# COMMONSENSEQA: A Question Answering Challenge Targeting Commonsense Knowledge

**Alon Talmor**[*,1,2] **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

arXiv:1811.00937v2 [cs.CL] 15 Mar 2019

## Abstract

When answering a question, people often draw upon their rich world knowledge in addition to the particular context. Recent work has focused primarily on answering questions given some relevant document or context, and required very little general background. To investigate question answering with prior knowledge, we present COMMONSENSEQA: a challenging new dataset for commonsense question answering. To capture common sense beyond associations, we extract from CONCEPTNET (Speer et al., 2017) multiple target concepts that have the same semantic relation to a single source concept. Crowd-workers are asked to author multiple-choice questions that mention the source concept and discriminate in turn between each of the target concepts. This encourages workers to create questions with complex semantics that often require prior knowledge. We create 12,247 questions through this procedure and demonstrate the difficulty of our task with a large number of strong baselines. Our best baseline is based on BERT-large (Devlin et al., 2018) and obtains 56% accuracy, well below human performance, which is 89%.

## 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.

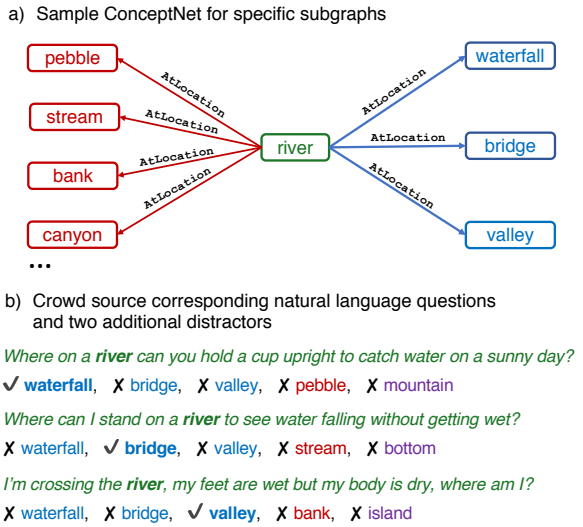


Figure 1: (a) A source concept (in green) and three target concepts (in blue) are sampled from CONCEPTNET (b) Crowd-workers generate three questions, each having one of the target concepts for its answer (✓), while the other two targets are not (✗). Then, for each question, workers choose an additional distractor from CONCEPTNET (in red), and author one themselves (in purple).

Work on Question Answering (QA) has mostly focused on answering factoid questions, where the answer can be found in a given context with little need for commonsense knowledge (Hermann et al., 2015; Rajpurkar et al., 2016; Nguyen et al., 2016; Joshi et al., 2017). 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, efforts have been invested in developing large-scale datasets for commonsense reasoning. In SWAG (Zellers et al., 2018b), given a textual description of an event, a probable subsequent event needs to be inferred. However, it has been quickly realized that models trained on large amounts of unlabeled data (Devlin et al.,

[*] The authors contributed equally

2018) capture well this type of information and performance on SWAG is already at human level. VCR (Zellers et al., 2018a) is another very recent attempt that focuses on the visual aspects of common sense. Such new attempts highlight the breadth of commonsense phenomena, and make it evident that research on common sense has only scratched the surface. Thus, there is need for datasets and models that will further our understanding of what is captured by current NLU models, and what are the main lacunae.

In this work, we present COMMONSENSEQA, a new dataset focusing on commonsense question answering, based on knowledge encoded in CONCEPTNET (Speer et al., 2017). We propose 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. Finally, for each question, the worker chooses one additional distractor from CONCEPTNET, and authors another distractor manually. Thus, in total, five candidate answers accompany each question.

Because questions are generated freely by workers, they often require background knowledge that is trivial to humans but is seldom explicitly reported on the web due to reporting bias (Gordon and Van Durme, 2013). Thus, questions in COMMONSENSEQA have a different nature compared to prior QA benchmarks, where questions are authored given an input text.

