An empirical analysis of existing systems and datasets toward general simple question answering

Namgi Han[♠♡♢] **Goran Topić**[♠] **Hiroshi Noji**[♠] **Hiroya Takamura**[♠♣] **Yusuke Miyao**[♠♭]
[♠]Artificial Intelligence Research Center, AIST
[♡]The Graduate University for Advanced Studies
[♢]National Institute of Informatics
[♣]Tokyo Institute of Technology
[♭]The University of Tokyo
{han.namgi, goran.topic, hiroshi.noji}@aist.go.jp
takamura@pi.titech.ac.jp yusuke@is.s.u-tokyo.ac.jp

Abstract

In this paper, we evaluate the progress of our field toward solving simple factoid questions over a knowledge base, a practically important problem in natural language interface to database. As in other natural language understanding tasks, a common practice for this task is to train and evaluate a model on a single dataset, and recent studies suggest that SimpleQuestions, the most popular and largest dataset, is nearly solved under this setting. However, this common setting does not evaluate the robustness of the systems outside of the distribution of the used training data. We rigorously evaluate such robustness of existing systems using different datasets. Our analysis, including shifting of training and test datasets and training on a union of the datasets, suggests that our progress in solving SimpleQuestions dataset does not indicate the success of more general simple question answering. We discuss a possible future direction toward this goal.

1 Introduction

Simple factoid question answering over a knowledge base is an important task in natural language understanding. Although it only deals with factoid questions about a single entity and a predicate, they cover much of the real user queries (Dai et al., 2016), and also, accurate mapping of these is a critical subproblem in semantic parsing-based complex query generation (Berant et al., 2013; Bao et al., 2016; Reddy et al., 2016; Trivedi et al., 2017). SimpleQuestions (Bordes et al., 2015) is the largest and most popular dataset on this task. It was recently argued that this task, given abundant training data, is nearly solved with standard techniques in machine learning (Petrochuk and Zettlemoyer, 2018; Mohammed et al., 2018).

In this paper, we present a thorough empirical analysis to assess whether the success of one particular dataset indicates the success of the task itself in general. To this end, we evaluate the behaviors of four existing QA systems targeting SimpleQuestions (Mohammed et al., 2018; Yu et al., 2017; Wu et al., 2019; Huang et al., 2019), across four different datasets (Cai and Yates, 2013; Yih et al., 2016; Bordes et al., 2015; Jiang et al., 2019), under different conditions. One of our research goals is to evaluate the robustness of a model trained on a single dataset against questions that are outside of the distribution of the training data. Such robustness evaluation is recently actively studied in other language understanding tasks (Jia and Liang, 2017; Naik et al., 2018; McCoy et al., 2019; Ribeiro et al., 2020) while little effort has been made on question answering over a knowledge base, though, in practice, it would be critical because a practical system has to be robust on real user queries, which may be outliers in the training data.

Our experiments suggest that, while SimpleQuestions is the largest, the examples are too simple and the success on it does not indicate progress in factoid question answering in general. For example, we show that, under the same training data size, the system's accuracy on SimpleQuestions gets about 10 points higher than that on WebQuestions. Although the simplicity of SimpleQuestions is pointed out in past work (Jiang et al., 2019), our work provides an empirical evidence that this is indeed the case

Proceedings of the 28th International Conference on Computational Linguistics, pages 5321–5334
Barcelona, Spain (Online), December 8-13, 2020

with a careful comparison and manual analysis using the standardized datasets. Given our analysis, we suggest two possible future directions. One is to invent a clever novel data creation method that would be scalable while avoiding bias as much as possible. In this respect, we point out that a recent attempt by FreebaseQA (Jiang et al., 2019) is not successful, and that significant bias still exists. Another is to exploit useful information from the large dataset of SimpleQuestions in a better way. In the last analysis, we demonstrate that a simple approach of training on a union of the datasets (Talmor and Berant, 2019) is not satisfactory toward this end, calling for a more sophisticated method of exploiting useful features across datasets effectively.

2 Datasets

We use four QA datasets over a knowledge base (KB) as our target datasets. These datasets were selected because they share a common KB (Freebase), and a large portion of each dataset comprises of factoid questions, which are the main focus of this paper. A factoid question asks a single fact, or a triple (subject, predicate, object) on a KB, where the object corresponds to the answer. For example, "*Which country is Albert Bolender from?*" corresponds to a fact (`Albert_Bolender`, `people.person.nationality`, ?), where the placeholder found in the KB is the target object, which is `United_States`.

Free917 (Cai and Yates, 2013) This is the first dataset for machine learning-based semantic parsing over Freebase. It contains 917 questions on a subset of Freebase, called *Freebase Commons*, covering 81 domains. Berant et al. (2013) find that each question tends to contain words that are directly related to the target Freebase predicate. An example is "*What genre of music is B12?*", for which the gold predicate is `music.artist.genre`. We use an annotated Free917 dataset by Bast and Haussmann (2015).[1]

WebQSP (Yih et al., 2016) This is an extension of WebQuestions (Berant et al., 2013) with gold SPARQL query on each question, which is missing in the original dataset. Aiming at creating more natural questions than Free917, each question is derived from the Google Suggest API, followed by filtering by crowd workers. Consequently, the authors observe a larger divergence between the question words and predicates, such as "*What music did Beethoven compose?*", for the aforementioned predicate `music.artist.genre`. This dataset contains 4,737 questions.

