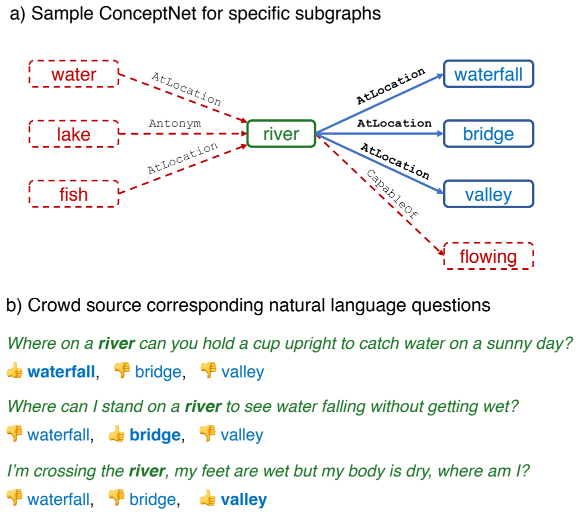
# COMMONSENSEQA: A Question Answering Challenge Targeting Commonsense Knowledge

**Alon Talmor**[*,1] **Jonathan Herzig**[*,1] **Nicholas Lourie**[2] **Jonathan Berant**[1,2]

[1]School of Computer Science, Tel-Aviv University
[2]Allen Institute for Artificial Intelligence
{alontalmor@mail,jonathan.herzig@cs,joberant@cs}.tau.ac.il,nicholasl@allenai.org

## Abstract

When answering a question, people often draw upon their rich world knowledge in addition to some task-specific context. Recent work has focused primarily on answering questions based on some relevant document or content, and required very little general background. To investigate question answering with prior knowledge, we present COMMONSENSEQA: a difficult new dataset for commonsense question answering. To capture common sense beyond associations, each question discriminates between three target concepts that all share the same relationship to a single source drawn from CONCEPTNET (Speer et al., 2017). This constraint encourages crowd workers to author multiple-choice questions with complex semantics, in which all candidates relate to the subject in a similar way. We create 9,500 questions through this procedure and demonstrate the dataset's difficulty with a large number of strong baselines. Our best baseline, the OpenAI GPT (Radford et al., 2018), obtains 54.8% accuracy, well below human performance, which is 95.3%.



*Where on a **river** can you hold a cup upright to catch water on a sunny day?*
👍 **waterfall**, 👎 bridge, 👎 valley

*Where can I stand on a **river** to see water falling without getting wet?*
👎 waterfall, 👍 **bridge**, 👎 valley

*I'm crossing the **river**, my feet are wet but my body is dry, where am I?*
👎 waterfall, 👎 bridge, 👍 **valley**

Figure 1: (a) One source concept and three target concepts are sampled from CONCEPTNET (b) Crowdsourcing workers generate three questions, one per target concept, where for each question one target concept is the answer (thumbs up), while the other target concepts are not (thumbs down).

## 1 Introduction

When humans answer questions, they capitalize on their common sense and background knowledge about spatial relations, causes and effects, scientific facts and social conventions. For instance, given the question *"Where was Simon when he heard the lawn mower?"*, one can infer that the lawn mower is close to Simon, and that it is probably outdoors and situated at street level. This type of knowledge seems trivial for humans, but is still out of the reach of current natural language understanding (NLU) systems.

Recent work on Question Answering (QA) has mostly focused on factoid QA (Hermann et al., 2015; Rajpurkar et al., 2016; Nguyen et al., 2016; Joshi et al., 2017), where an answer is extracted from a textual context using relatively little external knowledge. Other small benchmarks, such as the Winograd Scheme Challenge (Levesque, 2011) and COPA (Roemmele et al., 2011), targeted common sense more directly, but have been difficult to collect at scale. Recently, larger datasets, such as SWAG, tackled commonsense knowledge about situations (Zellers et al., 2018). Such lines of work advance research on common sense, but do not capture the full breadth of commonsense types employed by humans.

Moreover, it has become increasingly evident recently (Poliak et al., 2018; Gururangan et al., 2018), that dataset generation creates annotation artifacts that are difficult to remove and are specific to the annotation process. In particular, large pre-trained language models, that are fine-tuned

[*] The authors contributed equally

on a target task, can obtain surprisingly high performance on datasets such as SWAG and GLUE (Devlin et al., 2018; Wang et al., 2018). This emphasizes the need to create multiple data generation procedures for different types of common sense, and generating different types of text distributions, in order to examine the performance of current NLU models.

