What do Models Learn from Question Answering Datasets?

Priyanka Sen
Amazon Alexa
Cambridge, UK
sepriyan@amazon.com

Amir Saffari
Amazon Alexa
Cambridge, UK
amsafari@amazon.com

Abstract

While models have reached superhuman performance on popular question answering (QA) datasets such as SQuAD, they have yet to outperform humans on the task of question answering itself. In this paper, we investigate what models are really learning from QA datasets by evaluating BERT-based models across five popular QA datasets. We evaluate models on their generalizability to out-of-domain examples, responses to missing or incorrect information in datasets, and ability to handle variations in questions. We find that no single dataset is robust to all of our experiments and identify shortcomings in both datasets and evaluation methods. Following our analysis, we make recommendations for building future QA datasets that better evaluate the task of question answering.

1 Introduction

Question answering (QA) through reading comprehension has seen considerable progress in recent years. This progress is in large part due to large-scale language models and the release of several new datasets that have introduced impossible questions (Rajpurkar et al., 2018), bigger scales (Kwiatkowski et al., 2019), and different angles, such as context (Choi et al., 2018; Reddy et al., 2019) and multi-hop reasoning (Welbl et al., 2018; Yang et al., 2018), to question answering.

At the time of writing this paper, models have outperformed human baselines on the widely-used SQuAD 1.1 and SQuAD 2.0 datasets, and datasets made to be more challenging, such as QuAC, have models 7 F1 points away from humans. Despite these increases in F1 scores, we are still far from saying question answering is a solved problem.

Concerns have been raised about how challenging QA datasets really are. Previous work has found that simple heuristics can give good performance on SQuAD (Weissenborn et al., 2017), and successful SQuAD models lack robustness by giving inconsistent answers (Ribeiro et al., 2019) or being vulnerable to adversarial attacks (Jia and Liang, 2017; Wallace et al., 2019).

If state-of-the-art models are excelling at test sets but not necessarily solving the underlying task of question answering, then our test sets are flawed. To make further progress in the field, we need to understand if models have learned the correct task and address issues with the way current datasets test model performance. In this work, we analyze QA datasets by asking three questions: (1) Does performance on individual datasets generalize to new datasets? (2) Are models learning reading comprehension for question answering?, and (3) How well do models handle question variations?

To answer these questions, we evaluate five QA datasets by fine-tuning BERT-based models and conducting six experiments. We find that (1) High performance on individual test sets does not generalize well outside of simple heuristics like word overlaps, (2) Removing or corrupting parts of the question or answer does not always harm model performance, showing that models can perform well without learning reading comprehension, and (3) No dataset fully prepares models to handle question variations like filler words or negation. Based on these findings, we make recommendations on how to create and evaluate new datasets that better test a models performance in question answering.

2 Datasets

We compare five datasets in our experiments: SQuAD 2.0, TriviaQA, Natural Questions, QuAC, and NewsQA. All of these datasets treat question answering as a reading comprehension task where the question is about a document and the answer is either extracted as a span of text or labeled unanswerable. To consistently compare models, we convert all datasets into a SQuAD 2.0 JSON format.[1] Since most datasets have a hidden test set, we evaluate models on their dev sets[2].

The following sections describe each dataset and any modifications we made to run our experiments. Table 1 shows a comparison of the datasets in terms of the average number of words in questions, contexts, and answers.

	Question	Context	Answer
SQuAD	10	120	3
TriviaQA	15	746	2
NQ	9	96	4
QuAC	7	395	14
NewsQA	8	709	4

Table 1: Comparison of question, context, and answer lengths by average number of words

SQuAD 2.0 (Rajpurkar et al., 2018) consists of 150K question-answer pairs on Wikipedia articles. To create SQuAD 1.1, crowd workers wrote questions about a Wikipedia paragraph and highlighted the answer (Rajpurkar et al., 2016). SQuAD 2.0 includes an additional 50K plausible but unanswerable questions.

TriviaQA (Joshi et al., 2017) includes 95K question-answer pairs from trivia websites. The questions were written by trivia enthusiasts and the evidence documents were retrieved by the authors retrospectively. We use the variant of TriviaQA where the documents are Wikipedia articles.