SimpleQuestions (Bordes et al., 2015) This is the largest dataset in our experiments, containing over 100,000 questions answerable by a single fact. Contrary to WebQuestions, each question in this dataset is created from a sampled fact in Freebase, which is then verbalized and paraphrased by a crowd worker. Possibly due to this procedure starting from a KB fact, we find that, as in Free917, this dataset also tends to verbalize a predicate with directly related terms, such as "*What type of music ...?*" for `music.artist.genre`.[2] This approach eases the collection of a lot of data and is popular in data creation for semantic parsing (Wang et al., 2015; Trivedi et al., 2017; Talmor and Berant, 2018). However, we will see in Section 4.3 that it also tends to introduce certain biases, which affect models' generalization. The authors also define a subset of Freebase called FB2M that covers 2M entities and 5K predicates, including all entities appearing in WebQuestions, and create all questions from this subset.

FreebaseQA (Jiang et al., 2019) This is the latest dataset aiming at more difficult factoid questions than SimpleQuestions while maintaining the scale of data size. Specifically, the questions in this dataset are first sampled from TriviaQA (Joshi et al., 2017) and then filtered by heuristics to collect factoid questions answerable on Freebase. Although the authors argue that their procedure reliably eliminates non-factoid questions, we find several problems in this dataset, which we describe in Section 4.2.

2.1 Preprocessing

Apart from the difference in construction methods, the four datasets additionally differ based on (1) whether they contain non-factoid questions and (2) the assumed subset of Freebase. Because we aim

[1] `https://github.com/ad-freiburg/aqqu`

[2] Cai and Yates (2013) only mention that questions are written by two native English speakers and do not state whether they access to a predicate when writing questions, but we find two datasets are similar in this respect.

Dataset	Original			One triple questions			Answerable by FB2M		
	Training	Valid	Test	Training	Valid	Test	Training	Valid	Test
Free917	512	129	276	0	0	637	0	0	347
WebQSP	2,478	620	1,639	1,350	338	915	1,292	323	861
SimpleQuestions	75,910	10,845	21,687	75,910	10,845	21,687	75,895	10,843	21,680
FreebaseQA	20,358	3,994	3,996	13,495	2,660	2,677	10,427	2,048	2,102

Table 1: Data statistics after preprocessing (number of examples). We use "Answerable by FB2M" subset in this paper. Since Free917 is small, we use the entire dataset as the test set.

Training	Validation	# of questions	Training	Validation	# of questions	Training	Validation	# of questions
FBQ	FBQ	71 (3.47%)	SQ	FBQ	137 (6.69%)	WQ	FBQ	1,068 (52.15%)
	SQ	2,582 (23.87%)		SQ	71 (0.66%)		SQ	6,862 (63.45%)
	WQ	52 (16.15%)		WQ	19 (5.90%)		WQ	26 (8.07%)

Table 2: Numbers of examples with unseen relations across one training set and one validation set. The number in a bracket denotes a ratio in the validation split. For example, 71 (3.47%) examples in the valid set of FBQ contain relations not appearing in the training set of FBQ.

to evaluate the behavior of a single model across these four datasets, we perform some preprocessing on each dataset to eliminate those factors. Specifically, from all datasets, we filter questions that do not match the domain of SimpleQuestions; that is, we remove the questions that involve a multi-hop path or multi constraints, such as "*What character did Natalie Portman play in Star Wars?*" in WebQSP, and questions with entities or predicates that are outside of FB2M. Table 1 shows the resulting statistics of each dataset.[3] Unfortunately, because this procedure makes Free917 too small, we use the entire dataset as the test set. Following Berant et al. (2013), we take 20% of the training split as the validation split for WebQSP. We abbreviate these four datasets as F917, FBQ, WQ, and SQ, respectively.

Table 2 summarizes how much of the predicates in one dataset (valid split) are unseen (i.e., zero-shot) in another dataset (training split). Since zero-shot prediction is hard (Wu et al., 2019), we use these as a rough estimate on the difficulty of an experiment in Section 4.

3 Systems

Now we describe four systems that we compare across the datasets. Since we only deal with single fact questions in this paper, all questions can be answered by correctly predicting a subject entity e and a relation r on the KB. To search for the best pair, all systems in this paper employ a pipeline, which is comprised of three different submodules below:

1. entity linking, which outputs a set of candidate subject entities $\{e\}$;
2. relation prediction, which outputs a set of candidate predicates $\{r\}$; and
3. query generation, which finds the best $(\hat{e}, \hat{r})$ pair by reranking the candidate pairs.

While some earlier systems, such as Bordes et al. (2015), employ a different approach, their accuracies are not state-of-the-art. We do not include these in our experiments.

