## A Knowledge Retrieval Module

This module is the first part of a two stage model for incorporating knowledge from an external source $K$. For each instance $(q, C)$ in the dataset, where $q$ is a question and $C = \{c_1, \ldots, c_4\}$ a set of answer choices, it performs information retrieval (IR) on $K$ to select a fixed size subset $K_{q,C}$ of potentially relevant facts. The second module is a neural network that takes $(q, C, K_{q,C})$ as input, and predicts the answer $a_{q,C}$.

For the IR module, we use TfIdfVectorizer[14] to build vector representations $\vec{q}_{\text{tfidf}}$, $\vec{c^i}_{\text{tfidf}}$ and $\vec{k}_{\text{tfidf}}$ for the question $q$, choice $c_i \in C$, and fact $k \in K$ based on the tokens in the training set. We then calculate similarity scores $t_{q,k}$ and $t_{q,c_i,k}$ between $q$ and $c_i$, resp., and each of the external facts in $k \in K$:

$$t_{q,k} = 1 - \text{sim}(\vec{q}_{\text{tfidf}}, \vec{k}_{\text{tfidf}})$$
$$t_{q,c_i,k} = 1 - \text{sim}(\vec{c^i}_{\text{tfidf}}, \vec{k}_{\text{tfidf}}) \cdot t_{q,k},$$

where sim is implemented as cosine distance. Based on these similarity scores, we obtain a set $K_{q,C}$ of facts for each $(q, C, K)$ as $K_q \cup \bigcup_i K_{q,c_i}$, where $K_q$ and $K_{q,c_i}$ are the top $N_k$ facts each with highest similarity $t_{q,k}$ and $t_{q,c_i,k}$, respectively. $N_k$ is a hyper-parameter chosen from $\{5, 10, 20\}$ so as to yield the best Dev set performance.

For experimentation with knowledge, we consider the 'open book' set of facts $\mathcal{F}$ in conjunction with two sources of common knowledge: the Open Mind Common Sense (Singh et al., 2002) part of ConceptNet (Speer et al., 2017), and its WordNet (Miller, 1995) subset.

## B Implementation and Training

Our neural models are implemented with *AllenNLP*[15] (Gardner et al., 2017) and *PyTorch*[16] (Paszke et al., 2017). We use *cross-entropy* loss and the *Adam* optimizer (Kingma and Ba, 2015) with initial learning rate 0.001. For the neural models *without* external knowledge, we typically train the model with a maximum of 30 epochs and stop training early if the Dev set accuracy does not improve for 10 consecutive epochs. We also halve the learning rate if there is no Dev set improvement for 5 epochs. For the neural models *with* external knowledge, we typically train for 60 epochs with a patience of 20 epochs. For most of our neural models, we use $h = 128$ as the *LSTM* hidden layer size. The embedding dropout rate is chosen from $\{0.1, 0.2, 0.5\}$, again based on the best Dev set performance.

For each model configuration, we perform 5 experiments with different random seeds. For each run, we take the model with the best performance on Dev and evaluate on Test. We report the average accuracy for the best Dev score and the average of the corresponding Test score $\pm$ the standard deviation across the 5 random seeds.

The code for the models and the configuration files required for reproducing the results are available at http://data.allenai.org/OpenBookQA.

[14] Term frequency, Inverse document frequency based vectorizer from *scikit-learn* (Pedregosa et al., 2011).

[15] https://allennlp.org

[16] https://pytorch.org

## C Additional Experiments

### C.1 Question Answering: ARC

We also perform experiments with the **Question Match** system on the Challenge (hard) set of the AI2 Reasoning Challenge or ARC (Clark et al., 2018). We train several models with different LSTM hidden sizes (128, 256, **384 (best)**, 512), and dropout of the embedding layer (**0.0 (best)**, 0.2, 0.5) on the questions from the Challenge Train set and take the model that has the highest accuracy on the Dev set. The resulting system scores 33.87% on the Challenge Test set, which is 2.17% higher than the previous best score by Zhang et al. (2018). The code and model configuration are available at https://github.com/allenai/ARC-Solvers.

### C.2 Textual Entailment: SciTail

We perform textual entailment experiments on the Science enTailment dataset SciTail (Khot et al., 2018). We change the **Question Match** model to a classic **BiLSTM Max-Out** (Conneau et al., 2017) for textual entailment, by replacing the question $q$ and a choice $c_i$ with the premise $p$ and the hypothesis $h$, resp., and perform binary classification on the entailment labels (Entail, Neural). We run experiments with BiLSTM encoders with LSTM hidden size of 384 and share the encoder parameters between the premise and the hypothesis. Without additional hyper-parameter tuning, this yields entailment accuracy scores of 87.9% and 85.4% on the Dev and Test sets, respectively.

## D Success and Failure Examples

We give some examples of questions that were answered correctly/incorrectly by various groups of models. We include here the first three questions in each case.

