SimpleQuestions Nearly Solved: A New Upperbound and Baseline Approach

Michael Petrochuk[*†] **Luke Zettlemoyer**[*]
[*]Paul G. Allen School of Computer Science & Engineering, Univ. of Washington, Seattle, WA
[†]Allen Institute for Artificial Intelligence, Seattle WA
{mikep5, lsz}@cs.washington.edu

Abstract

The SimpleQuestions dataset is one of the most commonly used benchmarks for studying single-relation factoid questions. In this paper, we present new evidence that this benchmark can be nearly solved by standard methods. First, we show that ambiguity in the data bounds performance at 83.4%; many questions have more than one equally plausible interpretation. Second, we introduce a baseline that sets a new state-of-the-art performance level at 78.1% accuracy, despite using standard methods. Finally, we report an empirical analysis showing that the upperbound is loose; roughly a quarter of the remaining errors are also not resolvable from the linguistic signal. Together, these results suggest that the SimpleQuestions dataset is nearly solved.

1 Introduction

We present new evidence that the SimpleQuestions benchmark (Bordes et al., 2015) can be nearly solved by standard methods. First, we show that ambiguity in the data bounds performance; there are often questions have more than one equally plausible interpretation. Second, we introduce a baseline that sets a new state-of-the-art performance level, despite using standard methods. Finally, we report an empirical analysis showing that the upperbound is loose.

The simple questions task involves mapping an English question (e.g. "Who wrote Gulliver's travels?") to an analogous Freebase (Bollacker et al., 2008) query, used to answer the question. The query consists of a Freebase relation (e.g. /film/film/story_by) and subject (e.g. 090s_0 [gulliver's travels]). To understand how we might bound performance on the SimpleQuestions dataset, our first contribution in this paper, consider the following examples:

a. who wrote gulliver's travels?
(film/film/story_by, 090s_0 [gulliver's travels, TV miniseries])

b. Name a character from gullivers travels.
(book/book/characters, 0btc7 [gulliver's travels])

In example (a) the phrase "Gulliver's travels" is mapped to a TV miniseries, while in (b) it is mapped to a book. This introduces an unintended ambiguity, since either mapping is equally plausible for both examples (i.e. both books and TV miniseries have authors and characters). We introduce a method for automatically identifying many such ambiguities in the data, for both the entities and relations, and show that performance is upperbounded at 83.4%.

Our second main contribution is a baseline that sets a new state-of-the-art performance level, despite using standard methods. Our approach includes (1) a CRF used to tag the mention of the subject in a question and (2) a BiLSTM used to classify the Freebase relation. Despite its simplicity, this approach achieves 78.1% accuracy for predicting Freebase subject-relation queries, surpassing all previous models.

Finally, we present an empirical error analysis of this model which shows the upperbound is loose and that there is likely not much more than 4% of performance to be gained with future work on the data. Together, these results suggest that the SimpleQuestions dataset is nearly solved. Our code and pretrained models are available at github.com/PetrochukM/Simple-Question-Answering.

2 Background

Single-relation factoid questions (simple questions) are common in many settings (e.g. Microsoft's search query logs (Yih et al., 2014) and WikiAnswers web questions (Fader et al., 2013)). The SimpleQuestions dataset is one of the most

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 554–558
Brussels, Belgium, October 31 - November 4, 2018.

commonly used benchmarks for studying such questions.

The Freebase knowledge graph (KG) provides the facts for answering the questions in the SimpleQuestions dataset. It includes 3 billion triples of the form (subject, relation, object) (e.g. [04b5zb_ (Fires Creek), location/location/containedby, 0f80hy (Nantahala National Forest)]). We denote such triples as (s, r, o).