The systems differ in each submodule. There are also some minor variations in pipeline constructions. We select the four target systems considering their high accuracies on SimpleQuestions, as well as the availability of the code.[4]

BuboQA (Mohammed et al., 2018)[5] In this system, both entity linking and relation prediction are modeled with simple classifiers. Despite its simplicity, this approach outperforms several more complex

[3]The reason for the decrease in the first step for FreebaseQA is that it contains two-hop questions involving a mediator node in Freebase, which we exclude from the target.

[4]When searching for open software, we often found that many systems along with a paper are not self-contained; in particular, they often are missing an entity linking module. This is especially the case for systems targeting WebQuestions, for which many systems rely on the outputs of the entity linker used in Yih et al. (2015) and found in `https://github.com/scottyih/STAGG`, while the entity linker itself is not available.

[5]`https://github.com/castorini/BuboQA`

architectures (Bordes et al., 2015; Yin et al., 2016). Specifically, for entity linking, a trained LSTM first detects the entity spans, which are then heuristically mapped to the candidate KB entities and scored with the Levenshtein distance to the canonical entity label. Relation prediction is performed independently by another classifier on top of a different LSTM. Finally, the best combination of $(\hat{e}, \hat{r})$ is found according to a weighted sum of these two module scores.[6] This is an extension of an even simpler baseline of Ture and Jojic (2017), and a similar approach is employed in Petrochuk and Zettlemoyer (2018).

Note that this system treats relation prediction as classification among the predicates appearing in the training data. This means that it cannot solve zero-shot relation prediction, which occurs to some extent especially in the dataset transfer experiment (Section 4.3). On the other hand, the other three systems theoretically can handle them, as described in the following.

Hierarchical Residual BiLSTM (HR-BiLSTM) (Yu et al., 2017) On this system (and the next, KBQA-Adapter), relation prediction is performed differently, not by classification on a fixed set of relations, but by mapping on a shared embedding space for KB relations and texts. This model simply encodes both question tokens and relation tokens (e.g., "*music artist genre*" for `music.artist.genre`) by different encoders. Relation candidates are then ranked by cosine similarity between the outputs of two encoders. This method allows us to calculate the score of an unseen relation. For this system, since Yu et al. (2017) do not release their code, we use the implementation by Wu et al. (2019). Unfortunately, this implementation only includes the relation prediction module rather than the full pipeline. We thus try to reproduce the pipeline described in Yu et al. (2017), using the entity linking module of BuboQA. See Appendix A for details. We use the same pipeline for KBQA-Adapter.

KBQA-Adapter (Wu et al., 2019)[7] This is an improvement to HR-BiLSTM with an additional adversarial adapter coupled with the relation encoder. The motivation of this adapter is to improve the zero-shot relation prediction performance. To this end, the adapter receives a relation embedding for r provided by KG embeddings, which is JointNRE (Han et al., 2018), transforming it to an embedding space where unseen relations can be handled properly. We employ the same pipeline with the BuboQA entity linker for this system.

Knowledge Embedding-based QA (KEQA) (Huang et al., 2019)[8] This system also builds on an external knowledge graph embedding, TransE (Bordes et al., 2013), which is used as the more direct and central part in the system. Given a knowledge graph embedding, which is fixed, this model tries to map each question into an entity embedding $\hat{\mathbf{e}}$ and relation embedding $\hat{\mathbf{r}}$, using separate LSTMs. We expect $\hat{\mathbf{e}}$ to be close to the gold node embedding in the graph and $\hat{\mathbf{r}}$ to the gold relation embedding. Also, we would expect that the transition defined by the embedding model (e.g., addition for TransE), $f(\hat{\mathbf{e}}, \hat{\mathbf{r}})$, will get close to the answer node embedding. The query generation step of the system selects the $(\hat{e}, \hat{r}, \hat{o})$ triple based on this intuition, by minimizing the summed distances from embeddings corresponding to $\hat{e}$, $\hat{r}$, and $\hat{o}$ to the obtained encoded embeddings.

Settings and Notes[9] For all models, we employ the best architectures and hyperparameters reported in the paper or a related document. For the first three systems, we set the number of entity linking outputs as 50, and that of relation prediction as 5, which are the default settings for BuboQA. Note also that these three systems share the same entity linking module of BuboQA. For evaluation, following the standard practice of SimpleQuestions, we evaluate the accuracy of predicted (subject, predicate) pairs.

4 Experiments

4.1 Results when trained on single datasets

Here, we evaluate different models (Section 3) primarily suggested to solve SimpleQuestions across the normalized datasets (Section 2). In addition to the standard experiment on a single dataset, we also

[6] Although the paper mentions that the two scores are multiplied, in the implementation they are summed with fixed weights.
[7] `https://github.com/wudapeng268/KBQA-Adapter`
[8] `https://github.com/xhuang31/KEQA_WSDM19`
[9] You can find the code used in our experiments at `https://github.com/aistairc/simple-qa-analysis`.