### D.1 Neural Baseline Successes

We begin with three examples of questions that all neural models without external knowledge (namely Question Match, Plausible Answer, One-Odd-Out, and ESIM from the fourth group in Table 5) predicted correctly.

| |
|---|
| A body may find its temperature to be lowered after (A) water is heated up (B) **fluid spreads from pores** (C) the air becomes arid (D) the sky stays bright |
| Oil is a non-renewable resource which tells us that when (A) it can be remade (B) it can be found in other places (C) there is an endless supply (D) **the final barrel is gone, there supply is finished** |
| Magma contains (A) **particles of iron** (B) Loads of leaves (C) Soda (D) Silly Putty |

Table 5: Sample questions predicted **correctly** (172/500) by all trained neural models without external knowledge.

In these examples, we observe that the correct answer usually contains a word that is semantically closer (than words in other answer choices) to an important word from the question: *pores* to *body*; *non-renewable* (negative sentiment) to *gone, finished* (also negative sentiment); *iron* to *magma (liquid rock)*.

### D.2 Neural Baseline Failures, Oracle Success

Table 6 shows example questions (with the Oracle facts) from the Dev set that were predicted correctly by the $f + k$ Oracle model (405/500) but incorrectly by all of the 4 neural models without knowledge (69/405). In contrast to Table 5, a simple semantic similarity is insufficient. The questions require chaining of multiple facts in order to arrive at the correct answer.

### D.3 Neural Baseline and Oracle Failures

42/500 questions in the Dev set were predicted incorrectly by all models without external knowledge, as well as by the Oracle $f + k$ model. In Table 7 we show 3 such questions. In all cases, the Oracle $f + k$ model made an incorrect prediction with confidence higher than 0.9.

| |
|---|
| Frilled sharks and angler fish live far beneath the surface of the ocean, which is why they are known as (A) **Deep sea animals** (B) fish (C) Long Sea Fish (D) Far Sea Animals. **Oracle facts:** ($f$) deep sea animals live deep in the ocean. ($k$) Examples of deep sea animals are angler fish and frilled sharks. |
| Gas can fill any container it is given, and liquid (A) is standard weight and size (B) is the opposite of variable (C) only needs a few (D) **uses what it needs**. **Oracle facts:** ($f$) Matter in the liquid phase has definite volume. ($k$) liquid cannot spread endlessly. |
| When birds migrate south for the winter, they do it because (A) **they are genetically called to** (B) their children ask for them to (C) it is important to their happiness (D) they decide to each year. **Oracle facts:** ($f$) migration is an instinctive behavior. ($k$) instinctive is genetic. |

Table 6: Sample questions predicted **correctly** by the $f+k$ Oracle model (405/500) but were predicted incorrectly by all of the 4 neural models without knowledge (total of 69 out of 405).

As noted earlier, there are several broad reasons why even this so-called oracle model fails on certain questions in OpenBookQA. In some cases, the core fact $f$ associated with a question $q$ isn't actually helpful in answering $q$. In many other cases, the corresponding second fact $k$ is noisy, incomplete, or only distantly related to $q$. Finally, even if $f$ and $k$ are sufficient to answer $q$, it is quite possible for this simple model to be unable to perform the reasoning that's necessary to combine these two pieces of textual information in order to arrive at the correct answer.

In the shown examples, the first question falls outside the domain of *Science* where most of the core facts come from. The scientific fact "($f$) An example of collecting data is measuring" is transformed into a question related to the law and judicial domain of *collecting data for a (court) case*. This is an indication that the model trained on the Train set does not perform well on distant domains, even if the core facts are provided.

In the second question, we have an option *all of these*. Indeed, the selected answer seems the most relevant (a generalized version of the other two), but the model did not know that if we have an option *all of these* and all answers are plausible,

| |
|---|
| An example of data collection is: (A - 0.9977) Deleting case files on the computer, (B - 0.0000) Touching evidence without gloves, (C - 0.0004) **speaking with a witness**, (D - 0.0019) Throwing documents in the trash. **Oracle facts**: ($f$) An example of collecting data is measuring. ($k$) Interviews are used to collect data. |
| If a farmland up the hill gets rainfall, what could happen to lower lands? (A - 0.0005) **all of these**, (B - 0.0245) they could get fertilizer washed to them, (C - 0.9542) they could experience unfavorable chemical change in their lands, (D - 0.0208) they could have their lands poisoned. **Oracle facts**: ($f$) runoff contains fertilizer from cropland. ($k$) fertilizers for certain crops could poison other crops or soil types. |
| Layers of the earth include all but: (A - 0.0429) mantle, (B - 0.0059) **center**, (C - 0.0334) crust, (D - 0.9177) inner core. **Oracle facts**: ($f$) the crust is a layer of the Earth. ($k$) the last layer is the outer core. |

Table 7: Sample questions predicted incorrectly by all models models w/o knowledge, as well as the $f + k$ Oracle model, even though the Oracle model has confidence higher than 0.90.

it should decide if all answers are correct and not pick the "most likely" individual answer.

The third question again requires the model to select a special type of aggregate answer ("all but xyz"), but the related Oracle facts are pointing to a specific answer.