In this work, we present a method for generating commonsense questions at scale by asking crowd workers to author questions that describe the relation between concepts from CONCEPTNET (Figure 1). A crowd worker observes a source concept (*'River'* in Figure 1) and three target concepts (*'Waterfall'*, *'Bridge'*, *'Valley'*) that are all related by the same CONCEPTNET relation (`AtLocation`). The worker then authors three questions, one per target concept, such that only that particular target concept is the answer, while the other two distractor concepts are not. This primes the workers to add commonsense knowledge to the question, that separates the target concept from the distractors.

In addition, because questions are generated freely by workers, they often require background knowledge that is trivial to humans but is only seldom explicitly reported on the web (*reporting bias*). Thus, questions in COMMONSENSEQA have a different nature compared to questions that are authored given an input text, which is common practice in prior work.

Using our generation framework, we collect 9,500 commonsense natural language questions. We present an analysis of the dataset that illustrates the uniqueness of the gathered questions compared to prior QA datasets, and the types of commonsense skills that are required. We extensively evaluate models on COMMONSENSEQA, experimenting with pre-trained models, pre-trained models that are fine-tuned for COMMONSENSEQA, and RC models that utilize web snippets extracted from Google search on top of the question itself. We find that a fine-tuning the OpenAI GPT (Radford et al., 2018) on COMMONSENSEQA obtains the best performance, reaching an accuracy of 54.8%. This is substantially lower than human performance, which is 95.3%.

To summarize, the contributions of this paper are the following:

1. A new QA dataset centered around common sense, containing 9,500 examples.
2. A new method for generating commonsense questions at scale, using CONCEPTNET.
3. An empirical evaluation of state-of-the-art NLU models on COMMONSENSEQA, showing that humans substantially outperform current models, leaving much room for improvement.

The dataset can be downloaded from https://www.tau-nlp.org/commonsenseqa

## 2 Related Work

Machine common sense, or the knowledge of and ability to reason about an open ended world, has long been acknowledged as a critical component for natural language understanding. Early work sought programs that could reason about an environment in natural language (McCarthy, 1959), or leverage a world-model for deeper language understanding (Winograd, 1972). Many commonsense representations and inference procedures have been explored (McCarthy and Hayes, 1969; Kowalski and Sergot, 1986) and large-scale commonsense knowledge-bases have been developed (Lenat, 1995; Speer et al., 2017). However, evaluating the degree of common sense possessed by a machine remains difficult.

One important benchmark, the Winograd Schema Challenge (Levesque, 2011), asks models to correctly solve paired instances of coreference resolution in which sentences differ only in the substitution of a word or short phrase. While the Winograd Schema Challenge remains a tough dataset, the difficulty of generating examples has led to only a small collection being available for model development.[1] The Choice of Plausible Alternatives (COPA) is a similarly important but small dataset consisting of 500 development and 500 test questions (Roemmele et al., 2011). Each question asks which of two alternatives best reflects a cause or effect relation to the premise. For both datasets, scalability is an issue when evaluating modern methods.

With the recent adoption of crowdsourcing, several larger datasets have emerged, focusing on predicting relations between situations or events in natural language. JHU Ordinal Commonsense Inference requests a label from 1-5 for the plausibility that one situation entails another (Zhang

[1] To the best of our knowledge, about 150 examples available at https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html.

et al., 2017). The Story Cloze Test (also referred to as ROC Stories) pits ground-truth endings to stories against implausible false ones (Mostafazadeh et al., 2016). Interpolating these approaches, Situations with Adversarial Generations (SWAG), asks models to choose the correct description of what happens next after an initial event (Zellers et al., 2018). LM-based techniques achieve very high performance on the Story Cloze Test and SWAG by fine-tuning a pre-trained LM on the target task (Radford et al., 2018; Devlin et al., 2018).

Investigations of commonsense datasets, and of natural language datasets more generally, have revealed the difficulty in creating benchmarks that measure the understanding of a program rather than its ability to take advantage of distributional biases, and to model the annotation process (Gururangan et al., 2018; Poliak et al., 2018). Annotation artifacts in the Story Cloze Test, for example, allow models to achieve high performance while only looking at the proposed endings and ignoring the stories (Schwartz et al., 2017; Cai et al., 2017). Thus, the development of benchmarks for common sense remains a difficult challenge.

Researchers have also investigated question answering that utilizes common sense. Science questions often require common sense, and have recently received attention (Clark et al., 2018; Mihaylov et al., 2018); however, they also need specialized scientific knowledge. Similarly, MCScript asks multiple choice questions based on commonsense inferences about a short passage (Ostermann et al., 2018). In contrast to these efforts, our work studies common sense without requiring additional information. SQUABU[2] created a small hand-curated test of common sense and science questions (Davis, 2016), which are difficult for current techniques to solve. In this work, we create similarly well-crafted questions but at a larger scale.