Training dataset	Test dataset	BuboQA	HR-BiLSTM	KBQA-Adapter	KEQA
FBQ	F917	17.29	**36.31**	35.73	36.02
	FBQ	**38.25**	28.40	28.78	28.73
	SQ	23.77	38.55	39.19	**42.97**
	WQ	29.10	30.27	31.43	**33.18**
SQ	F917	40.92	56.20	**59.37**	45.24
	FBQ	**20.08**	17.84	18.13	14.03
	SQ	74.81	72.30	72.01	**75.35**
	WQ	**41.79**	35.27	36.32	40.40
WQ	F917	12.68	29.97	29.39	**32.85**
	FBQ	7.94	7.61	8.37	**8.90**
	SQ	16.46	33.18	35.32	**38.01**
	WQ	61.23	49.94	49.36	**65.19**

Table 3: Comparison of top-1 accuracies across datasets (on the test set). The bold value denotes the highest accuracy in each row.

provide an experiment across two datasets, where we train a model on one dataset and test on another. We are interested in this setting, because in a practical scenario, there might be a gap between the distribution of training data, which depends on the way the data was created, and that of test data, which would be real user queries. For example, as we saw in Section 2, the data creation of SimpleQuestions allows collecting a lot of data easily while the data distribution of WebQuestions may match the distribution in the wild. We evaluate these models' robustness to the shift of data distributions.

Table 3 summarizes the main results on the test data. The grey rows correspond to the single dataset settings. Comparing these three rows, the accuracies on FBQ and WQ are consistently lower than SQ, suggesting that FBQ and WQ have some data characteristics that cause difficulties for the current models, which we inspect in detail in Section 4.2. When evaluated on a different dataset, which we call dataset transfer in the following, the accuracies degrade even more. Note that F917 is used as a test-only dataset (Section 2), and the accuracy on it is relatively high when trained on SQ. As we discussed in Section 2, SQ and F917 are somewhat similar. This suggests that, as can be expected, the accuracies on this transfer setting are affected by some notion of distance between datasets and the current models are quite sensitive to it. We will analyze in Section 4.3 what exactly is the main cause of these degradations.

4.2 What makes WebQSP and FreebaseQA more difficult?

WQ and FBQ are more challenging than SQ according to Table 3 (gray rows). Understanding the cause of this difficulty is important because it directly relates to the remaining challenges in solving factoid questions in general. We test several possibilities including the dataset quality to reach an accurate answer.

Upperbounds due to the ambiguity are not the reason Petrochuk and Zettlemoyer (2018) find that the upperbound accuracy of SQ is around 83% due to the inherent ambiguity in the data; e.g., given a question "*who wrote gulliver's travels?*", there is more than one equally plausible interpretation since there are multiple entities for *guliver's travels* such as the book, TV miniseries, and films, all of which could be compatible with "*who wrote ...?*". To test the possibility that lower accuracies on WQ and FBQ are due to even more severe ambiguity in the data, we perform the same analysis on FBQ and WQ, finding that the upperbounds are 86.85% for WQ and 84.16% for FBQ, respectively, which are comparable to SQ. This rejects the possibility that the upperbounds for these two datasets are low.

Quality of FreebaseQA is not high Inspecting datasets, we find that some questions in FBQ are not a factoid question, such as "*What is the highest volcano in Africa?*", which requires an aggregate operation but the gold subject and predicate are just `(Africa, location.contains)`. We suspect that these questions remain in FBQ due to noisy filtering from unrestricted questions, which only assesses

Label	F917	FBQ	SQ	WQ
impossible	1	13	1	4
notsimple	0	15	5	1
badgold	0	12	4	5
multisubj	1	9	0	1
multirel	1	4	2	1
other	0	1	3	0
okay	97	46	85	88

Table 4: Labeling results on random 100 questions from the validation split for each dataset.

Label	Example	Details
impossible	*In the musical Annie, what is Orphan Annie's dog called?*	There is no identifier for Annie's dog in FB2M.
notsimple	*What is the highest peak on Dartmoor?*	The highest cannot be evaluated in a single triple.
badgold	*Where was Princess Leia raised?*	The gold relation is `place_of_birth`, but Leia was raised elsewhere since infancy.
multisubj	*Who wrote the novels "Berlin Game", "Mexico Set" and "London Match"?*	*Berlin Game*, *Mexico Set*, and *London Match* can all derive the correct answer.
multirel	*Where is South Salt Lake, Utah located?*	Both *location.hud_county_place.county* and *location.location.containedby* can be the correct relation.
other	*What operating system uses ssh file transfer protocol?*	Not operating systems, but sshftp programs use sshftp.

Table 5: Examples for the labels used in Table 4.

the path from a subject to an object with little care for additional constraints. The overall quality might be exacerbated by a reliance on non-experts (crowds) for the final assessment.

To quantify how much of the examples are problematic, we sample 100 questions from the validation split on each dataset and categorize them with the labels defined in Table 5. Table 4 is the result. For this labeling, *impossible, notsimple*, and *badgold* labels indicate non-faithful (question, gold label) pairs as in the above example, while *multisubj* and *multirel* are rather the problems due to the evaluation method, because they mean that there are multiple correct labels while the current evaluation only allows a gold one. From Table 4, we can see that 40% of questions in FBQ are non-faithful, much higher than the other datasets. From this result, we argue that lower accuracies on FBQ are not due to the true difficulty as factoid questions, but rather due to the undesirable complexity incurred by an inaccurate data creation process. Considering this problem, we will pay little attention to this dataset in the following analysis.

