ProtoQA: A Question Answering Dataset for Prototypical Common-Sense Reasoning

Michael Boratko* Xiang Lorraine Li* Rajarshi Das* Tim O'Gorman*
Dan Le Andrew McCallum
College of Information and Computer Sciences
University of Massachusetts Amherst
{mboratko,xiangl,rajarshi,togorman,dhle,mccallum}@cs.umass.edu

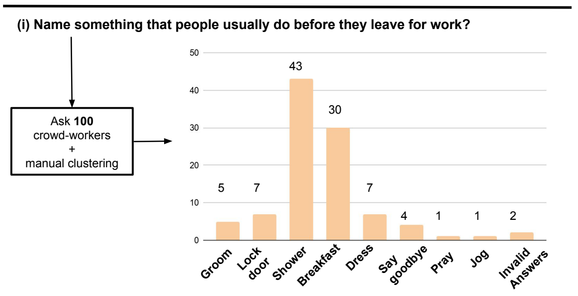
Abstract

Given questions regarding some prototypical situation — such as *Name something that people usually do before they leave the house for work?* — a human can easily answer them via acquired experiences. There can be multiple right answers for such questions with some more common for a situation than others.

This paper introduces a new question answering dataset for training and evaluating common-sense reasoning capabilities of artificial intelligence systems in such prototypical situations. The training set is gathered from an existing set of questions played in a long-running international trivia game show – FAMILY-FEUD. The hidden evaluation set is created by gathering answers for each question from *100 crowd-workers*. We also propose an open-domain task where a model has to output a ranked list of answers, ideally covering all prototypical answers for a question. On evaluating our dataset with various competitive state-of-the-art models, we find there is a significant gap between the best model and human performance on a number of evaluation metrics.

1 Introduction

Humans posses the innate ability to implicitly reason using a wealth of shared background knowledge, much of which is acquired via experiences. For example, consider the question in Figure 1 — "*Name something that people usually do before they leave the house for work?*". Humans can easily answer such questions with '*prototypical*' answers — i.e. a set of answers that they commonly associate with situations invoked by the question. These questions require common-sense reasoning, however, because of its 'common' nature, such knowledge is often not explicitly captured in text (reporting bias) (Gordon and Van Durme, 2013).



(ii) Name a piece of equipment that you are likely to find at your office and not at home?
Categories: *printer/copier (**37**), office furniture (**15**), computer equipment (**17**), stapler (**11**) files (**10**), office appliances (**5**), security systems (**1**)*

(iii) Name something that an athlete would not keep in her refrigerator.
Categories: *unhealthy / fast food (**36**), unhealthy drinks (**24**), clothing/shoes (**24**) accessories (**7**)*

(iv) Name something that you might forget in a hotel room?
Categories: *phone (**24**), toothbrush/towels (**17**), clothing/shoes (**15**) keys (**14**), purse/wallet (**14**), accessories (**8**), charger (**5**)*

Figure 1: We focus on common-sense reasoning over prototypical situations when there could be many different answers but some are more common than others. Our task is in open-domain style (*not* multiple-choice format). Answers to a question are crowd-sourced from 100 workers and are then manually clustered into categories. To perform well, a model has to output a ranked list of answers covering multiple categories.

We present a new dataset to train and evaluate models about their common-sense knowledge of prototypical situations. In such situations, there are often multiple right answers, with some answers more prototypical (common) than others, thereby forming a distribution over them. For example, when we polled *100 people* (Figure 1), popular answers to the previous question were 'shower/cleaning' (43) or 'breakfast' (30). However, we also received very reasonable answers in the tail such as 'lock door/grab keys' (7), 'say goodbye' (4) and 'pray' (1). We think, for artificial intelligence (AI) systems to achieve human-level common-sense reasoning, they should be able to match the *distribution* over prototypical answers.

[*] Equal contribution.