## 3 Dataset Generation

Our goal is to develop a method for generating questions that can be easily answered by humans without context, and require commonsense knowledge. We generate multiple-choice questions in a process that comprises of the following steps.

1. We extract subgraphs from CONCEPTNET, each with one source concept and three target concepts.

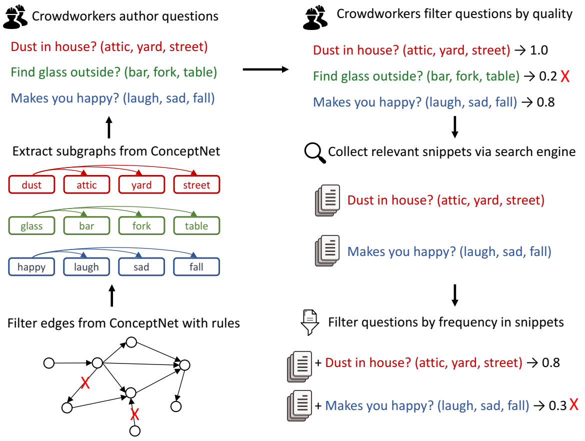

Crowdworkers author questions
Dust in house? (attic, yard, street)
Find glass outside? (bar, fork, table)
Makes you happy? (laugh, sad, fall)
Extract subgraphs from ConceptNet
dust
attic
yard
street
glass
bar
fork
table
happy
laugh
sad
fall
Filter edges from ConceptNet with rules
X
X
Crowdworkers filter questions by quality
Dust in house? (attic, yard, street) → 1.0
Find glass outside? (bar, fork, table) → 0.2 X
Makes you happy? (laugh, sad, fall) → 0.8
Collect relevant snippets via search engine
Dust in house? (attic, yard, street)
Makes you happy? (laugh, sad, fall)
Filter questions by frequency in snippets
+ Dust in house? (attic, yard, street) → 0.8
+ Makes you happy? (laugh, sad, fall) → 0.3 X

Figure 2: COMMONSENSEQA generation process. The input is CONCEPTNET knowledge base, and the output is a set of multiple-choice questions with corresponding relevant context (snippets).

2. We ask crowdsourcing workers to author and verify three questions per sub-graph (one for each target concept).
3. We add textual context to each question by querying a search engine and retrieving web snippets.
4. We prune questions whose answer could be easily detected from the snippets.

The entire data generation process is summarized in Figure 2. We now elaborate on each of the steps:

**Extraction from CONCEPTNET** CONCEPTNET is a graph knowledge-base $G \subseteq \mathcal{C} \times \mathcal{R} \times \mathcal{C}$, where the nodes $\mathcal{C}$ represent natural language concepts, and edges $\mathcal{R}$ represent commonsense relations. Triplets $(c_1, r, c_2)$ carry commonsense knowledge such as '(*gambler*, `CapableOf`, *lose money*)'. CONCEPTNET contains 32 million triplets. To select a subset of triplets for crowdsourcing we take the following steps:

1. We filter triplets with general relations (e.g., `RelatedTo`) or relations that should be easy for NLP models (e.g., `IsA`). In total we use 22 relations.
2. We filter triplets where one of the concepts is more than four words or not in English.
3. We filter triplets where the edit distance between $c_1$ and $c_2$ is low.

This results in a set of 236,208 triplets $(q, r, a)$, where we call the first concept the *question concept* and the second concept the *answer concept*

Our goal is to generate questions that contain the question concept and where the answer is the answer concept. However, to create multiple-

[2] Science questions appraising basic understanding

choice questions we need to choose *distractors* for each question. Choosing distractors uniformly at random from CONCEPTNET is a bad idea, because such distractors are easy to eliminate using simple surface clues.

To remedy this, we propose to create *question sets*: for each question concept $q$ and relation $r$ we group three different triplets $\{(q, r, a_1), (q, r, a_2), (q, r, a_3)\}$ (see Figure 1). This generates three answer concepts that are semantically similar and have a similar relation to the question concept $q$. This primes crowd workers to formulate questions that require background knowledge about the concepts in order to answer the question.

The above procedure generates approximately 130,000 triplets (43,000 question sets), for which we can potentially generate questions.

**Crowdsourcing questions** We used Amazon Mechanical Turk (AMT) workers to generate and validate commonsense questions.

AMT workers saw for every question set the question concept and three answer concepts. They were asked to formulate three questions, where all questions contain the question concept. Each question should have as an answer one of the answer concepts, but not the other two. To encourage workers not to provide simple surface clues for the answer, they were instructed to avoid using words that have a strong relation to the answer concept, for example, not to use the word *'open'* when the answer is *'door'*. Naturally, this is a challenging task, and thus we also employ other pruning mechanisms as shown below.