Data size does not account for the gap between WQ and SQ One major difference between SQ and WQ is the training data size (Table 1), with SQ being roughly 60 times larger. Is this data size the main source of the performance gap seen in Table 3? Or, is it due to the inherent complexity of WQ compared to SQ? To answer this question, we compare SQ and WQ eliminating the data size effects, by preparing a smaller SQ dataset, which has an equal size as WQ. When sampling data from SQ, we only sample examples with predicates that appear in the corresponding split of WQ. We also keep the ratio of unseen relations in the validation split as roughly 8%, the same as WQ (Table 2). We create 10 different subsets of SQ and report the average accuracies on them. We evaluate the systems on the validation splits.

In Table 6, we summarize the scores of BuboQA and KEQA, which perform better on original SQ and WQ in Table 3. Interestingly, the accuracies on small-sized SQ are the same level as those of the original dataset. This indicates that the main factor causing the performance gap between WQ and SQ is not the data size, but the complexity or the inherent difficulty of the dataset, which we inspect next.

Entity linking is challenging Among the three steps in the systems (Section 3), we hypothesize that relation prediction is the main bottleneck on WQ since predicates tend to be nontrivially verbalized compared to SQ (Section 2). Table 7 shows in particular for BuboQA that this is not the case. Here, we evaluate the component-wise performance of entity linking (EL) and relation prediction (RP). We evalu-

Dataset	BuboQA	KEQA
SQ (valid)	75.79	76.69
Small-sized SQ (valid)	73.47±1.69	76.27±2.39
WQ (valid)	59.32	66.15

Table 6: Comparison of end-to-end accuracies (on the validation split) across SQ, small-sized SQ, and WQ. The scores for small-sized SQ are averaged across 10 cases (see body).

Dataset	BuboQA-Final	BuboQA-EL	BuboQA-RP	KEQA-Final	KEQA-EL	KEQA-RP
FBQ	38.25	58.28	**81.21**	28.73	47.62	**55.42**
SQ	74.81	90.40	**95.64**	75.35	90.74	**94.38**
WQ	61.23	78.23	**90.92**	65.19	82.75	**84.97**

Table 7: Comparison of module-level accuracies (R@50 for entity linking (EL) and R@5 for relation prediction (RP)) for BuboQA and KEQA. "Final" denotes end-to-end top-1 accuracies.

Training	Test	BuboQA-Final	BuboQA-EL	BuboQA-RP	KEQA-Final	KEQA-EL	KEQA-RP
FBQ	F917	17.29	70.32	29.11	36.02	60.81	60.23
	SQ	23.77(−51.04)	71.96(−18.44)	39.79(−55.85)	41.83(−33.52)	69.94(−20.80)	71.79(−22.59)
	WQ	29.10(−32.13)	69.85(−8.38)	59.95(−30.97)	33.18(−32.01)	75.32(−7.43)	62.86(−22.11)
SQ	F917	40.92	85.30	55.04	45.24	69.45	69.45
	FBQ	20.08(−18.17)	48.62(−9.66)	49.00(−32.21)	14.03(−14.70)	34.06(−13.56)	37.73(−17.69)
	WQ	41.79(−19.44)	75.90(−2.33)	78.11(−12.81)	40.40(−24.79)	74.62(−08.13)	67.05(−17.92)
WQ	F917	12.68	65.99	18.44	32.85	59.08	54.47
	FBQ	07.94(−30.31)	35.25(−23.03)	24.79(−56.42)	08.90(−19.83)	36.20(−11.42)	26.07(−29.35)
	SQ	16.46(−58.35)	66.71(−23.69)	25.00(−70.64)	38.01(−37.34)	68.49(−22.25)	65.00(−29.38)

Table 8: Comparison of module-level accuracies in the dataset transfer setting. Final: end-to-end accuracy; EL: R@50; and RP: R@5. The number in brackets denotes the difference from the non-transfer baseline (Table 7). The cells for FBQ are represented in gray considering the issues in the dataset.

ate R@50 for EL and R@5 for RP, which are the sizes of candidates in two components of BuboQA.[10] We can see that for both systems EL scores degrade about 10 points from SQ to WQ, which is roughly the same level as decreases in final accuracies. Accuracy of entity linking is critical for both systems because, at the final query generation step, predicate candidates are restricted to ones connected to the selected entities. This means that if the entity linking performs poorly, that can be a bottleneck of the entire system. KEQA suffers from a larger decrease of RP (94.38→84.97) than BuboQA (95.64→90.92), but we conjecture that this can be mainly attributed to the dependence of RP on EL for KEQA (footnote 10).

Inspecting the errors of entity linking by BuboQA, we find a particularly challenging case, specific to WQ, is the superficially ambiguous entities, such as "*Mexico*", which matches to more than 1,000 different entities in Freebase, according to the inverted index by BuboQA. In the top candidates, we notice that many entities are song and album names. The handling of these ambiguous entities is challenging for BuboQA since it does not rely on statistical techniques for disambiguation (only the Levinstein distance). This suggests that we need a more sophisticated entity linker exploiting a context for disambiguation. KEQA's approach is promising, but the current system has an opposite problem, as we discuss in the next section.