Data and interactive demo available at http://protoqa.com

Our dataset and task are inspired by a source of naturally occurring common sense questions (not developed for any particular NLP task) used in a competitive trivia game show — Family-Feud. Family-Feud is a long-running trivia game show which started on American television in 1976 and has been adopted internationally in more than 50 countries. The game show is played by asking participants questions such as those in Figure 1, and they receive points not for a "correct" answer, but when their answer matches the answers from other people surveyed, and in proportion to how many people gave that answer. What makes Family-Feud appealing is the fact that the original answers to the question were collected by a professional polling company by doing a telephone survey of 100 different people all over the country and further clustered into meaningful categories (e.g. cantaloupe, honeydew, watermelons as '*melons*') — thereby automatically giving us a distribution over both the answers and the underlying concepts those answers refer to.

We present a common-sense reasoning task in which a model has to output a ranked list of answers for each question. Our evaluation metrics (§ 3) are designed to encourage both popularity and diversity in answers, i.e. models that not only predict more popular answers, but also cover all plausible answer categories are encouraged. We present both a publicly available set of around 9.7K questions and along with 7-8 labeled answer categories for each question oriented towards this kind of commonsense knowledge, and a newly annotated and unseen test set for evaluation. This test set provides a set of 15,400 crowd-sourced human judgments over 154 new questions. These questions were created by us by perturbing existing Family-Feud questions to ensure that they do not occur in any iteration of the game show, while maintaining the same level of common-sense and prototypical reasoning to answer them (§ 2.4). The crowd-sourced answers were further categorized manually by two expert annotators, creating the same setup as in the original game-show.

Recent common-sense reasoning benchmarks often use a multiple-choice paradigm, where the task is to identify the most plausible answer from a list of options (Mostafazadeh et al., 2016; Zellers et al., 2018; Talmor et al., 2019). However, it has been shown that language models (LMs) trained on large amounts of unlabeled data such as Bert (Devlin et al., 2019) do exceedingly well on these datasets, achieving human-level accuracy in a few of them. To counter this, Zellers et al. (2019) propose a new adversarial filtering approach where benchmarks evolve in an adversarial way as new models are proposed and introduced the HellaSwag dataset created by using powerful LMs such as Bert. Soon after its introduction, RoBERTa (Liu et al., 2019) improved upon the accuracy of Bert model by 45 points. A primary reason for this is that generating hard negative examples for the multiple choice format is challenging, even for humans (Schwartz et al., 2017; Gururangan et al., 2018; Poliak et al., 2018).

Instead of the multiple-choice paradigm, we setup our task in an open-domain question answering (QA) format where a model has to output a ranked list of answers which is 'matched' to crowd-sourced answers in each category. While such an approach can penalize a correct model prediction when it does not match an existing reference answer, we counter this issue in (a) by collecting *100 answer annotations* per question, a number substantially higher than any other work in open-domain QA and (b) by proposing evaluation metrics which use large lexical resources such as WordNet (Miller, 1995) to perform matching, and (c) by focusing upon methods to score ranked lists of answers, instead of focusing upon a top score. We suggest that open-domain evaluation of some common-sense reasoning tasks is a natural and realistic paradigm, and one which shares natural similarities to the evaluation challenges found in various natural language generation tasks such as summarization (Radev et al., 2003) and translation (Callison-Burch et al., 2010) and should be an area of active research.

We evaluate this dataset on a variety of baseline models — from models trained on symbolic common-sense knowledge store such as ConceptNet (Speer et al., 2017), to QA models powered by large masked LMs such as Bert, to the direct prediction of answers in a language-modeling paradigm using a large Gpt-2 LM (Radford et al., 2018). While most models perform quite poorly at this challenging task, when Gpt-2 was fine-tuned using the Family-Feud training set, its performance improved drastically, although remaining significantly below a score of human-level performance.

The contributions of this paper are as follows. (a) We introduce a large scale QA dataset regarding

http://shorturl.at/guW34
http://shorturl.at/bhKS1
http://shorturl.at/rFMT2

common-sense knowledge of prototypical situations, and a rich evaluation set for models trained upon that data. (b) We present this as an open-domain task, and review a range of directions for robust evaluation in this open-domain setting, both with rich data (large sets of reference answers and clustering over answers) and with evaluation measures such as a WordNet-based similarity. (c) We also design evaluation metric that encourage models to provide diverse answers covering all plausible answer categories (d) We evaluate our dataset on existing models, and reveal the strong ability of large contextualized language models when fine-tuned on this data. Finally, (e), we discuss the gap between model and human performance on this task, showing that this is still a challenging task for models with room for improvement.