Using our method, we collected 12,247 commonsense questions. We present an analysis that illustrates the uniqueness of the gathered questions compared to prior work, and the types of commonsense skills that are required for tackling it. We extensively evaluate models on COMMONSENSEQA, experimenting with pre-trained models, fine-tuned models, and reading comprehension (RC) models that utilize web snippets extracted from Google search on top of the question itself. We find that fine-tuning BERT-LARGE (Devlin et al., 2018) on COMMONSENSEQA obtains the best performance, reaching an accuracy of 55.9%. This is substantially lower than human performance, which is 88.9%.

To summarize, our contributions are:

1. A new QA dataset centered around common sense, containing 12,247 examples.
2. A new method for generating commonsense questions at scale from CONCEPTNET.
3. An empirical evaluation of state-of-the-art NLU models on COMMONSENSEQA, showing that humans substantially outperform current models.

The dataset can be downloaded from `www.tau-nlp.org/commonsenseqa`. The code for all our baselines is available at `github.com/jonathanherzig/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. While the Winograd Schema Challenge remains a tough dataset, the difficulty of generating examples has led to only a small available collection of 150 examples. 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 modeling approaches.

With the recent adoption of crowdsourcing, several larger datasets have emerged, focusing on pre-

dicting 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 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., 2018b). 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; Ostermann et al., 2018); however, they also need specialized scientific knowledge. In contrast to these efforts, our work studies common sense without requiring additional information. SQUABU 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 the following steps.

1. We extract subgraphs from CONCEPTNET, each with one source concept and three target concepts.
2. We ask crowdsourcing workers to author three questions per subgraph (one per target concept), to add two additional distractors per question, and to verify questions' quality.
3. We add textual context to each question by querying a search engine and retrieving web snippets.

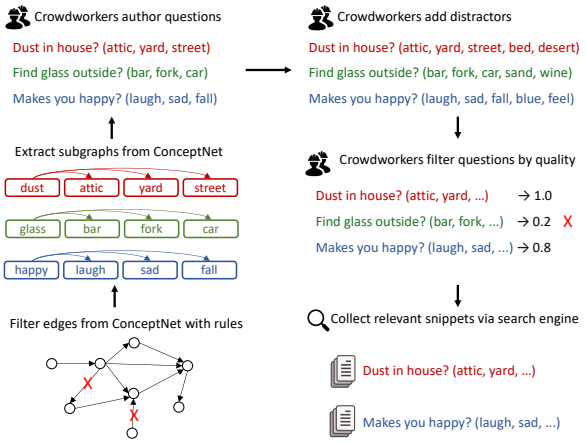


Figure 2: COMMONSENSEQA generation process. The input is CONCEPTNET knowledge base, and the output is a set of multiple-choice questions with corresponding relevant context (snippets).

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 are already well-explored in NLP (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 too 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*.

We aim to generate questions that contain the question concept and where the answer is the answer concept. To create multiple-choice questions we need to choose *distractors* for each question.

Sampling distractors at random from CONCEPTNET is a bad solution, as 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 discourage workers from providing 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'*.

Formulating questions for our task is non-trivial. Thus, we only accept annotators for which at least 75% of the questions they formulate pass the verification process described below.

**Adding additional distractors** To make the task more difficult, we ask crowd-workers to add two additional incorrect answers to each formulated question. One distractor is selected from a set of answer concepts with the same relation to the question concept in CONCEPTNET (Figure 1, in red). The second distractor is formulated manually by the workers themselves (Figure 1, in purple). Workers were encouraged to formulate a distractor that would seem plausible or related to the question but easy for humans to dismiss as incorrect. In total, each formulated question is accompanied with five candidate answers, including one correct answer and four distractors.

**Verifying questions quality** We train a disjoint group of workers to verify the generated questions.

| Measurement | Value |
|---|---|
| # CONCEPTNET distinct question nodes | 2,254 |
| # CONCEPTNET distinct answer nodes | 12,094 |
| # CONCEPTNET distinct nodes | 12,107 |
| # CONCEPTNET distinct relation lables | 22 |
| average question length (tokens) | 13.41 |
| long questions (more than 20 tokens) | 10.3% |
| average answer length (tokens) | 1.5 |
| # answers with more than 1 token | 44% |
| # of distinct words in questions | 14,754 |
| # of distinct words in answers | 4,911 |

Table 1: Key statistics for COMMONSENSEQA

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. This processes filters out 15% of the questions.