4.3 What makes dataset transfer challenging?

So far we have seen that the system's performance gaps between two datasets, SQ and WQ, largely come from the gaps in entity linking performance. Can the same explanation hold for the large gaps with the

[10] For KEQA, we get the same numbers of candidates for EL and RP that are closest to the predicted embeddings in the vector space. In this process, we restrict the candidates for RP as ones that are connected to one of the entity candidates, mimicking the final process of the system.

Label	BuboQA	HR-BiLSTM	KBQA-Adapter	KEQA
relnotfound	**8**	**3**	**7**	**9**
wrongent	14	13	12	35
wrongrel	**23**	**23**	**21**	**31**
ambient	2	1	1	-
ambirel	**29**	**27**	**18**	**24**
unknown	7	-	-	-
other	-	-	-	1
Total	83	67	59	100

Table 9: Labeling of errors on examples (in the validation set of WQ), which are missed by changing the training data from WQ to SQ. Bold font denotes the errors on relation prediction.

Label	Example	Datails
relnotfound	*Who was vice president under Lincoln?*	Gold relation `us_president.vice_president` is an unseen relation (not appear in the SQ training split).
wrongent	*What to do with kids in phx az?*	The systems finds a different entity than the correct entity `Phoenix, Arizona`.
wrongrel	*What money is used in England?*	The systems finds a different relation than the correct `location.country.currency_used`.
ambient	*Where were the Chickasaw Indians located?*	Predicted entity is `Chickasaw Nation` while gold entity is `Chickasaw`. Both are OK on Freebase.
ambirel	*Who is Aidan Davis?*	Gold answer is `people.person.profession`, but prediction is `common.topic.notable_types`.
unknown	*Where was the battle of Antietam creek?*	The system outputs nothing by failing to bridge predicted entities and relations.
other	*What is the actual current local time now in uk?*	Freebase cannot answer the current time.

Table 10: Examples for the labels used in Table 9

dataset transfer setting in Table 3? To answer this question, Table 8 summarizes the submodule accuracies for the transfer setting, on which the numbers in parentheses are degradations from the *non-transfer* setting. For example, R@50 of BuboQA's entity linking drops 2.33 points on WQ, when changing training data from WQ to SQ. From the table, we can see that score drops are more severe in relation prediction. We conjecture that entity linking is less affected by transfer because expressions of entities (e.g., the name of a person) are relatively fixed compared to predicates across datasets.

To confirm what kinds of questions become hard by shifting training data, we manually analyze errors on examples from SQ→WQ case in Table 8. This analysis is on the validation split. For each system, we select up to 100 examples, which are originally solved, but failed when trained on WQ, and categorize the errors according to Table 10. If multiple labels would apply, we choose the highest one from the table. Since an entity linking error often accompanies a relation prediction error (Section 4.2), we prioritize errors related to entity linking (under the same category). The top priority for *relnotfound* (zero-shot relation prediction) is under the assumption that they are particularly hard for models.

Table 9 shows the result. Note that the total numbers are not 100 for some systems, because we only consider examples that original models (trained on WQ) answer correctly. We can see that errors related to relation prediction are dominant across systems, which is consistent with Table 8. We distinguish two types of relation errors: *wrongrel* means a totally wrong prediction while *ambirel* is a spurious error, for which, the predicted relation leads to the correct answer on Freebase, but the current label-based metric penalizes it. We find that most of this latter case occurs by ambiguities of `profession` and `notable_types`, which are often aliases. For a question "*who is . . .?*", the gold relation of SQ is often `notable_types`, but that is often `profession` in WQ. This can be seen as a kind of dataset bias, and one way to resolve it is to change the evaluation metric to evaluate the answers, not labels. Under the current metric, this can be seen as an inherent limitation of solving all questions under the dataset transfer setting. While these are spurious, the other half of relation prediction errors are *wrongrel*. We

Test dataset	BuboQA	HR-BiLSTM	KBQA-Adpater	KEQA
F917	43.80(+1.98)	55.33(−0.87)	59.08(−0.29)	46.69(+1.45)
FBQ	36.01(−2.24)	28.64(+0.24)	26.55(−2.23)	27.07(−1.66)
SQ	74.18(−0.63)	71.87(−0.43)	71.56(−0.45)	74.89(−0.46)
WQ	60.65(−0.58)	45.05(−4.89)	46.33(−3.03)	61.35(−3.84)

Table 11: The final top-1 accuracies by a single model trained on a union of FBQ, SQ, WQ training set. The number in brackets denotes the difference from the model trained on a single target dataset (in Table 3). F917 is compared with the best model (best training data) for each system.

find that these are essentially due to different paraphrasing patterns of a predicate across datasets, as we discussed in Section 2, and this result suggests such variation for a predicate is the main challenge for the transfer.

Finally, we notice that KEQA contains more entity linking errors (*wrongent*), and in many cases, these errors are distinguished in that they are completely irrelevant to the target entity. This suggests that the KEQA entity linker would be more affected by a dataset bias, possibly due to not relying on a string match when linking. An interesting future direction is an extension with additional features to take into account the surface similarities as in BuboQA, which would lead to more robust generalization.

