

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

Michael Boratko* Xiang Lorraine Li* Tim O’Gorman* Rajarshi Das*
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 game show – FAMILY-FEUD. The hidden evaluation set is created by gathering answers for each question from 100 crowd-workers. We also propose a generative evaluation task where a model has to output a ranked list of answers, ideally covering all prototypical answers for a question. After presenting multiple competitive baseline models, we find that human performance still exceeds model scores on all evaluation metrics with a meaningful gap, supporting the challenging nature of the task.

1 Introduction

Humans possess the ability to implicitly reason using a wealth of common background knowledge, much of which is acquired through shared experiences. For example, consider the question in Figure 1 — “*Name something that people usually do before they leave the house for work.*”. Humans can agree about the details and characteristics of a prototypical event or situation (Schank and Abelson, 1975, 1977) due to commonalities in their shared lived experiences, cultural norms and expectations. This rough agreement extends beyond an agreement on a single top response, but can be viewed

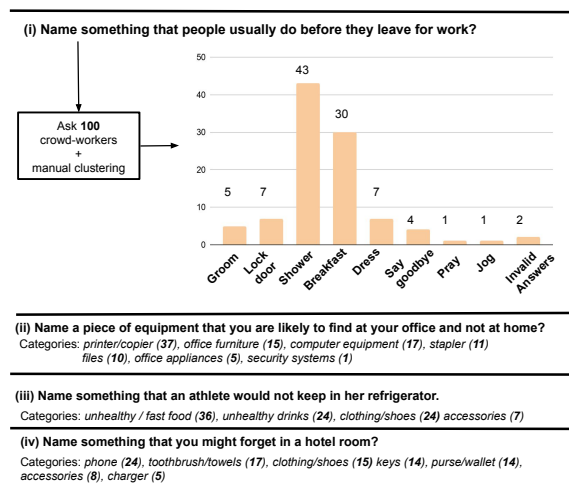


Figure 1: We focus on common-sense reasoning over prototypical situations when there could be many different answers but some are more common than others. Our task is in generative style (*not* multiple-choice format). Answers to a question are crowd-sourced from 100 workers and are then manually clustered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

as a ranked list of plausible answers, as demonstrated in Figure 1. Such sets of diverse answers represent the nature of common sense knowledge and may be useful in applications such as dialogue systems, where multiple responses are appropriate for a given context (Zhang et al., 2019b).

We present a new question/answer dataset capturing both the plausibility of the answers and the ranking preference of each answer about such prototypical situations inspired by the long-running American game show FAMILY-FEUD, which also provides the training data for the task.¹ The game show is played by prompting participants with queries such as *Name something that people usually do before they leave the house for work* (as shown in

¹Dataset: <https://github.com/iesl/protoqa-data>.
Interactive demo: <http://protoqa.com>.

* Equal contribution.

Figure 1). The answers to such questions are provided by 100 randomly selected individuals and clustered into general categories by a professional polling company. Contestants attempt to provide an answer which matches these categories and get points according to the proportion of surveyed responses within a matched category. For example, when we polled 100 people with the same question (Figure 1), they provided 43 answers involving showering/cleaning, 30 answers mentioning breakfast, and the remainder fell into smaller groups such as locking a door/grabbing keys, saying goodbye, and praying. In a FAMILY-FEUD game, if two participants on a team answered “grab a shower” and “eggs and coffee”, they would receive 73 points for providing answers which matched these two large categories. We suggest that this is an appealing paradigm for such question answering tasks where a wide range of acceptable answers exist, as it encourages both highly popular answers as well as wide coverage over the range of good answers.

We frame this task as a generative evaluation task in which a model outputs a ranked list of answers to a given question. Each answer string is then matched to one or more clusters of reference answers for that question. Matching an answer cluster gives the model a score equal to the cluster size. Our evaluation metrics (§ 3) reward models which provide the most common answers, while also measuring the model’s ability to provide a diverse set of answers in order to match all the answer clusters. While such an approach can penalize a correct model prediction when it does not match an existing reference answer, we counter this issue by (a) gathering and clustering a large number of reference answers, and (b) utilizing methods of matching non-exact matches, such as WordNet (Miller, 1995) and contextual language models such as RoBERTa (Liu et al., 2019). Generative evaluation approaches are also used in other NLP tasks such as summarization (Radev et al., 2003) and translation (Callison-Burch et al., 2010).