**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 candidate 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 of the five answer candidates, yielding a context of 500 snippets per question. Using this context, we can investigate the performance of reading comprehension (RC) models on COMMONSENSEQA.

Overall, we generated 12,247 final examples, from a total of 16,242 that were formulated. The total cost per question is $0.33. Table 1 describes the key statistics of COMMONSENSEQA.

## 4 Dataset Analysis

**CONCEPTNET concepts and relations** COMMONSENSEQA builds on CONCEPTNET, which contains *concepts* such as `dog`, `house`, or `row boat`, connected by *relations* such as `Causes`, `CapableOf`, or `Antonym`. The top-5 question concepts in COMMONSENSEQA are *'Person'* (3.1%), *'People'* (2.0%), *'Human'* (0.7%), *'Water'* (0.5%) and *'Cat'* (0.5%). In addition, we present the main relations along with the percentage of questions generated from them in Table 2. It's worth noting that since question formulators were not shown the CONCEPTNET relation, they often asked questions that probe other relationships between the concepts. For example, the question

| Relation | Formulated question example | % |
|---|---|---|
| AtLocation | *Where would I not want a fox?* **A.** hen house, **B.** england, **C.** mountains, **D.** ... | 47.3 |
| Causes | *What is the hopeful result of going to see a play?* **A.** being entertained, **B.** meet, **C.** sit, **D.** ... | 17.3 |
| CapableOf | *Why would a person put flowers in a room with dirty gym socks?* **A.** smell good, **B.** many colors, **C.** continue to grow , **D.** ... | 9.4 |
| 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 , **D.** ... | 8.5 |
| HasSubevent | *How does a person begin to attract another person for reproducing?* **A.** kiss, **B.** genetic mutation, **C.** have sex , **D.** ... | 3.6 |
| 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 , **D.** ... | 3.3 |
| CausesDesire | *What do parents encourage kids to do when they experience boredom?* **A.** read book, **B.** sleep, **C.** travel , **D.** ... | 2.1 |
| Desires | *What do all humans want to experience in their own home?* **A.** feel comfortable, **B.** work hard, **C.** fall in love , **D.** ... | 1.7 |
| PartOf | *What would someone wear to protect themselves from a cannon?* **A.** body armor, **B.** tank, **C.** hat , **D.** ... | 1.6 |
| HasProperty | *What is a reason to pay your television bill?* **A.** legal, **B.** obsolete, **C.** entertaining , **D.** ... | 1.2 |

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

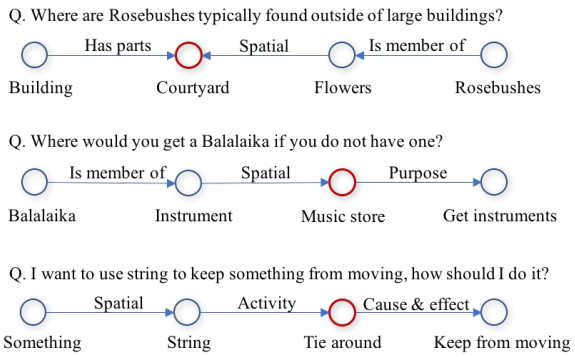


Figure 3: Examples of manually-annotated questions, with the required skills needed to arrive at the answers (red circles). Skills are labeled edges, and concepts are nodes.

| 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 not sum to 100%.

*"What do **audiences** clap for?"* was generated from the AtLocation relation, but focuses on social conventions instead.

**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.

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. We now elaborate on the different baselines.