To validate questions, we train a disjoint group of workers to verify generated questions. Verifiers annotate a question as unanswerable, or choose the right answer. Each question is verified by 2 workers, and only questions verified by at least one worker that answered correctly are used. The verification processes filters out 15% of the questions.

Formulating questions for our task is non-trivial. Thus, we only accept annotators for which at least 75% of the questions they formulate pass all filtering steps. Those include verification as described above and an automatic pruning procedure based on a search engine, described below.

**Adding textual context** To examine whether web text is useful for answering commonsense questions, we add textual information to each question in the following way: We issue a web query to Google search for every question and answer, concatenating the answer to the question, e.g., *'What does a parent tell their child to do after they've played with a lot of toys? + "clean room"'*. We take the first 100 result snippets for each answer term, yielding a context of 300 snippets per question. Using this context, we can investigate the performance of reading comprehension (RC) models on COMMONSENSEQA.

| **Measurement** | **Value** |
|---|---|
| # CONCEPTNET distinct question nodes | 2,242 |
| # CONCEPTNET distinct answer nodes | 9,386 |
| # CONCEPTNET distinct nodes | 9,400 |
| # CONCEPTNET distinct relation lables | 22 |
| Question average length (tokens) | 13.6 |
| Long questions (more than 20 tokens) | 11% |
| Answer average length (tokens) | 1.5 |
| # answers with more than 1 token | 47% |
| # of distinct words in questions | 12,311 |
| # of distinct words in answers | 4,400 |

Table 1: Key statistics for COMMONSENSEQA

**Pruning with a search engine** Harvesting web snippets from a search engine enables filtering out questions that may be solved with surface clues. If the correct answer concept appears frequently with question words, then it might be easy to detect the answer from pure associations.

For each example containing a question, a correct answer, and two distractor answers, we issue a Google query for each answer as described above. We then filter out examples where the number of times the correct answer appears in its resulting snippets is substantially higher than the number of times the distractor answers appear in their resulting snippets on average.

Overall, we generated 9,500 final examples, from a total of 14,991 that were formulated. The total cost per question is $0.33. Table 1 describes key statistics of COMMONSENSEQA.

## 4 Dataset Analysis

**CONCEPTNET concepts** COMMONSENSEQA builds on CONCEPTNET, which contains relations such as ATLOCATION, CAUSES, CAPABLEOF, ANTONYM, etc. We present the main relations along with their frequency in COMMONSENSEQA in Table 2. Question formulators were not shown the CONCEPTNET relation, and therefore they often asked questions that are different from the semantics of the relation. For example, the ques-

| Relation | Formulated question example | % |
|---|---|---|
| AtLocation | *Where would I not want a fox?* **A.** hen house, **B.** england, **C.** mountains, | 46.5 |
| Causes | *What is the hopeful result of going to see a play?* **A.** being entertained, **B.** meet, **C.** sit | 18.7 |
| CapableOf | *Why would a person put flowers in a room with dirty gym socks?* **A.** smell good, **B.** many colors, **C.** continue to grow | 9.6 |
| Antonym | *Someone who had a very bad flight might be given a trip in this to make up for it?* **A.** first class, **B.** reputable, **C.** propitious | 7.7 |
| HasSubevent | *How does a person begin to attract another person for reproducing?* **A.** kiss, **B.** genetic mutation, **C.** have sex | 3.5 |
| HasPrerequisite | *If I am tilting a drink toward my face, what should I do before the liquid spills over?* **A.** open mouth, **B.** eat first, **C.** use glass | 3.2 |
| CausesDesire | *What do parents encourage kids to do when they experience boredom?* **A.** read book, **B.** sleep, **C.** travel | 2.3 |
| Desires | *What do all humans want to experience in their own home?* **A.** feel comfortable, **B.** work hard, **C.** fall in love | 1.8 |
| MotivatedByGoal | *Why do people ready gossip magazines?* **A.** entertained, **B.** get information, **C.** learn | 1.5 |
| HasProperty | *What is a reason to pay your television bill?* **A.** legal, **B.** obsolete, **C.** entertaining | 1.1 |

Table 2: Top ConceptNet relations in CommonsenseQA, along with their frequency in the data and an example question. The first answer (**A**) is the correct answer

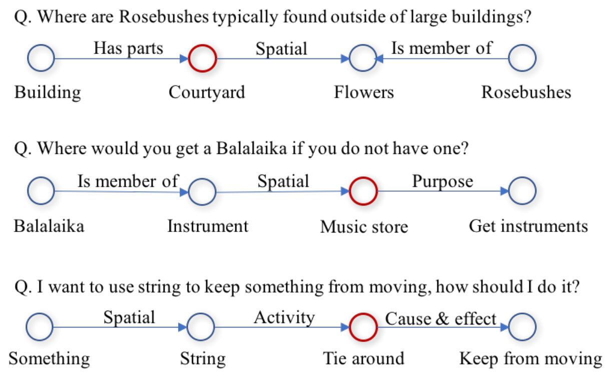


Figure 3: Examples of manually-annotated questions, with the required skills needed to arrive at the answers. The answer concept is the red circle. Skills are presented as labeled edges, and concepts as nodes.

tion *"What do **audiences** clap for?"* was generated from the AtLocation relation, but is focused on the social context of an audience.