We evaluate on a set of competitive baseline models — from 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), as well as GPT-2 fine-tuned upon the training data. While most models perform quite poorly at this challenging task, when GPT-2 was fine-tuned using the FAMILY-FEUD training set its performance did improve drastically, although remaining significantly

below the score of human-level performance.

The contributions of this paper are as follows.

1. We introduce a large-scale QA dataset of 9.7k questions regarding common sense knowledge of prototypical situations with 7-8 labeled answer categories per question, and a corresponding evaluation set of 15,400 crowd-sourced human judgments over 154 unseen questions.
2. We present methods for robust evaluation of this task to encourage models to provide diverse answers covering all plausible answer categories.
3. We evaluate against a range of plausible baselines, showing that while large contextualized language models fine-tuned on this data can perform well at the task, a meaningful gap still exists between model and human performance, suggesting room for improvement.

2 Dataset Creation and Analysis

2.1 Training Corpus Collection

A number of fan websites exist which have transcribed FAMILY-FEUD questions and answer clusters. We use publicly available text from two such websites to provide a training dataset on this task.² Well over 10,000 questions (with answer clusters) were collected, and a set of 9,762 questions remained after filtering, quality control, and deduplication.

That filtering included the omission of questions that were taxonomic in character rather than probing common sense knowledge, such as *name a vegetable*, as well as the omission of questions encoding stereotypes. A small set of training instances which ascribe specific stereotypes or expectations to a particular group or gender — such as *“name something little boys love to build models of”* — were separated from the main training data set to avoid encouraging trained models to learn such biases³. We note, however, that common sense questions may carry a wide range of more nuanced culturally-specific information and biases. Studying the bias in such datasets, and natural stereotypical biases which pre-trained language models have been shown to have (Sheng et al., 2019), would be a valuable topic of future work.

²Scraping details and site names are provided in the datasheet (following Gebru et al. (2018)) provided with the data

³Criteria for exclusion are listed in the appendix

2.2 Test Corpus Collection

In order to establish a rich, open-ended answer generation task, we created new questions similar to those seen in the training set, collected 100 answers for each question⁴ from the crowd-sourcing platform FigureEight⁵ and manually clustered them. Because we gathered large sets of possible answers and clustered them, the evaluation set represents rough distributions over the expected raw string answers for each question, thereby increasing the ability to recognize any way of expressing one of those answers.

We attempted to make sure that this set of new questions maintained the same domain and the same common sense reasoning seen in the training data. In order to maintain similarity to existing questions, these questions were created by removing a set of questions from the scraped data and perturbing important aspects, making sure that the perturbations were sufficient to meaningfully change the answer set (thus being similar to the “counterfactually augmented” permutations of Kaushik et al. (2019)). 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 the 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 confident that any high-performing model has not yet seen the data. In order to control the quality of perturbed questions, the quality of each each perturbed question was scored by four experts (criteria listed in the appendix), and only the top-scoring questions were used to build the evaluation set.

We then created tasks on FigureEight for each selected question 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 development set of 102 and 52 respectively.

⁴Each worker, on average, provides 41 judgments, and 5 cents per judgment.

⁵Now <https://appen.com/>.

2.3 Answer Clustering

Each list of 100 raw string answers was manually clustered by two different experts familiar with the task. Clusters were assigned separately and then compared, and a final clustering was agreed on.⁶ During this clustering phase answers could be marked as invalid as well — most commonly, either due to low-quality annotations 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 clusters did not contain at least 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 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.

2.4 Analysis of the Dataset

The data presented here involves a range of different types of common sense knowledge. To explore the distribution of different kinds of reasoning, and to test whether that distribution of reasoning varied between the publicly available data and the crowd-sourced development and test set, we propose a small inventory of six types of common sense reasoning.

We are not aware of an agreed-upon typology of all commonsense reasoning types. Categorizations of different types of commonsense reasoning exist (LoBue and Yates, 2011; Boratko et al., 2018), but since each provided categorizations needed for specific tasks (RTE and the ARC dataset, respectively), neither fully covered the range of commonsense types seen in the current work. After consulting both those prior works and a separate part of the training data, we characterize the data into the following six types.

These types consist of (1) MENTAL OR SOCIAL REASONING, (2) KNOWLEDGE OF PROTOTYPICAL SITUATIONS which one is familiar with, (3) REASONING ABOUT NOVEL, COMPLEX

⁶The four total expert annotators annotated a random set of 10 questions together to calibrate their clustering granularity. Furthermore, two annotator’s results are aggregated by a third person to reduce bias.