2 Dataset Creation and Analysis

2.1 Training Corpus Collection

Three publicly available fan websites for the show Family Feud were used to collect a large collection of questions.

A range of fan websites for Family Feud have transcribed such Family Feud questions. Well over 10,000 questions were collected with their answers, and a set of 9,762 questions remained after filtering, quality control, and de-duplication. That filtering included the omission of classes of questions (such as *name a vegetable*) which did not evaluate interesting commonsense knowledge.

While any "commonsense" dataset inherently bears the risks of encoding culture-specific information and biases, questions regarding prototypical scenarios and prototypical behavior are naturally quite susceptible to this, and we note it as an important issue to be aware of regarding such data. A small subset of 29 questions which might be viewed as problematic or encoding stereotypes were explicitly labeled and will be released separately with the training data, so that one might evaluate the extent to which models trained on such a task might acquire undesirable biases.

2.2 Test Corpus Collection

In order to focus upon a rich, open-ended answer generation task, we collected 100 answers for each question from the crowd-sourcing platform FigureEight, and then provided rich double-annotated clustering over those answers. By gathering large sets of possible answers and clustering them we can provide rough distributions over the expected answers, increasing the ability to recognize any way of expressing one of those answers.

We gathered a test set of new questions with an eye towards maintaining the same domain and the same commonsense reasoning seen in the training data. In order to maintain similarity to existing questions, we removed a set of questions from the scraped data and then perturbed important aspects of them. For example, given an existing question of "*Name something a person might forget to put on if they leave the house in a hurry.*", changes of polarity and events would derive a related question "*Name something that people usually do before they leave house for work*". Deriving such unseen test questions was especially important to avoid the risk of having a publicly-available question be included in the training data for contextual language models; by making new data, we can be more confident that any high-performing model has not yet seen the data.

Having derived new questions, we then created tasks on FigureEight for each of those questions to be answered by 100 workers. To match the training data (which is inherently grounded in US culture), we limited workers to US locations. Low-quality workers were automatically detected through test questions during annotation, and the clustering pass provided a second manual quality control check. This left us with 154 questions which we split into a test set and dev set of 102 and 52 respectively.

2.3 Answer Clustering

After initial collection of 100 answers for each question, we then clustered answers of each question. Each list was manually clustered by two different experts familiar with the task. The clusterings were generated separately and then compared, and a final clustering was agreed on. During this clustering phase, answers could be marked as invalid as well — most commonly, either due to low-quality annotators or a clear misunderstanding of a question. In order to keep these clusters roughly similar to the granularity of answers used in the training data and to avoid low-quality evaluation, we eliminated questions for which the 8 most popular cluster did not cover more than 85 of the 100 responses.

Since each set of answers was clustered twice and adjudicated, we measure the agreement with a cluster agreement metric BLANC (Recasens and Hovy, 2011; Luo et al., 2014), an extension of the Rand index used to score coreference clustering. Using this, the similarity between the clusters

for example, https://protoqa.com/

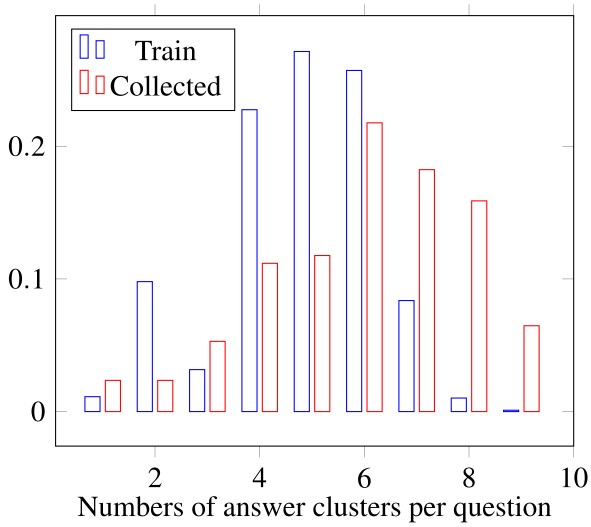


Figure 2: Number of answer clusters per question, in the scraped (blue) and crowdsourced (red) datasets.

produced by any two annotators averaged out to a BLANC score of 83.17, suggesting a coherent amount of agreement regarding the clustering of answers.