Natural Questions (NQ) (Kwiatkowski et al., 2019) consists of 300K questions from the Google search engine logs. For each question, a crowd worker highlighted a long and short answer, if possible, in a Wikipedia page. We use the subset of NQ with a long answer and frame the task as finding the short answer span in the long answer paragraph.

QuAC (Choi et al., 2018) contains 100K questions. To create QuAC, a student crowd worker asked questions about a Wikipedia article to a teacher crowd worker, who answered by selecting a text span. To standardize training, we do not model contextual information, but we include QuAC to see how models trained without context handle context-dependent questions.

[1]The formatted datasets and code to reproduce our experiments will be available on GitHub after double-blind review.

[2]As a result, we will refer to the dev sets as test sets for the remainder of this paper.

NewsQA (Trischler et al., 2017) is made up of 100K questions on 10K CNN articles. One set of crowd workers wrote questions based on a headline and summary, and a second set of workers found the answer in the article. To match SQuAD 2.0, we reintroduce unanswerable questions that were excluded during training in the original paper.

3 Model

Hyperparameter	Value
Batch Size	24
Learning Rate	3e-5
Epochs	2
Max Seq Length	384
Doc Stride	128

Table 2: Hyperparameter values for fine-tuning BERT

All models are initialized from a pre-trained BERT-Base uncased model[3]. For each dataset, we fine-tune a model on its training set using Devlin et al. (2019)'s default hyperparameters shown in Table 2. We evaluate each model with the SQuAD 2.0 evaluation script (Rajpurkar et al., 2018).

Dataset	Reference	Ours
SQuAD	76.3 (Liu et al., 2019)	75.6
TriviaQA	56.3 (Yang et al., 2019)	58.7
NQ	52.7 (Alberti et al., 2019)	73.5
QuAC	54.4 (Qu et al., 2019)	33.3
NewsQA	66.8 (Takahashi et al., 2019)	60.1

Table 3: Comparison to previously reported F1 scores

In Table 3, we provide a comparison between our models and previously published BERT-based model results. The significant differences are when we make modifications to match SQuAD. We simplified NQ by removing the long answer identification task and framed the short answer task in a SQuAD format, so we see higher results than the NQ BERT baseline. For QuAC, we stripped all context-related fields and treated each example as an independent question, so we see lower results than models built on the full dataset. For NewsQA, we reintroduced impossible questions to follow SQuAD 2.0, resulting in lower performance.

[3]`https://github.com/google-research/bert#pre-trained-models`

		Evaluated on					
		SQuAD	TriviaQA	NQ	QuAC	NewsQA	Avg Δ
Fine-tuned on	SQuAD	**75.6**	46.7	48.7	20.2	41.1	-17.2
	TriviaQA	49.8	**58.7**	42.1	20.4	10.5	-29.9
	NQ	53.5	46.3	**73.5**	21.6	24.7	-20.4
	QuAC	39.4	33.1	33.8	**33.3**	13.8	-36.9
	NewsQA	52.1	38.4	41.7	20.4	**60.1**	-22.2

Table 4: F1 scores of each fine-tuned model evaluated on each test set

We accept these drops in performance since we are interested in comparing changes to a baseline rather than achieving state-of-the-art results.

4 Experiments

In this section, we discuss the experiments run to evaluate how well QA datasets evaluate the task of question answering. All results are reported as F1 scores since they are correlated with Exact Match scores and are more forgiving to sometimes arbitrary cutoffs of answers (for example, we prefer to give some credit to a model for selecting Charles III even if the answer was King Charles III).

4.1 Does performance on individual datasets generalize to new datasets?

For our first experiment, we evaluate the generalizability of models on out-of-domain examples. While most work in QA has focused on evaluating a datasets own test set, generalizability is an important feature for models to learn. If we cannot get good, generalizable performance on research datasets, we will struggle to expand to the variety of questions a QA system can face in a real-world setting. Indeed several recent papers have focused on generalizability by evaluating transferability across datasets (Talmor and Berant, 2019; Yatskar, 2019), testing generalizability to out-of-domain data (Fisch et al., 2019), or building cross-dataset evaluation methods (Dua et al., 2019).