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, characterized with reasoning types described in § 2.4

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.

Following other characterizations of reasoning type (LoBue and Yates, 2011; Boratko et al., 2018), we annotated a random sample of questions (25 from dev and 25 from train) using six basic common sense reasoning categories in order to provide a simple approximation of the distribution over reasoning types contained in the data. 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 common sense 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 common sense reasoning.

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

3 Evaluation

We present a number of methods for evaluating system-generated answers against these sets of clus-

tered answers. In each, models are evaluated by providing a ranked list of answers in response to a question. These answers are then compared to the set of reference answers for that question and scored based upon how similar they are to the known answers. While one might instead convert question-answer pairs into a multiple-choice paradigm by generating negatives, it is difficult to generate good negative examples, and the quality of a dataset can be compromised if such examples are either too easy or easily identified using biases in the negative example generation process (Mostafazadeh et al., 2016; Zellers et al., 2018; Talmor et al., 2019; Schwartz et al., 2017; Gururangan et al., 2018; Poliak et al., 2018).

We outline here our proposed method for scoring these ranked lists of predicted answers. The dataset ground truth is a ranked list of clusters of answers, including weights(cluster sizes) associated with each cluster. A first component in such an evaluation is to match each answer to an existing cluster of answers, if any cluster is acceptable. We try both simple methods such as exact match as well as more flexible ways of matching to clusters, such as using synonyms from WordNet (Miller, 1995) or a vector-based similarity method using RoBERTa (Liu et al., 2019). The second component in this generative evaluation is to provide an overall score for the entire ranked list of answers by mapping individual answers to answer clusters or marking them wrong. Scoring answers against clusters alone does not take into account the ranking. To that end, we propose two different metrics, one similar to hits@k in traditional information retrieval task and one which limits the number of incorrect answers, which is closer to how humans are typically evaluated on this task.

In each case the score reported is calculated as a percentage of the oracle score. Both proposed methods of scoring reward models which provide a

Name something that people usually do before they leave for work.

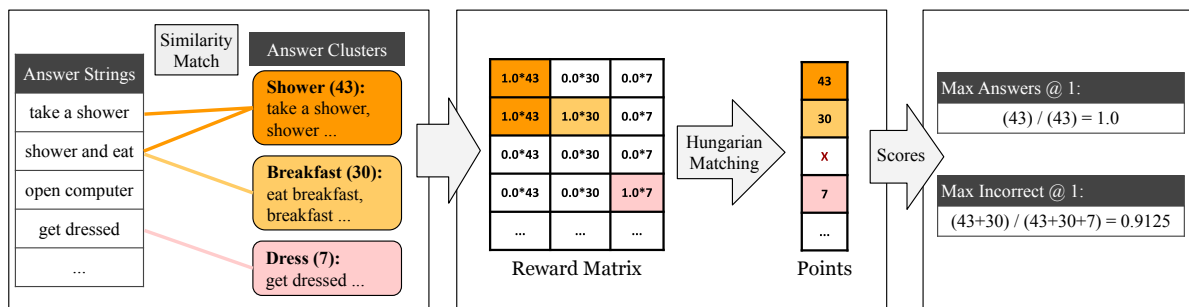


Figure 2: Example steps for evaluating a ranked list of answers

diverse set of guesses to a given query and penalize models which provide many variations of the same answer. (See figure 2 for a general idea of the steps involved.)

3.1 Matching Answers to Clusters

3.1.1 Exact Match

In our simplest way of matching answers to clusters, we compare each answer with the answer strings from crowd-source workers for a given cluster, returning a score of 1 if it matched any string in the cluster and returning 0 if not. By construction, therefore, a given answer string will match at most a single cluster with this method.

3.1.2 WordNet Similarity

Reasonable answer strings may be incorrectly marked as wrong with an exact string match, even when they are clear synonyms of a reference answer. 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, tokenizing a proposed answer string and comparing it to the tokenization of the answers in each answer cluster. Since some words in WordNet are multi-word phrases (eg. “chewing gum”) we furthermore perform this matching on all possible partitions of the tokenization. For each answer in an answer cluster we return the maximum (over all possible partitions) of the average number of matched tokens. The assignment of answers to clusters proceeds as in the exact match case. Further details are included in the appendix.