Formulators use question concepts to author natural language questions. The top-5 question concepts were person (3.1%), people (2.0%), human (0.7%), water (0.5%) and cat (0.5%).

**Question formulation** Question formulators were instructed to create questions with high language variation. 122 formulators contributed to question generation. However, 10 workers formulated more than 85% of the questions.

We analyzed the distribution of first and second words in the formulated questions along with example questions. Figure 4 presents the breakdown. Interestingly, only 44% of the first words are WH-words. In about 5% of the questions, formulators used first names to create a context story, and in 7% they used the word *"if"* to present a hypothetical question. This suggests high variability in the question language.

**Commonsense Skills** To analyze the types of commonsense knowledge needed to correctly answer questions in CommonsenseQA, we randomly sampled 100 examples from the development set and performed the following analysis.

| Category | Definition | % |
|---|---|---|
| Spatial | Concept A appears near Concept B | 41 |
| Cause & Effect | Concept A causes Concept B | 23 |
| Has parts | Concept A contains Concept B as one of its parts | 23 |
| Is member of | Concept A belongs to the larger class of Concept B | 17 |
| Purpose | Concept A is the purpose of Concept B | 18 |
| Social | It is a social convention that Concept A correlates with Concept B | 15 |
| Activity | Concept A is an activity performed in the context of Concept B | 8 |
| Definition | Concept A is a definition of Concept B | 6 |
| Preconditions | Concept A must hold true in order for Concept B to take place | 3 |

Table 3: Skills and their frequency in the sampled data. As each example can be annotated with multiple skills, the total frequency does does not sum to 100%.

For each question, we explicitly annotated the types of commonsense skills that a human uses to answer the question. We allow multiple commonsense skills per questions, with an average of 1.75 skills per question. Figure 3 provides three example annotations. Each annotation contains a node for the answer concept, and other nodes for concepts that appear in the question or latent concepts. Labeled edges describe the commonsense skill that relates the two nodes. We defined commonsense skills based on the analysis of LoBue and Yates (2011), with slight modifications to accommodate the phenomena in our data. Table 3 presents the skill categories we used, their definition and their frequency in the analyzed examples.

## 5 Baseline Models

Our goal is to collect a dataset of commonsense questions that are easy for humans, but hard for current NLU models. To evaluate this, we experiment with multiple baselines. Table 4 summarizes the various baseline types and characterizes them based on (a) whether training is done on CommonsenseQA or the model is fully pre-trained, and (b) whether context (web snippets) is used.