We test generalizability by fine-tuning models on each of the datasets and evaluating them against all five test sets without any further fine-tuning. The results for each model are reported as F1 scores in Table 4. The rows show a single models performance across all five datasets, and the columns show the performance of all the models on a single dataset. The model-on-self baseline is indicated in bold. The final column shows the average difference between each model's performance on a dataset and the model-on-self baseline.

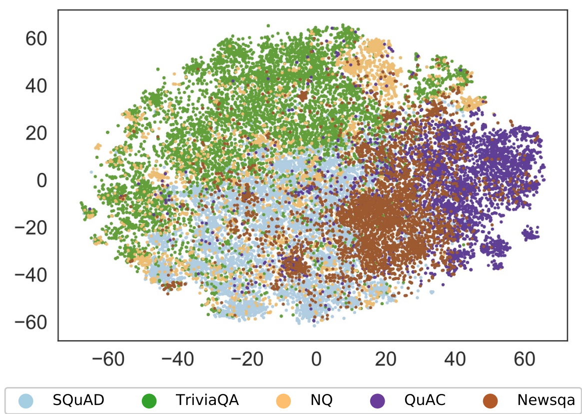

Figure 1: A t-SNE visualization of test set questions

All of the models take a considerable F1 drop when they are evaluated on a different dataset, so performance on an individual dataset does not generalize well across datasets. This finding confirms results found in previous work on different mixes of datasets (Talmor and Berant, 2019; Yogatama et al., 2019). However there is variation in how the models perform, both in terms of F1 drop across datasets, and performance on a single dataset across models. We investigate these differences further by looking into test set similarity and test set difficulty.

4.1.1 Test Set Similarity

The SQuAD model has the lowest average F1 drop across all datasets with a delta of -17.2, while QuAC has the highest with -36.9. This suggests that the SQuAD model is better prepared to answer questions from out-of-domain datasets. We hypothesize this is because of test set similarity. If two test sets are similar, a model that is successful on one will likely be successful on the other. As test sets decrease in similarity, we can expect model generalizability to deteriorate.

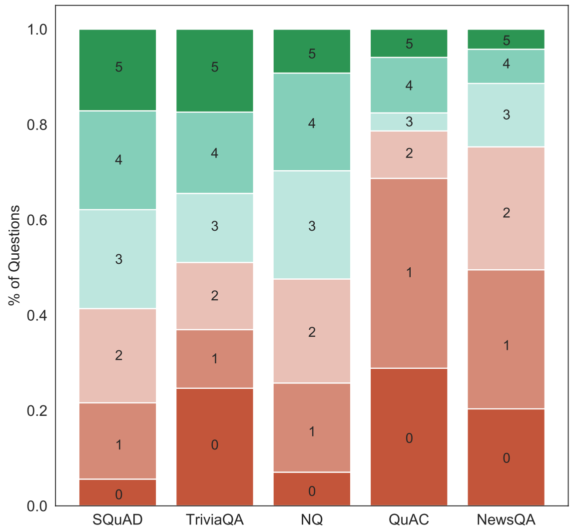

Figure 2: A bar graph of how many questions in each dataset are answered by 0, 1, 2, 3, 4, or 5 models

To visualize similarity, we map the questions from the test sets to a vector space. We calculate fixed length vectors for each question using bert-as-a-service[4] with a pretrained BERT-Base model as the sentence encoder and extracting from the second-to-last layer. We get 768-dimension vectors for each question and use t-SNE (Maaten and Hinton, 2008) to produce a lower dimensional graph.

Figure 1 is a t-SNE visualization of the test set questions from all five datasets. Each point represents a question, and the five colors represent the five datasets. SQuAD and NQ overlap with most of the datasets, which can explain why the SQuAD and NQ models have lower F1 losses. QuAC and NewsQA are further away from the other distributions and have worse generalizability. TriviaQA and NewsQA, for example, have very little overlap and TriviaQA only achieves a 10.5 F1 score on NewsQAs test set.

Overall, this visualization supports the hypothesis that good generalization across datasets can be explained by test set similarity.