Figure 2 illustrates how this crowd-sourced test set related to the training data; the actual size of the largest clusters remains similar between the two datasets, but our data tends to have more clusters, generally capturing all possible answers within the top 8 clusters, but often using seven or eight clusters. More cluster provide more relaxed evaluation as we include more answer strings with smaller size clusters, which also provide with us more interesting answer strings.

2.4 Exploration of Dataset

The data presented here involves a range of different types of commonsense knowledge. To examine the distribution of different kinds of reasoning, and to examine whether that distribution of reasoning varied between the publicly available data and the crowdsourced development and test set, we propose a small inventory of six types of commonsense reasoning often present in these questions.

These types consist of (1) MENTAL OR SOCIAL REASONING, (2) KNOWLEDGE OF PROTOTYPICAL SITUATIONS which one is familiar with, (3) REASONING ABOUT NOVEL, COMPLEX EVENTS, (4) NEGATION AND EXCEPTIONS and understanding their consequences, (5) SPECIFIC ENTITY KNOWLEDGE of named people, locations, or organizations, and finally (6) KNOWLEDGE OF HABITUAL ACTIVITIES of specific occupations or types of entities.

To study the distribution over the data, we took 25 questions from the training collection and 25 questions from the crowd-sourced development set, and marked each one with any number of the six categories which seemed necessary for the question as a simple approximation of prior works which examine the types of knowledge required for reasoning tasks (LoBue and Yates, 2011; Boratko et al., 2018). Table 1 illustrates examples of questions with these types, and one can see the frequency of each type used in Table 2. The counts shown for each dataset illustrate that while the creation methodology varied between the two resources, the kind of commonsense reasoning tasks evaluated by these models is quite similar between the two corpus types. The greatest difference to note is that the crowd-sourced data makes less use of questions regarding specific entities, which were avoided as they tended to involve fact-based world-knowledge rather than commonsense reasoning.

3 Evaluation

Recent commonsense reasoning benchmarks often use a multiple-choice paradigm where the task is to identify the most plausible answer from a list of options (Zellers et al., 2018, 2019). However generating challenging negative examples is hard, so that often within months of the release of a dataset, models may achieve human-level or near human-level performance, as occured with BERT for the SWAG dataset and ROBERTA for the HELLASWAG dataset. Such issues highlight the difficulty of establishing robust and stable metrics using negative samples and adversarial methods alone. It also has been shown that generating negative examples is even hard for humans, who can inadvertently introduce *annotation artifacts* which models can easily identify in order to solve the task (Schwartz et al., 2017; Gururangan et al., 2018; Poliak et al., 2018).

An appealing alternative for benchmarking models is via open-domain answer generation tasks where the model has to generate the correct answers. This side-steps the need to find challenging negative examples. However, this paradigm introduces another challenge — the possibility of models getting wrongly penalized for predictions not in the list of correct answers. This problem is also faced in other natural language generation task such as machine translation (MT), summarization and dialog generation.