**a VECSIM** A 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

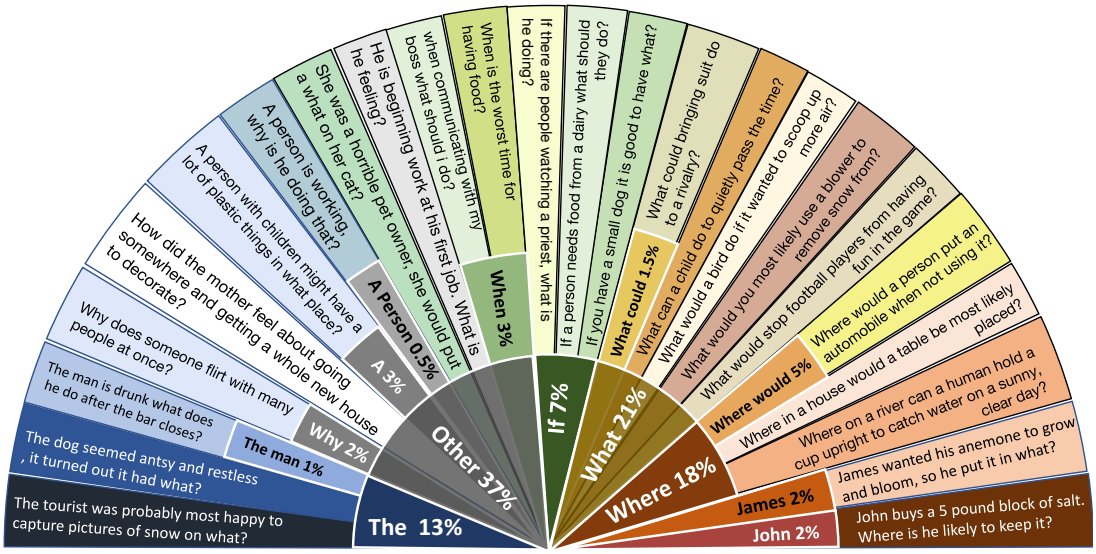


Figure 4: Distribution of the first and second words in questions. The inner part displays words and their frequency and the outer part provides example questions.

| Model | Training | Context |
|---|---|---|
| VECSIM | ✗ | ✗ |
| LM1B | ✗ | ✗ |
| QABILINEAR | ✓ | ✗ |
| QACOMPARE | ✓ | ✗ |
| ESIM | ✓ | ✗ |
| GPT | ✓ | ✗ |
| BERT | ✓ | ✗ |
| BIDAF++ | ✓ | ✓ |

Table 4: 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.

employ a large language model (LM) from Jozefowicz et al. (2016), which was pre-trained on the One Billion Words Benchmark (Chelba et al., 2013). We use this model in two variations. In the first (LM1B-CONCAT), we simply concatenate each answer to the question. In the second (LM1B-REP), we first cluster questions according to their first two words. Then, we recognize five high-frequency prefixes that cover 35% of the development set (e.g., *"what is"*). We rephrase questions that fit into one of these prefixes as a declarative sentence that contains the answer. E.g., 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 return 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: $hW_2 + b_2$. Average pre-trained embeddings and softmax are used to train the model.

**e ESIM** We use ESIM, a strong NLI model (Chen et al., 2016). Similar to Zellers et al. (2018b), we change the output layer size to the number of candidate answers, and apply softmax to train with cross-entropy loss.

**f BIDAF++** A state-of-the-art RC model, that uses the retrieved Google web snippets (Section 3) as context. We augment BIDAF (Seo et al., 2016) with a self-attention layer and ELMo representations (Peters et al., 2018; Huang et al., 2018). To adapt to the multiple-choice setting, we choose the answer with highest model probability.

**g GENERATIVE PRE-TRAINED TRANSFORMER (GPT)** Radford et al. (2018) proposed a method for adapting pre-trained LMs to perform a wide range of tasks. We applied their model to COMMONSENSEQA by encoding each question and its candidate answers as 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"* would become "`[start]` *If ... ?* `[sep]` *bedroom* `[end]`". The hidden representations over each `[end]` token are converted to logits by a linear transformation and passed through a softmax to produce final probabilities for the answers. We used the same pre-trained LM and hyper-parameters for fine-tuning as Radford et al. (2018) on ROC Stories, except with a batch size of 10.