3.1.3 RoBERTa Similarity

Recent works in MT evaluation (Zhang et al., 2019a; Sellam et al., 2020) used pre-trained language models to compare predictions to reference

answers. We implement a simple version of such vector-based comparisons, but this current task differs in that we assign each predicted answer to a particular cluster of correct answers, or decide whether to reject the answer. As clusters vary in size and specificity we cannot determine a universal threshold for how similar a mention must be to a cluster. Instead, we train a small classifier in L2 distance space for each answer cluster in order to decide membership in that answer cluster. We do this by obtaining a vector representation of each answer from RoBERTa (Liu et al., 2019), concatenating each answer with the question, and taking the mean of answer token representations. For each cluster we train a small one-vs-all classifier over the 100 answers to that question, predicting membership in that cluster (using gaussian process regression (Williams and Rasmussen, 1996) with an RBF kernel). At test time, a given answer is assigned to the highest-scoring cluster, as long as its likelihood of membership exceeds a minimum probability threshold, set at 0.1. Such an approach allows us to match answers to clusters while omitting answers which do not match existing clusters.

3.2 Evaluating Diverse Lists of Answers

As mentioned previously, we want to design evaluation metrics that favor models which take into account the ranking while still covering all plausible answer categories. We first compute an alignment score between each answer in the ranked list and each of our answer clusters. After computing the alignment scores between all pairs of answers and clusters we create a reward matrix where, for each answer and cluster, we assign a reward equal to the cluster size if the alignment score was a 1 and 0 otherwise. We employ the Hungarian matching algorithm (Kuhn, 1955; Munkres, 1957) to compute the exact optimal matching of answers to clusters based

on this reward matrix, so that an answer is assigned to only one cluster. It is worth noting that a model which produces a ranked list of answers only in one cluster will do worse than a model which maximally covers all plausible clusters. 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 up to k answers.⁷
2. `MAX INCORRECT@k` allows unlimited answers, but stops after k unmatched answers are provided.

In both conditions, we report the score as the percentage of the max score one could receive given that number of guesses, and only give credit for a given cluster once.

4 Baselines

We explore three baseline models for this task: a QA-based model which retrieves related posts in a discussion forum for each question, a language-modeling baseline which examines how well modern pre-trained language models do at directly producing the answers, and a fine-tuned version of the language-model baseline.

4.1 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). Unlike factoid-based QA, one may expect a limit in the performance of such QA models for common sense reasoning, as common sense data is well-known to have a *reporting bias* (Gordon and Van Durme, 2013) wherein many facts that are part of the common ground of known knowledge are less likely to be stated.

To train a model in this approach, we collected up to 20 documents for each of the 9.7k questions in the FAMILY-FEUD training dataset by using a web search for each question constrained to Reddit. This resulted in a set of 85,781 Reddit posts total. Searches were constrained to Reddit in order to focus upon advice and personal narratives which

⁷Note 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.

might discuss common sense questions. 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. The QA model used was the “Bert for QA” implementation within the Hugging Face Transformers package (Wolf et al., 2019); training details, and examples of the kind of noisy training data generated through this process, are provided in the appendix.

At test time documents were obtained by searching for the question in a google search restricted to Reddit, and the QA model was run on that set, taking the 20 best answers in context as possible answer strings. Those best answer strings from each passage were combined together, summing scores for identical strings, to provide a ranked list.

4.2 Language Model Baseline

We also report a language model generation baseline, due to the improved representation power of modern language models and recent evidence of their power in modeling common sense reasoning tasks (Weir et al., 2020; Tamborrino et al., 2020). The baseline is performed using the AI2 GPT-2 large model (Radford et al., 2019) (specifically, the Hugging Face PyTorch implementation (Wolf et al., 2019)). We perform both a zero-shot evaluation and an evaluation after fine-tuning with using our training data.

Because the original FAMILY-FEUD prompts are not structured as completion tasks, we transform the original question by hand-designed transformation rules in order for it to be compatible with the GPT-2 training data. E.g “Name something people do when they wake up.” → “One thing people do when they wake up is ...”. The hand-designed rules are including in the appendix. The transformed questions are used as input to the language model, and GPT-2 finishes the sentence. The reported fine-tuning result is trained on the scraped training corpus and the best model selected based on performance on our annotated development set. Training details and parameter setting for the model is provided in the appendix.