Our solution to the above problem is to collect and cluster a large number of open-ended responses

Question	Example Answers	Types
Name a profession where you might be fired if you lost your voice	radio host , teacher	3, 4, 6
Name something a boy scout might learn.	knot tying, camping	2, 5, 6
Name a bad sport for someone who is afraid of the water.	diving, water polo	1, 3 ,6
Name something a monk probably would not own.	weapons, smartphone	2, 4, 6
Name something parents tell their kids not to do	steal, smoke	1, 2, 4, 6
Name a reason why someone would wear gloves	cold weather, cleaning	2, 3

Table 1: Examples of questions from collected (top 3) and crowd-sourced (bottom 3) development sets

Reasoning type	Scraped Dev	Crowd-sourced
Mental/Social	16%	12%
Prototypical Events	68%	80%
Event Reasoning	28%	40%
Negation	12%	20%
Specific Entities	20%	4%
Habitual Activity	40%	24%

Table 2: Percentage of questions utilizing each reasoning type

(100 crowd-sourced responses in our case). This is much higher than other typical tasks (e.g. there are usually very few reference translation or summaries). Also, compared to summarization or MT, a prototypical answer for our task is only a word or a short phrase, making the problem less severe. Furthermore, as described shortly, we do not restrict ourselves to rigid exact matches and propose a similarity measure that uses synonyms from WordNet (Miller, 1995).

The next subsections describe how we score a ranked list of model predictions w.r.t the answer clusters containing the 100 crowd-sourced responses. We first describe a similarity function to compute similarity between two strings (§ 3.1) and then we describe how we score the ranked list encouraging diversity in the answers (§ 3.2).

3.1 WordNet based similarity

With the large number of raw answers retrieved for each question, exact string matching of a new answer to those in each answer cluster works surprisingly well. Still, it is clear that reasonable answer strings (eg. synonyms or slightly embellished phrases) may be incorrectly marked as wrong with such a stringent matching criteria. METEOR (Banerjee and Lavie, 2005; Lavie and Denkowski, 2009) addressed similar issues in machine translation via stemming and synonym matching. We take a similar approach, expanding the set of answer clusters using WordNet synsets, and comparing all possible partitions of the tokenization of the raw strings. For more details, please see the appendix.

3.2 Encouraging Diversity of Answers

As mentioned before, we want to design evaluation metrics that favor models that can cover all plausible answer categories and not just predict the most popular answer. We first compute an alignment score between each answer in the ranked list and each of our answer clusters. The alignment score is computed as the maximum score between the predicted answer string and any reference string present in the cluster, scaled by the size of the cluster. After computing the alignment scores between all pairs of answers and clusters, we employ Hungarian matching algorithm (Kuhn, 1955; Munkres, 1957) to compute the exact optimal matching of answers to clusters. It is worth noting that, a model which produces a ranked list of answers only in one cluster will be penalized and a model which maximally covers all plausible clusters will score the maximum. Lastly, to make the comparison between lists of different lengths uniform, we propose the following metrics.

1. Max Answers@k limits the total number of answers allowed to upto k answers.

2. Max Incorrect@k allows unlimited answers, but stops after k unmatched answers are provided.

4 Baselines

We explore three baseline models for this task – a knowledge-base baseline using ConceptNet (Speer

We also report a weaker metric that eschews the size of the cluster (Set Intersection) in table 3

Note that since our scores are always calculated as a percentage of the max score one could receive, Max Answers is slightly different than hits@k in this setting.

et al., 2017), a QA-based model which retrieves related posts in a discussion forum for each question, and a language-modeling baseline which examines how well modern pre-trained language models do at directly producing the answers.

4.1 Knowledge Base Baseline

ConceptNet (Speer et al., 2017) is a knowledge base containing common sense related triples which has been shown to be helpful for various downstream tasks. (Zhong et al., 2019; Wang et al., 2019) This makes it a good potential source for solving this task, as well as assessing how well this dataset captures existing notions of common sense. For example, with a question 'Besides music, name something you might hear on a morning radio show' and answer 'weather', the following ConceptNet triples (listen to radio, Cause, you hear local weather report) and (listen to radio, HasSubevent, hear weather report) provide valid support for the answer of weather report. We use this idea to conduct the ConceptNet baseline.

The ConceptNet baseline is a purely symbolic baseline which can be evaluated without the need for training. Given a question, we extract a list of key words from the question by removing stop words. We perform the same key word extraction procedure for the term1 and term2 of ConceptNet triples. Provided with the list of key words, we compare them against term1 in ConceptNet triples. If there are any overlap words between sentence key words and term1 key words, we return elements term2 as possible answers to the question.