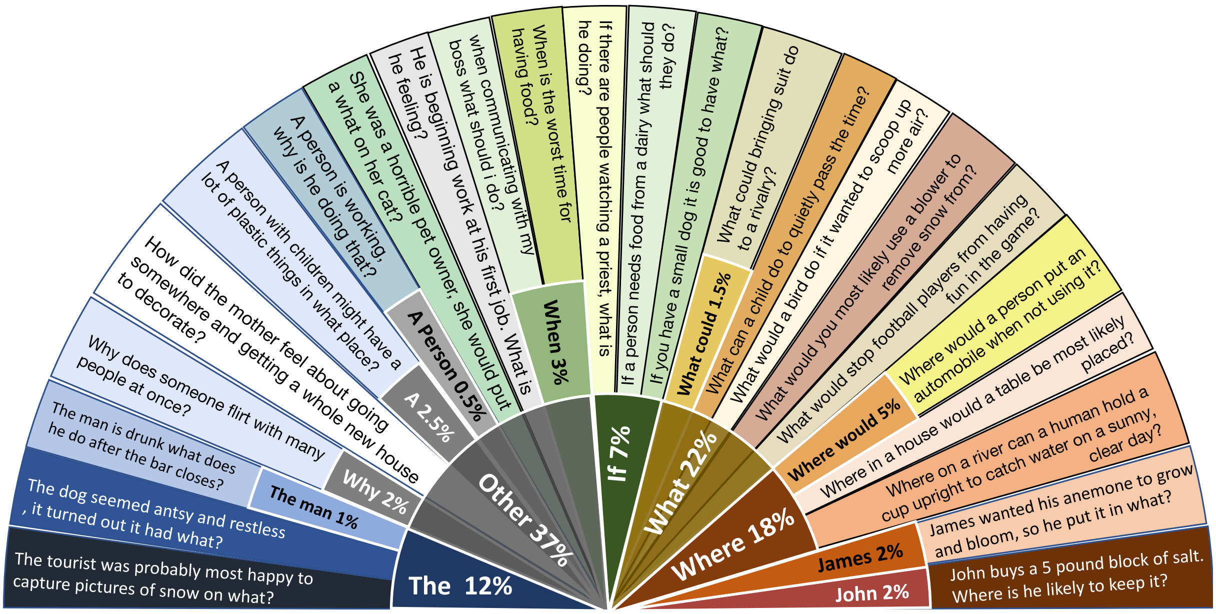


Figure 4: Distribution of the first and second words in questions. The inner part displays words and their frequency and the outer part provides example questions.

| Model | Training | Context |
|---|---|---|
| VECSIM | ✗ | ✗ |
| LM1B | ✗ | ✗ |
| QABILINEAR | ✓ | ✗ |
| QACOMPARE | ✓ | ✗ |
| ESIM | ✓ | ✗ |
| GPT | ✓ | ✗ |
| DOCQA-FIXED | ✗ | ✓ |
| DOCQA-TRAINED | ✓ | ✓ |

Table 4: Our baseline models along with their characteristics. *Training* states whether the model was trained on COMMONSENSEQA, or was only trained a different dataset. *Context* states whether the model uses extra context as input.

We now elaborate on the different baselines.

**a. VECSIM** An unsupervised model that chooses the answer with highest cosine similarity to the question, where the question and answers are represented by an average of pre-trained word embeddings.

**b. LM1B** Inspired by Trinh and Le (2018), we employ a large language model (LM) for our commonsense task. Specifically, we use the LM from Jozefowicz et al. (2016), which was pre-trained on the One Billion Words Benchmark (Chelba et al., 2013). We use this model with two variations. In the first variation (LM1B-CONCAT), we simply concatenate each answer to the question. For the second variation (LM1B-REP), we first cluster questions according to their two first words. Then, we recognize five high frequency prefixes that cover 35% of the development set (e.g., the prefix *"what is"*). We rephrase questions that fit to one of these prefixes as a declarative sentence that contains the answer. For example, we rephrase *"What is usually next to a door?"* and the candidate answer *"wall"* to *"Wall is usually next to a door"*. For questions that do not start with the above prefixes, we concatenate the answer as in LM1B-CONCAT. In both variations we choose the answer with highest LM probability.

**c. QABILINEAR** This model, propsed by Yu et al. (2014) for QA, scores an answer $a_i$ with a bilinear model: $qWa_i^\top$, where the question $q$ and answers $a_i$ are the average pre-trained word embeddings and $W$ is a learned parameter matrix. A softmax layer over the candidate answers is used to train the model with cross-entropy loss.

**d. QACOMPARE** This model is similar to an NLI model from Liu et al. (2016). The model represents the interaction between the question $q$ and a candidate answer $a_i$ as: $h = \text{relu}([q; a_i; q \odot a_i; q - a_i]W_1 + b_1)$, where ';' denotes concatenation and $\odot$ is element-wise product. Then, the model predicts an answer score using a feed forward layer: $hW_2 + b_2$. As in QABILINEAR, average pre-trained embeddings and softmax are used to train the model.

**e. ESIM** Here, we use ESIM, a strong model for NLI (Chen et al., 2016). Similar to Zellers et al.

(2018), we change the output layer size to the number of candidate answers, and apply softmax to train with cross-entropy loss.

**f. Generative Pre-trained Transformer (GPT)** Radford et al. (2018) recently introduced a method for adapting pre-trained LMs to perform a wide range of tasks that require a prediction based on a list of textual inputs. We applied their model to CommonsenseQA by taking each question and its candidate answers, and encoding them into a series of delimiter-separated sequences. For example, the question *"If you needed a lamp to do your work, where would you put it?"*, and the candidate answer *"bedroom"* become "`[start]` *If...* `[sep]` *bedroom* `[end]`".

These sequences are run through a 12-layer 12-head transformer (Vaswani et al., 2017) model. The hidden representations over each `[end]` token are converted to logits by a fully-connected layer and passed through a softmax to produce final probabilities for each answer. The transformer layers were pre-trained as a LM and then fine-tuned on CommonsenseQA. We used the same pre-trained weights and hyper-parameters as Radford et al. (2018) did on ROC Stories, except with a batch size of 16.