In order to generate diverse answers for a given sentence we use Nucleus Sampling (Holtzman et al., 2019) as our decoding method. We get 300 sampled answers for each question and group them by counts, returning a ranked list of 20 answers from most to least common.

Metrics %		QA Model	GPT-2	GPT-2 Fine Tune	Human	
Exact Match	Max Answers	1	2.1	5.6	29.4	78.4
		3	4.4	15.9	37.6	74.4
		5	6.8	18.3	40.1	72.5
		10	11.0	23.2	45.9	73.3
	Max Incorrect	1	0.8	3.3	18.7	55.8
		3	3.6	15.1	35.0	69.4
5		6.4	19.3	41.1	72.4	
WordNet Similarity	Max Answers	1	3.4	6.2	36.4	78.4
		3	6.4	18.5	44.4	76.8
		5	9.1	23.0	46.6	76.0
		10	15.7	30.5	53.5	77.0
	Max Incorrect	1	1.4	4.3	26.1	59.0
		3	5.3	17.9	41.7	74.0
5		8.4	24.2	48.2	77.9	
RoBERTa Similarity	Max Answers	1	49.1	38.7	55.0	81.2
		3	53.3	48.8	60.7	78.9
		5	57.1	52.0	63.0	80.1
		10	65.0	60.5	71.2	83.5
	Max Incorrect	1	49.1	38.7	55.0	81.2
		3	53.3	48.8	60.7	78.9
5		57.1	52.0	63.0	80.1	

Table 3: Results on the annotated test set. Scores are normalized by the maximum score obtainable with that number of guesses, and therefore may go down as k increases

4.3 Human Performance

To measure human performance against such models, we collected 30 additional human responses per question with the same setup in collecting test data and aggregated them by counts, just as the sampled answers from GPT-2 models were ranked. The last column in table 3 reports this human performance. We can see that the best-performing automatic system is still meaningfully behind human performance in all metrics.

5 Discussion and Analysis

Table 3 shows the results of the baseline models using different measures of similarity, and different measures for the MAX ANSWERS and MAX INCORRECT metrics. One can see that GPT-2 without fine-tuning outperforms the baseline QA implementation, and fine-tuned GPT-2 outperforms both, but a large gap still remains between human performance and any of the baselines, even on the generous RoBERTa-based similarity metric. The human baseline scores are relatively stable regardless of which similarity metric is used, whereas the model scores change drastically (most significantly for the QA model) as more generous similarity metrics are used. We suggest that WordNet Similarity be used as the primary similarity metric as it strikes a reasonable balance between precision and recall,

as discussed in § 5.2.

5.1 Knowledge Base Comparison

To show the dataset indeed containing meaningful commonsense knowledge, we did an additional analysis between our dataset and ConceptNet. ConceptNet (Speer et al., 2017) is a knowledge base containing triples related to common sense which has been shown to be helpful for various downstream tasks (Zhong et al., 2019; Wang et al., 2019) and conversational text generation (Wu et al., 2020; Zhang et al., 2020). We evaluate its potential relevance to this task by evaluating how often a (question, answer cluster) pair has a possible matching triple within ConceptNet. We extract a list of keywords from the question and a ground-truth answer string (by removing stop words) and similarly extract keywords from the head and tail of each ConceptNet relation. We then evaluate whether a given question-answer pair has potential “coverage” in ConceptNet by checking whether a keyword in the question is related to a keyword in the answer. For example, given the question “*Besides music, name something you might hear on a morning radio show*” and the answer “*weather report*”, we would find the triples (*listen to radio, Cause, you hear local weather report*) and (*listen to radio, Has-Subevent, hear weather report*). By this measure, we find that 24.3% of the answer clusters in our

	Precision	Recall	F1
Exact Match	1.0	0.466	0.636
WordNet Similarity	0.996	0.581	0.734
RoBERTa Similarity	0.762	0.661	0.708

Table 4: Measurement of different score function against human cluster assignment.

development set have some match within ConceptNet. This suggests that a common sense KB might provide a useful resource for this task, however ConceptNet has a large number of relations with no direct ability to provide a ranking and thus we exclude such a model from our baseline comparisons. A similar analysis shows that the human baseline match 46.5% of the clusters, whereas a list of 20 top answers from the fine-tuned GPT-2 model match 30.3%.

5.2 Score Function Comparison

In order to compare the various similarity functions outlined in § 3, we manually annotated answers – from both the human baseline and fine-tuned GPT-2 outputs – to the correct answer clusters. Four annotators separately mapped each answer string to an existing cluster.