The SimpleQuestions task is to rewrite questions into subject-relation pairs of the form (subject, relation), denoted in this paper as (s, r). Each pair defines a graph query that can be used to answer the corresponding natural language question. The subject is a Freebase object with a identifier called an MID (e.g. 04b5zb_). Freebase objects also typically include one or more string aliases (e.g. MID 04b5zb_ is named "Fires Creek"), which we will use later when computing our upper bounds. The relation is an object property (e.g. location/location/containedby) defined by the Freebase ontology. For example, the question "which forest is fires creek in" corresponds with the subject-relation pair (04b5zb_ [Fires Creek], location/location/containedby). Finally, the SimpleQuestions task is evaluated on subject-relation pair accuracy.

The SimpleQuestions dataset provides a set of 108,442 simple questions; each question is accompanied by a ground truth triple (s, r, o). This dataset also provides two subsets of Freebase: FB2M and FB5M.[1]

3 Dataset Ambiguity and Upperbound

The ambiguity in the SimpleQuestions dataset likely comes from the way the data was created. Annotators were shown a single Freebase triple and asked to write a question. For example, given any of the following triples:

- (0btc7 [Gulliver's Travels, Book], book/written_work/author, o3_dj [Dean Swift])
- (06znpjr [Gulliver's Travels, American film], film/film/written_by, 03whnyn [Nicholas Stroller])

[1]The FB2M and FB5M subsets of Freebase KG can complete 7,188,636 and 7,688,234 graph queries respectively; therefore, the FB5M subset is 6.9% larger than the FB2M subset. More previous research has cited FB2M numbers than FB5M; therefore, we report our numbers on FB2M.

Subject	Description
0btc7	Gulliver's Travels (Book)
090s_0	**Gulliver's Travels (TV miniseries)**
06znpjr	Gulliver's Travels (American film)
02py9bj	Gulliver's Travels (French film)

Table 1: FB2M entities with the alias "gulliver's travels"

Relation	Count
book/written_work/author	132
film/film/written_by	67
film/film/story_by	**9**
...	...

Table 2: SimpleQuestions dataset abstract predicate "who wrote e?" relation count

- (06znpjr [Gulliver's Travels, American film], film/film/story_by, o3_dj [Dean Swift])

The annotator might reasonably contribute the question "who wrote gulliver's travels?" However, adding all of these pairs to the data is problematic. Systems are evaluated on producing the correct subject-relation pair, and cannot learn a deterministic mapping that would get these three examples correct. In this section, we present a simple heuristic method for finding many such instances of ambiguity, and use it to upper bound performance on this benchmark.

3.1 Approach

Given an example question q with the ground truth (s, r, o), our goal is to determine the set of all other subject-relation pairs that are equally supported by the text in q.

We first determine a string alias a for the subject by matching a phrase in q with a Freebase alias for s, in our example yielding "gulliver's travels". For 97% of questions q, some string alias a exactly matched a question q phrase. We then find all other Freebase entities that share this alias a and add them to a set S, in our example S is the subject column of Table 1.

We define an abstract predicate p (e.g. "who wrote e?") as q with alias a abstracted. We determine the set of potential relations R as the relations p co-occurs with in the SimpleQuestions dataset, in our example R is the relation column of Table 2.

Finally, if there exists a subject-relation pair $(s, r) \in KG$ such that $r \in R \wedge s \in S$ we de-

fine that as an accurate semantic interpretation of q. q is unanswerable if there exists multiple valid subject-relation pairs (s, r). In our example above, the question is unanswerable because of the many different subject, relation pairs that co-occur with "gulliver's travels" and "who wrote e?"

3.2 Results

We find that 33.9% of examples in the SimpleQuestions dataset are unanswerable. In these cases, we can predict a majority baseline (i.e. always guess the most commonly seen Freebase entity or relation), yielding an upperbound of 85.2%.

Finally, we also found that 1.8% of example questions were noisy. For example, "Which book is written about?" does not reference the corresponding ground truth subject 01n7q (california). We also consider these examples unanswerable, yielding a final upperbound of 83.4%.

4 Baseline Model

Our second main contribution is a baseline that sets a new state-of-the-art performance level, despite using standard methods. Our approach includes (1) a CRF tagger to determine the subject alias, and (2) a BiLSTM to classify the relation.