**h BERT** Similarly to the GPT, BERT fine-tunes a language model and currently holds state-of-the-art across a broad range of tasks (Devlin et al., 2018). BERT uses a masked language modeling objective, which predicts missing words masked from unlabeled text. To apply BERT to COMMONSENSEQA, we linearize each question-answer pair into a delimiter-separated sequence (i.e., "`[CLS]` *If ... ?* `[SEP]` *bedroom* `[SEP]`") then fine-tune the pre-trained weights from uncased BERT-LARGE.[1] Similarly to the GPT, the hidden representations over each `[CLS]` token are run through a softmax layer to create the predictions. We used the same hyper-parameters as Devlin et al. (2018) for SWAG.

## 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, we consider it the primary split of COMMONSENSEQA.

We evaluate all models on the test set using accuracy (proportion of examples for which prediction is correct), and tune hyper-parameters for all trained models on the development set. To understand the difficulty of the task, we add a SANITY mode, where we replace the hard distractors (that share a relation with the question concept and one formulated by a worker) with random CONCEPTNET distractors. We expect a reasonable baseline to perform much better in this mode.

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, which are also fixed during training.

**Human Evaluation** To test human accuracy, we created a separate 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 workers that were not involved in question generation. Humans obtain 88.9% accuracy, taking a majority vote for each question.

**Results** Table 5 presents test set results for all models and setups.

The best baselines are BERT-LARGE and GPT with an accuracy of 55.9% and 45.5%, respectively, on the random split (63.6% and 55.5%, respectively, on the 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 (20%), showing the ability of language models to store large amounts of information related to commonsense knowledge.

The top part of Table 5 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 BERT-LARGE obtains best performance, as mentioned above. ESIM models follow BERT-LARGE and GPT, and obtain 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 BERT-LARGE and GPT). The bottom part shows results for BIDAF++ that uses web snippets as context. We observe that using snippets does not lead to high performance, hinting that they do not carry a lot of useful information.

Performance on the random split is five points lower than the question concept split on average

[1] The original weights and code released by Google may be found here: https://github.com/google-research/bert

| | Random split | | Question concept split | |
|---|---|---|---|---|
| Model | Accuracy | SANITY | Accuracy | SANITY |
| VECSIM+NUMBERBATCH | 29.1 | 54.0 | 30.3 | 54.9 |
| LM1B-REP | 26.1 | 39.6 | 26.0 | 39.1 |
| LM1B-CONCAT | 25.3 | 37.4 | 25.3 | 35.2 |
| VECSIM+GLOVE | 22.3 | 26.8 | 20.8 | 27.1 |
| BERT-LARGE | **55.9** | **92.3** | **63.6** | **93.2** |
| GPT | 45.5 | 87.2 | 55.5 | 88.9 |
| ESIM+ELMO | 34.1 | 76.9 | 37.9 | 77.8 |
| ESIM+GLOVE | 32.8 | 79.1 | 40.4 | 78.2 |
| QABILINEAR+GLOVE | 31.5 | 74.8 | 34.2 | 71.8 |
| ESIM+NUMBERBATCH | 30.1 | 74.6 | 31.2 | 75.1 |
| QABILINEAR+NUMBERBATCH | 28.8 | 73.3 | 32.0 | 71.6 |
| QACOMPARE+GLOVE | 25.7 | 69.2 | 34.1 | 71.3 |
| QACOMPARE+NUMBERBATCH | 20.4 | 60.6 | 25.2 | 66.8 |
| BIDAF++ | 32.0 | 71.0 | 38.4 | 72.0 |
| HUMAN | **88.9** | | | |

Table 5: Test set accuracy for all models.