4.4 Effects of combining datasets

Our final experiment is to see the performance of a model trained on the union of the target datasets. This is inspired by the recent success of MultiQA (Talmor and Berant, 2019), which, on reading comprehension, shows that a single model trained on the union of multiple datasets outperforms a model trained specifically on each single dataset. We combine training data of FBQ, SQ, and WQ, and train a model on it. We are particularly interested in whether the accuracy of WQ improves with the help of statistical cues from other datasets, although we have seen that the transfer from SQ only is hard. Table 11 is the result along with the amount of increase/decrease from a model trained on the single dataset (corresponding to the test data). We can see that the model can handle each dataset well on average, but in most cases, the scores do not improve from the single dataset baselines.

This result might be reasonable from our detailed analysis so far. In Section 4.2, we find that the main challenge on remaining errors of WQ is in ambiguous and difficult cases of entity linking. However, entity linking of BuboQA is lexical pattern-based, not statistical, indicating that additional statistical cues from SQ are not very helpful for saving the difficult cases. For KEQA, we find that its entity linking performance is worse on WQ when trained only on SQ (Table 9). This suggests that, although it is statistical, KEQA does not exploit useful features from SQ examples to handle WQ, at least regarding entity linking. A better model or a learning method could utilize the data with different distribution in a clever way, but our analysis suggests that current methods do not have such an ability.

5 Conclusion

Through several experiments, we have shown that although the system performance on SimpleQuestions dataset is getting better and close to the upper bound, that does not indicate a more general success of simple factoid question answering overall. The main cause of this mismatch is that, as we have seen, there is often an inverse relationship between the *ease* of data collection and *naturalness* of collected questions. We found that although the data creation of SimpleQuestions, starting from a KB fact and verbalizing by a crowd worker, is advantageous in terms of scalability, the resulting dataset is too simple, as we demonstrated that the systems can achieve high accuracies even with a limited amount of training data. WebQuestions, on the other hand, is a collection of real user queries with several challenges including ambiguous entity mentions, but such questions are more difficult to collect, in particular in terms of the coverage of entities and relations. It is ideal that systems trained on a simpler and scalable dataset become robust on the questions outside of the distribution of the training data, but our experiment on dataset transfer suggests that the current approaches do not achieve this.

We suppose there are two possible directions toward general simple question answering, or question answering over a knowledge base in general. The first is to improve the dataset quality. We need to create a dataset, that is real and challenging, while still being scalable. FreebaseQA can be seen as an attempt toward this goal, but we found that this dataset has several issues. Another direction is to invent a model or a learning mechanism that can generalize robustly from biased datasets. Our data union can be seen as a simple approach toward this end, but we found that current models do not exploit useful information beyond each target dataset. More sophisticated approaches, such as distributionally robust optimization (Delage and Ye, 2010; Oren et al., 2019), may help. Another promising way is relying on strong pretrained language models, including BERT (Devlin et al., 2019). We have not included BERT-based models in this paper, because its application on SimpleQuestion has not outperformed a simpler baseline so far (Lukovnikov et al., 2019), and it is also nontrivial to integrate BERT with knowledge graph embeddings, which is necessary for KEQA-based approach and is currently actively studied (Peters et al., 2019; Weijie et al., 2020). The integration of such approaches, along with robustness evaluation as done in this paper, will be of practical importance toward robust question answering not specific to a single dataset.

Acknowledgements

This paper is based on results obtained from projects JPNP20006 and JPNP15009, commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and also with the support of RIKEN–AIST Joint Research Fund (Feasibility study). For experiments, computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

References

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Hannah Bast and Elmar Haussmann. 2015. More accurate question answering on freebase. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1431–1440, New York, NY, USA. Association for Computing Machinery.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

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

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria, August. Association for Computational Linguistics.

Zihang Dai, Lei Li, and Wei Xu. 2016. CFO: Conditional focused neural question answering with large-scale knowledge bases. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 800–810, Berlin, Germany, August. Association for Computational Linguistics.

Erick Delage and Yinyu Ye. 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4832–4839. AAAI Press.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 105–113, New York, NY, USA. ACM.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September. Association for Computational Linguistics.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 318–323, Minneapolis, Minnesota, June. 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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July. Association for Computational Linguistics.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. 2019. Pretrained transformers for simple question answering over knowledge graphs. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, pages 470–486.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July. Association for Computational Linguistics.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana, June. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. 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 4227–4237, Hong Kong, China, November. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China, November. Association for Computational Linguistics.

Michael Petrochuk and Luke Zettlemoyer. 2018. SimpleQuestions nearly solved: A new upperbound and baseline approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 554–558, Brussels, Belgium, October-November. Association for Computational Linguistics.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana, June. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy, July. Association for Computational Linguistics.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In Claudia d'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange, and Jeff Heflin, editors, *The Semantic Web – ISWC 2017*, pages 210–218, Cham. Springer International Publishing.