Table 4 measures how well different similarity functions performed in comparison to the manual human cluster assignment. Precision in this context measures how often an answer assigned by the automatic similarity measure is correctly assigned; recall measures how often an answer which should be assigned to a cluster is correctly assigned. Unsurprisingly, exact match has perfect precision in this context, but has relatively low recall. WordNet similarity increases recall while adding very little false positives. As was hoped, RoBERTa similarity does dramatically increase how often an answer is mapped to the correct cluster, but does so at the expense of a large loss in precision; we therefore suggest that the WordNet similarity is the safest evaluation option.

5.3 Error Analysis

To provide some notion for the tendencies of different models on this task we provide actual model outputs in Table 5. One can see that, before fine-tuning, GPT-2 results are often acceptable and

plausible situations (e.g. refrigerators might be replaced), but can fail to answer the specific criteria requested by the prompt. In contrast, the QA-based model is much noisier – occasionally providing very good answers, but often (as in the examples provided) failing to find answers that are even plausible. Fine-tuned GPT-2, in contrast to both, clearly learns to actually focus upon the expected format and details of such prototypical activities, however it fails in situations where a high-scoring answer would be very rarely discussed, such as knowing that light bulbs are commonly changed around the house.

6 Related Work

A wide variety of common sense reasoning datasets address related topics. Many datasets cover physical and spatial reasoning (Bisk et al., 2019), social common sense (Sap et al., 2019b), and common sense understanding of plausible sequences of events (Zellers et al., 2018, 2019; Huang et al., 2019; Bhagavatula et al., 2019; Sap et al., 2019a) or understanding of the entailments of a sentence (Zhang et al., 2017; Bowman et al., 2015; Roemmele et al., 2011; Levesque et al., 2012). There is also a long history of work in modeling scripts and frames (Schank and Abelson, 1977; Chambers and Jurafsky, 2009; Fillmore et al., 1976; Ferraro and Van Durme, 2016; Weber et al., 2020), which is related to the current focus on prototypical situations.

Recent works have also sought to characterize the ability of pre-trained language models to understand common sense reasoning, showing such models perform well at common sense reasoning tasks even without fine-tuning, allowing one to explore the common sense reasoning inherent in those models (Tamborrino et al., 2020; Weir et al., 2020). Of particular relevance to the current work, Weir et al. (2020) explored the ability of pre-trained models to predict *stereotypic tacit assumptions*, generalizing about entire classes of entities with statements such as “everyone knows that a bear has _____”.

Interestingly, ProtoQA is not the first time FAMILY-FEUD has been referenced in the common-sense literature. Common Consensus (Lieberman and et al., 2007) was a web-based game created with the intention of being a self-sustaining platform to collect and validate commonsense knowledge based on human goals. Prior work had established the idea of using online games to simultaneously entertain and collect commonsense

Prompt	<i>Name something around the house that's often replaced.</i>			
Human	light bulbs	toilet paper	furniture	food
GPT-2	TV	refrigerator	fridge	trash
GPT-2 Fine Tune	dishes	toilet	kitchen	furniture
QA	tune	time	name	song
Prompt	<i>Name something a monk probably would not own</i>			
Human	gun	wife	knife	pornography
GPT-2	gun	car	sword	motorcycle
GPT-2 Fine Tune	weapon	sword	car	cell phone
QA	arch	everything	togashi	power

largest cluster	cluster 2	cluster 3	smaller clusters
-----------------	-----------	-----------	------------------

Table 5: Top responses from human and model predictions for each prompt, color-coded with the answer cluster they might be aligned to

knowledge (Ahn et al., 2006), however the authors of Common Consensus found that the format of FAMILY-FEUD questions was more amenable to high-quality commonsense knowledge acquisition. Common Consensus serves as an excellent proof of concept for future gamification of the style of data presented in this dataset.

ProtoQA differs from other datasets in three different ways:

1. ProtoQA focuses on proto-typical situations. Humans can agree about the details and characteristics of a prototypical event or situation due to commonalities in their shared lived experiences, cultural norms and expectations. This rough agreement extends beyond an agreement on a single top response and that's why our task and evaluation values diversity of answers.
2. The evaluation ProtoQA is a generative evaluation task where a model has to output a ranked list of answers, ideally covering all prototypical answers for a question.
3. ProtoQA has a large number of annotations for each example which makes the generation evaluation possible.