4.1 Approach

Given a question q (e.g. "who wrote gulliver's travels?") our model must predict the corresponding subject-relation pair (s, r). We predict (s, r) with a pipeline that first runs top-k subject recognition and then relation classification.

We make use of two learned distributions. The subject recognition model $P(a|q)$ ranges over text spans A within the question q, in our example A includes the correct subject "gulliver's travels". This distribution is modeled with a CRF, as defined in more detail below. The relation classification model $P(r|q, a)$ will be used to select a Freebase relation r that matches q. The distribution ranges over all relations in Freebase that co-occur with a subject that is named a. It is modeled with an LSTM, that encodes q, again as defined in more detail below.

Given these distributions, we predict the final subject-relation pair (s, r) as follows. First, we determine the most likely subject alias a according to $P(a|q)$ that also matches a subject alias in the KG. We define set S as all Freebase entities named a, in our example S is the subject column of Table 1. Second, we define all potential relations R such that $\forall(s, r) \in KG\{r \in R \wedge s \in S\}$. Using the relation classification model $p(r|q, a)$, we predict the most likely relation $r_{max} \in R$.

Now, the answer candidates are subject-relation pairs such that $(s, r_{max}) \in KG\{r \in R \wedge s \in S\}$. In our example question, if r_{max} is film/film/story_by then S includes both subjects 06znpjr (Gullivers Travels, American film) and 02py9bj (Gullivers Travels, French film). Because there is no explicit linguistic signal to disambiguate this choice, we pick the subject that co-occurs most often with r_{max} in Freebase.

4.2 Model Details

Our approach requires two models, in this section we cover training and configuring these models.

Top-K Subject Recognition We model top-k subject recognition $P(a|q)$ using a linear-chain conditional random field tagger (CRF) with a conditional log likelihood loss objective. k candidates are inferred with the top-k Viterbi algorithm.

Our model is trained on a dataset of questions each with their corresponding subject alias span delimited with IO tagging. The gold standard subject alias spans are determined by heuristically matching a phrase in the question with a Freebase alias for the subject.

All hyperparameters are hand tuned and then a limited set are further tuned with grid search to increase validation accuracy. In total we evaluated at most 100 hyperparameter configurations. The word embeddings are initialized with GloVe (Pennington et al., 2014) and frozen. Adam (Kingma and Ba, 2014), initialized with a learning rate of 0.001, is employed to optimize the model weights. Finally, we halve the learning rate if the validation accuracy has not improved in 3 epochs.

Relation Classification The relation classification distribution $P(r|q, a)$ is modeled with a one layer BiLSTM batchnorm softmax classifier. The BiLSTM encodes an abstract predicate string (e.g. "who wrote e?"), as described in Section 4.1. The last LSTM output vector is provided as input to an output block consisting of batch normalization, ReLU, and softmax.

All hyperparameters are hand tuned and then a limited set are further tuned with Hyperband (Li et al., 2017) to increase validation accuracy. Hyperband is allowed at most 30 epochs per model

Previous Work	Acc.
Random guess (Bordes et al., 2015)	4.9
Memory NN (Bordes et al., 2015)	61.6
Attn. LSTM (He and Golub, 2016)	70.9
GRU (Lukovnikov et al., 2017)	71.2
BiGRU-CRF & BiGRU (Mohammed et al., 2017)	73.7
BiLSTM & BiGRU (Mohammed et al., 2017)	74.9
BiGRU & BiGRU (Dai et al., 2016)	75.7
CNN & Attn. CNN & BiLSTM-CRF (Yin et al., 2016)	76.4
HR-BiLSTM & CNN & BiLSTM-CRF (Yu et al., 2017)	77.0
BiLSTM-CRF & BiLSTM (Ours)	**78.1**

Table 3: Summary of past results on the SimpleQuestions benchmark along with the neural models employed. Note that an "&" indicates multiple neural models.

and a total of 1000 epochs. In total we evaluated at most 500 hyperparameter configurations. The word embeddings are initialized with FastText (Bojanowski et al., 2017) and frozen. We use the AMSGrad variant of Adam (Reddi et al., 2018), initialized with an learning rate of 0.001. Finally, we double the batch size (Smith et al., 2017) if the validation accuracy has not improved in 3 epochs.