The answers are further ranked by the corresponding ConceptNet triple score. Note that this is a fairly generous baseline since the model is able to return an unlimited number of answers, however the resulting answers will be noisy as well which prevents it from performing well on the Max Incorrect ranking task. The set intersection score will give some indication on the overlap between common sense captured in ConceptNet, and provides an idea of how much gain there could be had by a more sophisticated ranking of the outputs.

4.2 Question-Answering Baseline

As this dataset is in the form of questions and answers, it may be treated as a QA dataset, although the content is far from the fact-based data usually modeled in QA tasks. As the training set only shows answers out of context, one must use distant supervision in order to train a QA model on the data, a well-explored situation in modern QA work (Joshi et al., 2017).

We should note that unlike factoid-based QA, one may expect there to be a limit in the performance of such a models, as commonsense data is well-known to have a *reporting bias* (Gordon and Van Durme, 2013) wherein many parts of general knowledge are never explicitly stated in text. Because of that, models trained in this paradigm cannot be expected to find explicit statements of the generalizations (which would often be left unstated) but can only hope to learn how to identify situations where a particular fact is presupposed or entailed.

To train a model in this approach, we collected a set of 85,781 documents by using a web search for each question. All searches were constrained to Reddit, which contains a large amount of advice and personal narratives of a domain useful for the task. For any post matching that query, any strings matching an answer to that question in the training data would be treated as a positive example for the QA model. Table 4 illustrates the kind of examples found for a single query "name something you do at a concert", which illustrates that while many examples are roughly correct in that they address those activities in a concert environment, learning a QA model from them is more difficult.

For the baseline results reported here, we finetune the "Bert for QA" model of the transformers package (Wolf et al., 2019) designed for Squad 2.0 (Rajpurkar et al., 2018) , fine-turning BERT-large (Devlin et al., 2019). At test time, the model was applied to all passages of all Reddit threads found in the first page of the search query for the question, and the 20-best scores from each passage were combined together, reporting a ranked list using the summed scores.

4.3 Language Model Baseline

We also report a language model generation baseline due to the improved representation power by language models. The baseline is performed using AI2 GPT2 large model (Radford et al., 2019) of hugging face pytorch implementation (Wolf et al., 2019). We get the answer predictions through the language model by either doing fine-tuning by our training data or without fine tuning.

We transform the original question by hand-designed transformed rules in order for it to be compatible with the GPT2 training data. E.g "Name something people do when they wake up." → "One thing people do when they wake up is ...". The

Metrics %			ConceptNet	QA Model	GPT2	GPT2 Fine Tune	Human Ranking
Exact Match	Set Intersection		39.1	23.3	22.0	39.4	**46.5**
	Max Answers	1	2.8	2.1	7.5	34.6	**78.4**
		3	2.8	9.9	23.7	46.9	**74.4**
		5	2.6	12.6	25.8	51.6	**72.5**
		10	6.9	20.0	31.2	58.2	**73.3**
	Max Incorrect	1	1.7	00.8	5.9	26.5	**55.8**
		3	2.3	9.7	21.9	45.5	**69.4**
		5	2.4	12.4	26.1	52.8	**72.4**
WordNet Similarity	Set Intersection		-	28.9	32.8	46.1	**53.9**
	Max Answers	1	-	3.8	20.8	44.1	**78.4**
		3	-	13.1	36.0	56.2	**76.8**
		5	-	16.3	39.2	59.6	**76.0**
		10	-	26.7	44.4	65.8	**77.0**
	Max Incorrect	1	-	2.1	19.7	35.9	**59.0**
		3	-	12.8	34.4	54.4	**74.0**
		5	-	15.9	39.6	60.1	**77.9**

Table 3: Results on the annotated test set.