7 Conclusion

We have presented a new common sense dataset with many novel features. The collection of a large set of raw answer strings and further clustering of these strings facilitates a generative evaluation method, enabling actual use of trained models to

answer real common sense questions. The inclusion of counts over clusters of answers provides a very rich dataset for training and evaluation. 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.

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. Millions of people have played phone-based games based upon this same premise⁸, and prior works have obtained valuable annotations from trivia game participants (Rodriguez et al., 2019). This dataset lays the foundation for larger-scale data collection which leverages people's natural interest to encourage high-quality answers to more common sense questions.

Acknowledgments

We thank the IESL and NLP lab at UMass Amherst for their 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.

⁸Based on downloads of <https://play.google.com/store/apps/details?id=com.umi.feudlive>

References

- Luis Von Ahn, Mihir Kedia, and Manuel Blum. 2006. Verbosity: a game for collecting common-sense facts. In *In Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems, volume 1 of Games*, pages 75–78. ACM Press.
- 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 Metrics MATR*, 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*.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010*.
- 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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.
- Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.
- 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*.
- Henry Lieberman and et al. 2007. Common consensus: a web-based game for collecting commonsense goals.
- 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 Saggon, 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*.
- Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*.
- Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. *arXiv preprint arXiv:1904.04792*.
- 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 Le Bras, and Yejin Choi. 2019b. *Social IQa: Commonsense reasoning about social interactions*. pages 4463–4473.
- Roger C Schank and Robert P Abelson. 1975. Scripts, plans, and knowledge. In *IJCAI*, volume 75, pages 151–157.
- Roger C Schank and Robert P Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.
- 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*.
- Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.
- Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. *The woman worked as a babysitter: On biases in language generation*. pages 3407–3412.
- 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*.
- Alexandre Tamborrino, Nicola Pellicano, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training is (almost) all you need: An application to commonsense reasoning.
- 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*.
- Noah Weber, Rachel Rudinger, and Benjamin Van Durme. 2020. Causal inference of script knowledge. *arXiv preprint arXiv:2004.01174*.
- Nathaniel Weir, Adam Poliak, and Benjamin Van Durme. 2020. On the existence of tacit assumptions in contextualized language models.
- Christopher KI Williams and Carl Edward Rasmussen. 1996. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520.
- 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.
- Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2020. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th Annual Meeting of the Association of Computational Linguistics*.
- 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*.
- Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs.
- Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *TACL*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019b. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *ArXiv*, abs/1911.10484.
- 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 $\text{SynsetSim}(X, Y) : S \times S \rightarrow [0, 1]$ be a score for synset similarity, eg.

$$\text{SynsetSim}(X, Y) := \begin{cases} 1 & \text{if } X = Y, \\ 0 & \text{otherwise.} \end{cases}$$

3. A given string may correspond to multiple synsets. Given two strings x and y we define

$$\begin{aligned} \text{SynsetsScore}(x, y) = \\ \max\{\text{SynsetSim}(S_x, S_y) : S_x \in S(x), S_y \in S(y)\}. \end{aligned}$$

4. Some valid answer strings may not correspond to a synset at all, so we define

$$\begin{aligned} \text{SubstringScore} = \\ \max(\text{SynsetsScore}(x, y), \text{ExactMatch}(x, y)) \end{aligned}$$

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

$$\begin{aligned} \text{TokensScore}(A, B) \\ = \max_{m \in M(A, B)} \frac{\sum_{(a, b) \in m} \text{SubstringScore}(a, b)}{\max(|A|, |B|)} \end{aligned}$$

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

$$\begin{aligned} \text{WordNetScore}(x, C) = \\ \max\{\text{TokensScore}(T(x), T(y)) : y \in C\} \end{aligned}$$

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 GPT-2 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 6: Transformation rules from original question sentence to GPT-2 format sentence

In order to make the question more natural for GPT-2 model to answer, we use rule in Table 6 to re-write the questions.

C Criteria for test question acceptance

When creating new questions using the perturbation method described in § 2.2, we scored each question with the following criteria in mind:

- Most people are expected to be able to answer.
- The answer set category is relatively small; less than eight big categories of different answers.
- The question is hard for systems relying on co-occurrence patterns to answer, e.g., BERT
- The answers to the question are not too culturally dependent (e.g., we want to avoid questions such as *Name a dish made with ground meat*).