| **Category** | **Formulated question example** | **Correct answer** | **Distractor** | **Accuracy** | **%** |
|---|---|---|---|---|---|
| Surface clues | *If someone laughs after surprising them they have a good sense of what?* | **humor** | laughter | 77.7 | 35% |
| | *How might a automobile get off a freeway?* | **exit ramp** | driveway | | |
| Negation / Antonym | *Where would you store a pillow case that is not in use?* | drawer | **bedroom** | 42.8 | 7% |
| | *Where might the stapler be if I cannot find it?* | desk drawer | **desktop** | | |
| Factoid knowledge | *How many hours are in a day?* | twenty four | **week** | 38.4 | 13% |
| | *What geographic area is a lizard likely to be?* | west texas | **ball stopped** | | |
| Bad granularity | *Where is a well used toy car likely to be found?* | child's room | **own home** | 35.4 | 31% |
| | *Where may you be if you're buying pork chops at a corner shop?* | iowa | **town** | | |
| Conjunction | *What can you use to store a book while traveling?* | suitcase | **library of congress** | 23.8 | 23% |
| | *On a hot day what can you do to enjoy something cool and sweet?* | eat ice cream | **fresh cake** | | |

Table 6: BERT-LARGE baseline analysis. For each category we provide two examples, the correct answer, one distractor, model accuracy and frequency in the dataset. The predicted answer is in bold.

across all trained models. 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 difficulty for networks that memorize the relation between a question concept and an answer.

Lastly, all SANITY models that were trained on COMMONSENSEQA achieve very high performance (92% for BERT-LARGE), showing that selecting difficult distractors is crucial.

**Baseline analysis** To understand the performance of BERT-LARGE, we analyzed 100 examples from the development set (Table 6). We labeled examples with categories (possibly more than one per example) and then computed the average accuracy of the model for each category.

We found that the model does well (77.7% accuracy) on examples where surface clues hint to the correct answer. Examples that involve negation or understanding antonyms have lower accuracy (42.8%), similarly to examples that require factoid knowledge (38.4%). Accuracy is particularly low in questions where the correct answer has finer granularity compared to one of the distractors (35.4%), and in cases where the correct answer needs to meet a conjunction of conditions, and the distractor meets only one of them (23.8%).

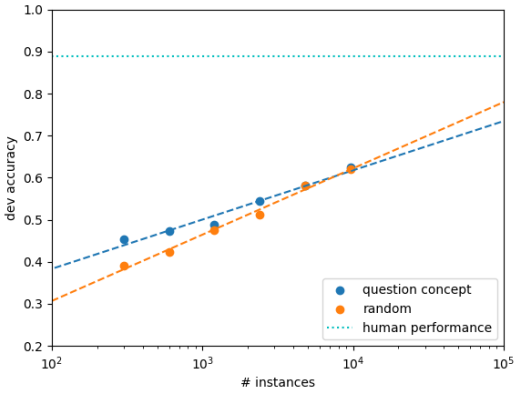


Figure 5: Development accuracy for BERT-LARGE trained with varying amounts of data.

**Learning Curves** To extrapolate how current models might perform with more data, we evaluated BERT-large on the development set, training with varying amounts of data. The resulting learning curves are plotted in figure 5. For each training set size, hyper-parameters were identical to section 5, except the number of epochs was varied to

keep the number of mini-batches during training constant. To deal with learning instabilities, each data point is the best of 3 runs. We observe that the accuracy of BERT-LARGE is expected to be roughly 75% assuming 100k examples, still substantially lower than human performance.

## 7 Conclusion

We present COMMONSENSEQA, a new QA dataset that contains 12,247 examples and aims to test commonsense knowledge. We describe a process for generating difficult questions at scale using CONCEPTNET, 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 55.9% accuracy, dozens of points lower than human accuracy. We hope that this dataset facilitates future work in incorporating commonsense knowledge into NLU systems.

## Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was completed in partial fulfillment for the PhD degree of Jonathan Herzig, which was also supported by a Google PhD fellowship. This research was partially supported by The Israel Science Foundation grant 942/16, The Blavatnik Computer Science Research Fund and The Yandex Initiative for Machine Learning.

## 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*.

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*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 25–30, New York, NY, USA. ACM.

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.

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683*.

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*.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv*.

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

O. Tange. 2011. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

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

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, Ali Farhadi, and Yejin Choi. 2018a. From recognition to cognition: Visual commonsense reasoning. *arXiv preprint arXiv:1811.10830*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018b. 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.