Q: Name something you do at a concert:
A: *But you are always expected to* ***clap*** *for the spalla .*
A: *I'll often buy a* ***drink*** *for something to do, or check my email on my phone, or whatever, to kill time . once the band starts i 'm focused on that*

Table 4: Examples of distant-learning positive examples used for training QA baseline

hand-designed rules are attached in appendix of the paper. The transformed questions can be used as the input to the language model, GPT2 is expected to finish the sentence, and we will take the generated tokens as our predicted answer. The reported fine-tuning result is trained on the scraped corpus and the best model selected based on performance on our annotated dev data.

In order to generate diverse answers for a given sentence we use Nucleus Sampling (Holtzman et al., 2019) as our decoding methods. We get 300 sampled answers for each question then group them by counts. The returned ranked list is ranked by each answer's occurrence in the 300 samples.

4.4 Human Performance

As can be seen from table 3, fine-tuned GPT2 model which independently samples answers performs the best. To benchmark human performance against such models, we collected 30 human responses per question and aggregated them by counts (like GPT2 predictions). Last column in table 3 reports the human performance.

Discussion It is clear from table 3 that both the KB and QA based model significantly underperform on our dataset. The low performance of the KB baseline hints towards the low coverage of ConceptNet required for answering the prototypical scenarios of our questions. Similarly powerful QA models and large LMs that are trained on large corpus also seem to not have the common sense knowledge required to answer the question. Surprisingly, the performance of GPT2 model that was further fine-tuned on our training data significantly improved, suggesting the usefulness of our accompanying training set. However, the human performance for all our metrics significantly outperform all baselines suggesting large scope of improvement.

5 Related Work

A wide variety of commonsense reasoning datasets now exist, although none address the same kind of commonsense generalizations evaluated here. Datasets exist evaluating plausible contexts, reasons or results for physical commonsense, social reasoning, visual question answering and abductive reasoning (Bisk et al., 2019; Bhagavatula et al., 2019; Huang et al., 2019; Sap et al., 2019b; Zellers et al., 2018, 2019), but differ in evaluating against negative samples during evaluation. The ATOMIC dataset (Sap et al., 2019a) is a more similar commonsense reasoning dataset that proposed here; while ATOMIC utilized if-then reasoning (such as the resultant states and motivations of participants), it also assumes an open-domain task of freely predicting strings, although they evaluate using human assessment. Related veins of work study commonsense reasoning and inference or entailment (Zhang et al., 2017; Bowman et al., 2015; Roemmele et al., 2011; Levesque et al., 2012).

This particular dataset might be said to study generalizations, or prototypical events and situations. This naturally has connections both to the modeling of scripts and frames (Schank and Abelson, 1977; Chambers and Jurafsky, 2009; Fillmore et al., 1976; Ferraro and Van Durme, 2016), but we assume no need to predict latent structures.

6 Conclusion

We have presented a new common sense dataset with many novel features. The inclusion of counts over clusters of answers provides a very rich structure to train and evaluate with. The collection of a large set of answers and a proposed automated method of assigning answers to clusters facilitates an open-ended style of evaluation, which is often the desired use-case for these models. As shown in table 3, existing fine-tuned state-of-the-art models have a way to go before modeling the distribution of this common sense data.

7 Future Work

In addition to the elements of this task which are appealing from a common sense modeling perspective, the inherent appeal of this task to humans opens a number of possibilities for future data collection and evaluation. While the high availability of crowdsource workers has led to great progress in dataset generation it is not without its flaws, and weeding out poor quality responses is often nontrivial for more interesting tasks. On the other hand, millions of people play this game as an app on their phones, not for any monetary gain but simply for their own enjoyment. In the future we propose to collect more data by creating a form of this game, leveraging people's natural interest and enjoyment and using mechanism design to encourage high quality answers to more common sense questions.

8 Acknowledgments

We thank the IESL and NLP labs at UMass Amherst for their generous efforts in assisting with data collection. This work was supported in part by the Center for Intelligent Information Retrieval and the Center for Data Science, in part by the Chan Zuckerberg Initiative under the project Scientific Knowledge Base Construction, and in part by DARPA. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

http://shorturl.at/lnoEZ

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*.

