2016 Task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *SemEval '16*.

G. A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to WordNet: An online lexical database. *International Journal of Lexicography*, 3(4):235–244.

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

N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. 2016. A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. In *NAACL*.

P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, a. A. Freihat, J. Glass, and B. Randeree. 2016. Semeval-2016 task 3: Community question answering. In *SemEval '16*, pages 525–545.

T. Onishi, H. Wang, M. Bansal, K. Gimpel, and D. McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *EMNLP*, pages 2230–2235, Austin, Texas.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

J. Pennington, R. Socher, and C. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

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

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392.

M. Richardson, C. J. Burges, and E. Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203.

P. Singh, T. Lin, E. Mueller, G. Lim, T. Perkins, and W. Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *Lecture Notes in Computer Science*, volume 2519, pages 1223–1237.

R. Speer, J. Chin, and C. Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

K. Stasaski and M. A. Hearst. 2017. Multiple choice question generation utilizing an ontology. In *BEA@EMNLP, 12th Workshop on Innovative Use of NLP for Building Educational Applications*.

S. Sugawara, H. Yokono, and A. Aizawa. 2017. Prerequisite skills for reading comprehension: Multiperspective analysis of mctest datasets and systems. In *AAAI*, pages 3089–3096.

A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

P. D. Turney. 2017. Leveraging term banks for answering complex questions: A case for sparse vectors. *CoRR*, abs/1704.03543.

D. Weissenborn, G. Wiese, and L. Seiffe. 2017. Making neural qa as simple as possible but not simpler. In *CoNLL*, pages 271–280.

J. Welbl, P. Stenetorp, and S. Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Y. Zhang, H. Dai, K. Toraman, and L. Song. 2018. KGˆ2: Learning to Reason Science Exam Questions with Contextual Knowledge Graph Embeddings. In *arXiv*.