**g. DocQA** A state-of-the-art RC model (Clark and Gardner, 2017), that uses the retrieved Google web snippets (Section 3) as context. We evaluate two variants: DocQA-fixed, which is the original model trained on TriviaQA (Joshi et al., 2017), and DocQA-trained, which we fine-tune on CommonsenseQA. To adapt the model to the multiple choice setting, we choose the answer with highest model probability.

## 6 Experiments

**Experimental Setup** We split the data into a training/development/test set with an 80/10/10 split. We perform two types of splits: (a) *random split* – where questions are split uniformly at random, and (b) *question concept split* – where each of the three sets have disjoint question concepts. We empirically find (see below) that a random split is harder for models that learn from CommonsenseQA, because the same question concept appears in the training set and development/test set with different answer concepts, and networks that memorize might fail in such a scenario. Since the random split is harder for neural models, we consider it the primary split of CommonsenseQA.

We evaluate all models from Section 5 on the test set using accuracy (proportion of examples for which the model predicts the gold answer). We tuned hyper-parameters for all trained models on the development set. To understand the quality of the baselines and the hardness of the task, we set an EASY mode for CommonsenseQA. In this mode, we replace the hard distractors that share a relation with the question concept with random ConceptNet distractors from the dataset. We expect a reasonable baseline to perform much better in this setup.

For pre-trained word embeddings we consider 300d GloVe embeddings (Pennington et al., 2014) and 300d Numberbatch ConceptNet node embeddings (Speer et al., 2017), which are kept fixed at training time. We also combine ESIM with 1024d ELMo contextual representations (Peters et al., 2018), which are also fixed during training (fine-tuning the representations reduced performance).

**Human Evaluation** To test human accuracy, we created a separate AMT task for which we did not use a qualification test, nor used AMT master workers. We sampled 100 random questions and for each question gathered answers from five AMT workers that were not involved in question generation. Taking the majority for each question we obtain 95% human accuracy and taking all answers we obtain 89% human accuracy (since workers in this task are untrained, we use a majority vote to mitigate the effect of malicious workers).

**Main Results** Table 6 presents test set results for all baselines and human accuracy, along with *EASY* mode accuracy, divided for both the random split and question concept split.

The best baseline is GPT with an accuracy of 54.8% on the random split (66.8% on question concept split). This is well below human accuracy, demonstrating that the benchmark is much easier for humans. Nevertheless, this result is much higher than random (33%), showing again the ability of language models to store large amounts of information that is correlated with commonsense knowledge.

The ESIM models, which follow GPT obtain

| Type | Formulated question example | Correct answer | distractor-1 | distractor-2 |
|---|---|---|---|---|
| Correct prediction | *Where could you find a cow that is **not real**?* | **fairy tale** | advertisement | nebraska |
| | *The newlyweds began copulating their **marriage**, they **wanted many** what?* | **babies** | rapport | odors |
| | *What happens if lovers want to show **affection**?* | **kiss each other** | break up | part ways |
| | *What would you be if you **do not have work**?* | **unemployed** | laziness | play |
| Incorrect prediction | *What could you find **in a bookstore** that is not for sale?* | carpeting | **magazines** | city |
| | *Bob was having fun with bill. How might Bob express his **feelings**?* | may laugh | **happiness** | stress relief |
| | *What might a teacher do most **during a week**?* | work in school | school children | **time test** |
| | *Where would you go if you want to see a beautiful **thunderstorm**?* | plain | **wet** | dull |
| Incorrect prediction, no clues found | *Where would you find the icebox in your home?* | kitchen | **junk yard** | antique store |
| | *What's another name for cargo?* | boat | ship's hold | **aeroplane** |
| | *Before we can become men, we are?* | boys | lady | **gods** |
| | *What could stop someone from opening business?* | busy | **get rich** | wealth |

Table 5: GPT baseline analysis. In bold is the answer predicted by the model. See body of text for details.

| | Random split | | Question concept split | |
|---|---|---|---|---|
| Model | Accuracy | EASY | Accuracy | EASY |
| VecSim+GloVe | 42.5 | 64.5 | 42.2 | 62.7 |
| VecSim+Numberbatch | 42.2 | 44.5 | 39.0 | 48.0 |
| LM1B-rep | 41.2 | 58.9 | 45.7 | 54.6 |
| LM1B-concat | 40.7 | 56.6 | 43.8 | 53.8 |
| GPT | **54.8** | 90.0 | **66.8** | 90.4 |
| ESIM+GloVe | **44.1** | 80.6 | **52.0** | 82.8 |
| ESIM+ELMo | **43.2** | 81.1 | **53.5** | 82.9 |
| QABilinear+GloVe | 41.5 | 80.2 | 45.8 | 80.8 |
| ESIM+Numberbatch | 38.4 | 80.4 | 45.0 | 78.2 |
| QABilinear+Numberbatch | 37.7 | 78.7 | 43.9 | 76.2 |
| QACompare+GloVe | 34.0 | 74.5 | 46.6 | 77.4 |
| QACompare+Numberbatch | 30.3 | 64.3 | 40.0 | 67.0 |
| DocQA-Trained | 37.1 | 63.4 | 42.6 | 58.8 |
| DocQA-Fixed | 42.0 | 54.9 | 40.6 | 56.2 |
| Human | **95.3** | | | |