Michael Boratko, Harshit Padigela, Divyendra Mikkilineni, Pritish Yuvraj, Rajarshi Das, Andrew McCallum, Maria Chang, Achille Fokoue-Nkoutche, Pavan Kapanipathi, Nicholas Mattei, et al. 2018. A systematic classification of knowledge, reasoning, and context within the arc dataset. *arXiv preprint arXiv:1806.00358*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Francis Ferraro and Benjamin Van Durme. 2016. A unified bayesian model of scripts, frames and language. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Charles J Fillmore et al. 1976. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC*.

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

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*.

Alon Lavie and Michael J Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *International Conference on the Principles of Knowledge Representation and Reasoning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for recognizing textual entailment. In *ACL*.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An extension of blanc to system mentions. In *ACL*.

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

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. In *NAACL*.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *SIAM*.

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

Dragomir R Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Celebi, Danyu Liu, and Elliott Drabek. 2003. Evaluation challenges in large-scale document summarization. In *ACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Tech report,OpenAI*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019b. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals, and understanding hillsdale. *U.: Laurence Erlbaum*.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Li 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*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL*.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. 2019. Improving natural language inference using external knowledge in the science questions domain. In *AAAI*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL*.

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

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Improving question answering by commonsense-based pre-training. In *CCF International Conference on Natural Language Processing and Chinese Computing*.

A WordNet Similarity Function

1. Let S be the set of synsets in WordNet, and let $S(x)$ be the set of synsets associated with the string x.

2. Let $\mathrm{SynsetSim}(X,Y) : S \times S \to [0,1]$ be a score for synset similarity, eg.
$$\mathrm{SynsetSim}(X,Y) := \begin{cases} 1 & \text{if } X = Y, \\ 0 & \text{otherwise.} \end{cases}$$

3. A given string may corresponse to multiple synsets. Given two strings x and y we define
$$\mathrm{SynsetsScore}(x,y) = \max\{\mathrm{SynsetSim}(S_x,S_y) : S_x \in S(x), S_y \in S(y)\}.$$

4. Some valid answer strings may not correspond to a synset at all, so we define
$$\mathrm{SubstringScore} = \max(\mathrm{SynsetsScore}(x,y), \mathrm{ExactMatch}(x,y))$$

5. Some answers are several words long, and therefore won't map to a synset even if some substring would. To account for this, we tokenize and strip stopwords from both the predicted and ground-truth answer strings. To compare these sets of tokens A, B we let $M(A,B)$ be the set of all possible (partial) matchings between elements in A and B, and then define
$$\mathrm{TokensScore}(A,B) = \max_{m \in M(A,B)} \frac{\sum_{(a,b)\in m} \mathrm{SubstringScore}(a,b)}{\max(|A|,|B|)}$$

6. We repeat this process for every element in an answer cluster C, which is a set of strings obtained from the survey, and then set the overall score for this answer cluster to be
$$\mathrm{WordNetScore}(x,C) = \max\{\mathrm{TokensScore}(T(x),T(y)) : y \in C\}$$

Remark. *Fully tokenizing the input has the potential to lose information, since some WordNet clusters are labeled with multiple words. Consider comparing "chewing gum" with "gum". The above process would assign this a score of* 0.5*, because tokenizing yields ["chewing", "gum"], however* *"chewing gum" is, itself, in the same WordNet synset as "gum". The solution to this problem in general is to compare all possible* partitions *of the tokens, and define the overall* PartitionsScore *to be the maximum among all pairs of possible partitions for the predicted answer and the ground-truth string. We replace the* TokensScore *with this* PartitionsScore *to capture such situations.*

With a scoring method as described, it is possible for an answer to receive a positive score for multiple clusters. We take the following approach:

1. Round the scores to $\{0, 1\}$ to make a "hard" cluster decision.

2. For a given question, if some predicted answers match with multiple clusters, we choose the maximum matching with respect to the final score.

B GPT2 Transformation rules

Original Sentence	Transformed Sentence
Name something ...	One thing ... is
Tell me something ...	One thing ... is
Name a/an ...	One ... is
How can you tell ...	One way to tell ... is
Give me a/an ...	One ... is

Table 5: Transformation rules from original question sentence to GPT2 format sentence