Ferhan Ture and Oliver Jojic. 2017. No need to pay attention: Simple recurrent neural networks work! In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2866–2872, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342, Beijing, China, July. Association for Computational Linguistics.

Liu Weijie, Zhou Peng, Zhao Zhe, Wang Zhiruo, Ju Qi, Deng Haotang, and Wang Ping. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI 2020*.

Peng Wu, Shujian Huang, Rongxiang Weng, Zaixiang Zheng, Jianbing Zhang, Xiaohui Yan, and Jiajun Chen. 2019. Learning representation mapping for relation detection in knowledge base question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6139, Florence, Italy, July. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany, August. Association for Computational Linguistics.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581, Vancouver, Canada, July. Association for Computational Linguistics.

Appendix A The pipeline for HR-BiLSTM and KBQA-Adapter

In this pipeline, rather than the three steps for BuboQA, entity linking will be performed twice. Specifically, the initial entity linking step outputs $s_{\text{linker}}(e; q)$, which is a score for entity e on question q, for K-best candidates, and for each relation r that is connected from these candidate entities, relation score $s_{\text{rel}}(r; q)$ is calculated by the relation prediction module. These scores are then used to recalculate the final entity scores $s_{\text{rerank}}(e; q)$, which will be used in the query generation step along with $s_{\text{linker}}(e; q)$ for ranking (e, r).

Although Yu et al. (2017) report the equation for $s_{\text{rerank}}(e; q)$ their hyperparameters are not specified, and we find that reproducing their results with that equation is difficult. We instead find that calculating $s_{\text{rerank}}(e; q)$ by a similar equation to the one used in the final step of BuboQA works well:

$$s_{\text{rerank}}(e; q) = 0.6 \cdot s_{\text{linker}}(e; q) + 0.1 \cdot \max_{r \in R_e} s_{\text{rel}}(r; q). \quad (1)$$

The weights (0.6 and 0.1) are the default values for BuboQA. Given these scores, the final query generation will be done by the same module as BuboQA.

Apart from Yu et al. (2017) we do not reduce the number of entity candidates with entity reranking (Eq 1). This means we just rescore the 50 candidates found by the BuboQA entity linker.

Appendix B All results of combining two datasets

Training	Test	BuboQA	HR-BiLSTM	KBQA-Adpater	KEQA
FBQ+SQ	**F917**	44.96(+04.04)	57.06(+00.86)	59.37(+00.00)	46.69(+01.45)
	FBQ	34.58(−03.67)	28.73(+00.33)	26.97(−01.81)	27.50(−01.23)
	SQ	74.33(−00.48)	72.37(+00.07)	71.65(−00.36)	74.90(−00.45)
	WQ	46.80(−14.43)	33.88(−16.06)	37.83(−11.53)	43.54(−21.65)
Training	**Test**	**BuboQA**	**HR-BiLSTM**	**KBQA-Adpater**	**KEQA**
FBQ+WQ	**F917**	21.33(−19.59)	38.04(−18.16)	40.06(−19.31)	36.89 (−08.35)
	FBQ	37.63(−00.62)	30.07(+01.67)	27.88(−00.90)	29.12(+00.39)
	SQ	26.50(−48.31)	43.69(−28.61)	44.22(−27.79)	45.92(−29.43)
	WQ	62.17(+00.94)	48.66(−01.28)	47.73(−01.63)	64.96(−00.23)
Training	**Test**	**BuboQA**	**HR-BiLSTM**	**KBQA-Adpater**	**KEQA**
SQ+WQ	**F917**	40.06(−00.86)	54.76(−01.44)	57.64(−01.73)	47.84(+02.60)
	FBQ	21.65(−16.60)	20.03(−08.37)	18.74(−10.04)	13.89(−14.83)
	SQ	74.56(−00.25)	72.36(+00.06)	71.98(−00.03)	75.16(−00.19)
	WQ	58.44(−02.79)	46.57(−03.37)	44.94(−04.43)	61.00(−04.19)

Table A: Experimental results of BuboQA, HR-BiLSTM, KBQA-Adapter, and KEQA with the FQ+SQ, FQ+WQ, and SQ+WQ. The number in bracket means the difference of accuracy with the same training-test dataset experiments shown as Table 3. F917 is compared with the best model (best training data) for each system.

Appendix C The comparison between our implementations and original papers for QA systems

Dataset	Implementation	Accuracy	Dataset	Original	Accuracy
SQ	**BuboQA**	74.8	**SimpleQuestions**	**Mohammed et al. (2018)**	74.9
SQ	**HR-BiLSTM**	72.3	**SimpleQuestions**	**Yu et al. (2017)**	78.7
SQ	**KBQA-Adapter**	72.0	**SimpleQuestion-Balance**	**Wu et al. (2019)**	63.7
SQ	**KEQA**	75.4	**SimpleQuestions**	**Huang et al. (2019)**	75.4

Table B: Experimental results for our implementations and reported accuracies by original papers of QA systems. Note that SQ is not equal with SimpleQuestions, so this comparison can not be interpreted directly. Especially, we need to notice that SimpleQuestions-Balance (Wu et al., 2019) is designed for the zero-shot learning task; therefore it is the more difficult dataset than SimpleQuestions.