Table 6: Main accuracy results for the different models on the test set.

| Model | With Filtering | Without Filtering | Δ |
|---|---|---|---|
| DocQA-Fixed | 41.6 | 47.7 | +5.1 |
| ESIM+ELMo | 47.5 | 49.9 | +2.4 |
| QABilinear+GloVe | 43.3 | 44.5 | +1.2 |
| QACompare+GloVe | 38.5 | 39.1 | +0.6 |
| GPT | 60.3 | 60.5 | +0.2 |

Table 7: Development accuracy of baselines with and without Google search pruning on the random split.

much lower performance. We note that ELMo representations did not improve performance compared to GloVe embeddings, possibly because we were unable to improve performance by back-propagating into the representations themselves (as we do in GPT). We assume a more careful training regimen might improve the performance of ESIM+ELMo.

The top part of Table 6 describes untrained models. We observe that performance is higher than random, but still quite low. The middle part describes models that were trained on CommonsenseQA, where GPT obtains best performance, as mentioned above. The bottom part shows results for RC models that use web snippets as context. We observe that models that use snippets do not perform very well, possibly hinting the snippets do not carry useful information (see also below).

Performance on the random split is 4-12 points lower than the question concept split across all trained models. As we alluded before, we hypothesize that this is because having questions in the development/test set that share a question concept with the training set, but have a different answer, creates an adversarial effect for high-capacity neural networks.

Last, all *EASY* models that were trained on CommonsenseQA achieve very high performance (90% for GPT), showing that indeed our process of selecting difficult distractors is crucial for creating a challenging task.

**Added Context Analysis** To examine whether filtering questions with Google search (Section 3) is effective, we evaluated a subset of our models on the random split without that filtering. This adds an additional 2,988 examples (an increase of 31%). We find that this filtering substantially affect the RC model – the accuracy of the RC model on filtered questions is 66.3%, while for unfiltered questions is 41.6% resulting in 47.7% final accuracy. However, for the other models, and in particular for GPT the effect is much smaller.

**Baseline analysis** We analyzed the performance of our top baseline, GPT, by manually inspecting its predictions. Table 5 presents three types of examples we identified. Correct predictions (top), incorrect predictions where we find surface clues in the question that may have misled the model (middle), and incorrect predictions where we did not find any surface clues for the model. Bold terms in the question indicate surface clues that we think helped the model reach its prediction. Bold answers are the prediction of the model.

## 7 Conclusion

In this work we present the CommonsenseQA dataset, a new QA challenge that contains 9,500 examples and aims to test commonsense knowledge. We describe a process for generating difficult questions at scale using ConceptNet and a search engine, perform a detailed analysis of the dataset, which elucidates the unique properties of our dataset, and extensively evaluate on a strong suite of baselines. We find that the best model is a pre-trained LM tuned for our task and obtains 54.8% accuracy, dozens of points lower than human accuracy. We hope that this dataset facilitates future work in incorporating commonsense knowledge into NLU systems.

## References

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending: Strong neural baselines for the roc story cloze task. In *ACL*.

C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

C. Clark and M. Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.

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

Ernest Davis. 2016. How to write science questions that are easy for people and hard for computers. *AI magazine*, 37(1):13–22.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

R Kowalski and M Sergot. 1986. A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95.

Douglas B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38:32–38.

Hector J. Levesque. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 329–334. Association for Computational Linguistics.

J. McCarthy. 1959. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*.

John McCarthy and Patrick J. Hayes. 1969. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press. Reprinted in McC90.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering.

N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *North American Association for Computational Linguistics (NAACL)*.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computing at NIPS*.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. Mcscript: A novel dataset for assessing machine comprehension using script knowledge. *CoRR*, abs/1803.05223.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proc. of *SEM*.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. 2018. Improving language understanding by generative pre-training. *Technical Report, OpenAI*.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

M. Roemmele, C. Bejan, and A. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the roc story cloze task. In *CoNLL*.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

T. Winograd. 1972. *Understanding Natural Language*. Academic Press.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *TACL*, 5:379–395.