- Not accidentally re-creating a well-explored question: We then searched all Family Feud data to ensure that no questions were being re-created, and searched online to make sure no obvious lists of answers can be found via search with Google. E.g., if we create a question and the top search for that question is a list of answers to that question, regardless of origin, we remove the question.

D Criteria for stereotypical bias issues

We define a relatively strict measure for stereotypical bias, primarily to avoid having overly problematic examples; we expect that more nuanced issues of stereotypes are common in the data, but are not as easy to measure with an all-or-nothing measure. We rule out questions if they match any of the following:

1. Attaining the right answer requires stereotypes regarding what activities are affiliated with each gender (e.g., that only girls play with dolls)
2. Questions that measure activities a particular gender would be proud or embarrassed to do.
3. We could not find any questions addressing race, sexual orientation, religion, or national origin, but these were searched for and would have also been removed if found.

Types of potentially biased questions which we could not consistently remove from all the training data, but which we note to be worthy of consideration, are:

1. Questions with heteronormative assumptions (questions about what women like, romantically, in men or vice versa)
2. Questions that can be specific to Western US culture: a vast array of questions would have different distributions over answers if asked to people of specific cultures, where stereotypical foods, greetings, habits, or objects may be different.
3. Questions that reference gender, but which might have similar answer clusters if the gender was removed – e.g., *Name something your parents always want to know about the man you're dating.*

E QA model details

For the baseline results reported, we fine-tune the “Bert for QA” model of the Huggingface transformers package, v2.6.0 (Wolf et al., 2019), using BERT-large-uncased (Devlin et al., 2019).

Table 7 illustrates examples of answer strings for the query “name something you do at a concert”, illustrating both that such a method finds passages that are relevant to the questions, but also illustrating the kind of noise being introduced by such a distance learning approach.

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

F GPT-2 model details

For the baseline results reported, we fine-tune GPT-2 Large model using the scrapped training data. The parameter for the best performing model is as follows: batch_size:1, training epoch: 1, gradient accumulation step: 8. The other parameters are the default value in the hugging face implementation. In generation phrase, the temperature is 0.69, top_p is 0.9, and other parameter values are using the default values. All parameters are tuned using dev data, and searched via greed search. The code will be publicly available upon publication.

G Alternative Human Performance Answers

The human performance numbers reported in § 4.3 were collected to be maximally similar to the proposed task: like both the training data and the crowdsourced evaluation data, they were generated by asking many humans for a single best answer. We also collected sets of answers from a small set of in-person annotators using a slightly different questioning paradigm, providing a prompt and asking a single annotator to provide eight different answers to that question. In practice, we found that this shift in evaluation this could penalize human performance. One primary issue with this was that the human annotator asked for all answers to

Prompt	<i>Name something around the house that's often replaced.</i>			
Single-human ranking	food	toilet paper	paper towels	garbage bags
Prompt	<i>Name something a monk probably would not own.</i>			
Single-human ranking	a fancy car	a fancy house	too much food	a bank account
	largest cluster	cluster 2	cluster 3	smaller clusters

Table 8: Top three responses from human ranking evaluation for the same data

Metrics %		Single-Human Ranking	
Exact Match	Max Answers	1	40.5
		3	39.4
		5	41.0
		10	45.6
	Max Incorrect	1	23.9
		3	36.0
5		40.5	
WordNet Similarity	Max Answers	1	45.2
		3	47.8
		5	50.7
		10	55.3
	Max Incorrect	1	29.2
		3	44.6
5		50.6	
RoBERTa Similarity	Max Answers	1	59.0
		3	64.0
		5	66.2
		10	71.7
	Max Incorrect	1	59.0
		3	64.0
5		66.2	

Table 9: Results for the “single human” ranking scores, replaced by a human evaluation closer to actual task

the same question would generally only provide a single answer string corresponding to the top answer clusters. This means that even if the human matched the correct answer, they would miss that answer cluster entirely if they provided a novel string for that answer cluster. Annotators also found it to be quite difficult to provide many answers for the same prompt and would go far afield with later answers, making such answers differ from the distribution of answers in the train and evaluation set. To avoid confusion using these noticeably different human performance scores, we shifted reporting to a set of data that is closer to the actual task evaluation but report those ranking scores here for transparency. One can see from Table 8 and 9 that such human answers look good, but that the actual scores are dramatically lower than what is

seen when humans are evaluated on the same task as the evaluation set, and only barely outperforms a fine-tuned GPT-2 system.