4.3 Results

Finally, we present our results on the SimpleQuestions test set.

SimpleQuestions Task Our model achieves 78.1% accuracy on the SimpleQuestions test set, a new state-of-the-art without ensembling or data augmentation (Table 3). These results suggest that relatively standard architectures work well when carefully tuned, and approach the level set by our upper bound earlier in the paper. This further confirms the results of Mohammed et al. 2017.

Further Qualitative Analysis We also analyze the remaining errors, to point toward directions for future work.

In Section 3, we showed that questions can provide equal evidence for multiple subject-relation pairs. To remove this ambiguity, we count any of these options as correct, and our performance jumps to 91.5%.

The remaining 8.5% error comes from a number of sources. First, we find that 1.9% of examples were incorrect due to noise, as described in Section 3. To better understand the remaining 6.5% gap, we do an empirical error analysis on a sample of 50 negative examples.

First we found that for 14 of 50 cases the question provided equal linguistic evidence for both the ground truth options and the predicted subject-relation pair, similar to the dataset ambiguity found in Section 3, suggesting that our upper bound is loose. We note that Section 3 did not cover all possible question-subject-relation pair ambiguities. The approach relied on exact string matching to discover ambiguity; therefore, missing other paraphrases. For example, the abstract predicate "what classification is e" had more examples than "what classification is **the** e" allowing our approach to programmatically define more subject-relation pair ambiguities for the former predicate than the latter.

The remaining 36 of 50 cases were linguistic mistakes by our model. Among the 36 cases, we identified these error cases:

- **Low Shot** (16 of 36) The relation label was seen in the training data less than 10 times.
- **Span Identification** (14 of 36) The subject span was incorrectly labeled.
- **Noise** (2 of 36) The question did not make grammatical sense.

Together, this error analysis shows that the upperbound is loose. There is likely not much more than 4% of performance to be gained with future work on the data.

5 Conclusions and Future Work

The SimpleQuestions dataset is one of the most commonly used benchmarks for studying single-relation factoid questions. In this paper, we presented new evidence to suggest that this benchmark can be nearly solved by standard methods. These results suggest there is likely not much more than 4% to be gained with future work on the data.

Finally, other KG (e.g. Freebase) query datasets should consider providing a set of correct subject-relation pairs when there is ambiguity in the linguistic input.

[1] Türe and Jojic 2017 reported a 86.8% accuracy but we and Mohammed et al. 2017 have not been able to replicate their results. Wang et al. 2017 scored 77.5% but removed 0.5% of the test examples.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

Zihang Dai, Lei Li, and Wei Xu. 2016. Cfo: Conditional focused neural question answering with large-scale knowledge bases. *CoRR*, abs/1606.01994.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *ACL*.

Xiaodong He and David Golub. 2016. Character-level question answering with attention. In *EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet S. Talwalkar. 2017. Hyperband: Bandit-based configuration eval- uation for hyperparameter optimization.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *WWW*.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2017. Strong baselines for simple question answering over knowledge graphs with and without neural networks. *CoRR*, abs/1712.01969.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of adam and beyond. In *ICLR*.

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. 2017. Don't decay the learning rate, increase the batch size. *CoRR*, abs/1711.00489.

Ferhan Türe and Oliver Jojic. 2017. No need to pay attention: Simple recurrent neural networks work! In *EMNLP*.

Runze Wang, Chen-Di Zhan, and Zhen-Hua Ling. 2017. Question answering with character-level lstm encoders and model-based data augmentation. In *CCL*.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *ACL*.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *COLING*.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *ACL*.