4.1.2 Test Set Difficulty

As seen in Table 4, models score up to 53.5 F1 points on SQuAD without seeing SQuAD examples before, while models do not score above 21.6 F1 points on QuAC without seeing QuAC examples. This suggests some test sets are easier than others. To quantify this, we calculate what proportion of each test set can be correctly answered by

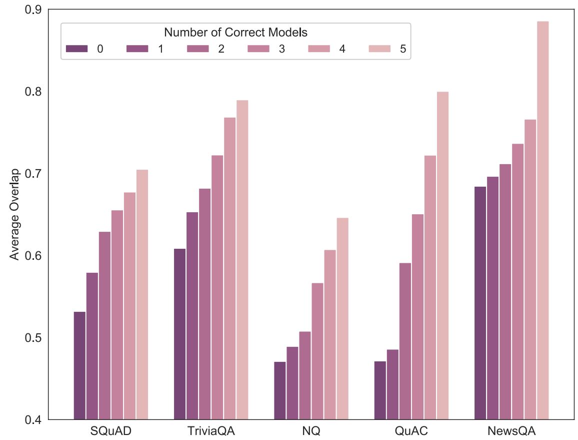

Figure 3: More models correctly answer answerable questions if they have higher question-context overlap.

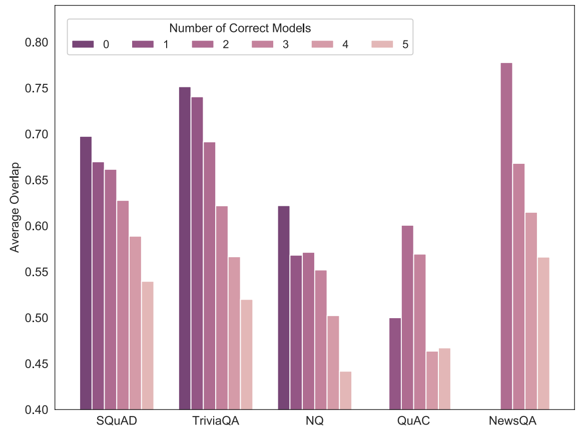

Figure 4: More models correctly answer impossible questions if they have lower question-context overlap. NewsQA has four bars since all impossible NewsQA questions were correctly answered by at least 1 model.

how many models. This data is represented as a bar graph in Figure 2. Each bar represents one dataset, and the segments show how much of the test set is answered correctly by 0 to 5 of the models.

We consider questions easier if more models correctly answer them. The figure shows that QuAC and NewsQA are more challenging test sets and contain a higher proportion of questions that are answered by 0 or 1 model. In contrast, more than half of SQuAD and NQ and almost half of TriviaQA can be answered correctly by 3 or more models.

While difficult questions pose a challenge for QA models, too many easy questions inflate a models performance. What makes a question easy? We identified a trend between difficulty of a question and the overlap between the question and the context. We measured overlap as the number of

[4]`https://github.com/hanxiao/bert-as-service`

Experiment	Question	Answer Text	Answer Start
Original	Who was the Norse leader	Rollo	308
Random Label	Who was the Norse leader	succeeding	721
Shuffled Context	Who was the Norse leader	Rollo	480
Incomplete (first half)	Who was	Rollo	308
Incomplete (first word)	Who	Rollo	308
Filler word	Who really was the Norse leader	Rollo	308
Negation	Who wasn't the Norse leader		-1

Table 5: Examples of how question-answer pairs were modified in each experiment

words that appeared in both the question and the context divided by the number of words in the question. For answerable questions, Figure 3 shows that more models return correct answers when there is higher average overlap. For impossible questions, Figure 4 shows that fewer models return correct answers when there is higher average overlap. This suggests that the models are good at identifying answers when the answer appears in a context sentence that is similar to the question. In fact, they can over-rely on this strategy and return answers to impossible questions when there is high question-context overlap even when no answer exists.

These results show that identifying overlap is a method models are able to exploit even when the question is out-of-domain. Reducing the number of high overlap questions in a dataset can create more challenging datasets and better test more complex strategies for reading comprehension.

4.2 Are models learning reading comprehension for question answering?

State-of-the-art models get good performance on QA datasets. But does good performance mean that models are learning reading comprehension? Or are they able to cheat and take shortcuts to arrive at the same answers? We explore this by performing three dataset ablation experiments with random labels, shuffled contexts, and incomplete questions. If models are able to pass test sets even with incorrect or missing information, then the models are likely not learning the task of reading comprehension. The three experiments and their results are discussed in the next sections.

4.2.1 Random Labels

A robust model should withstand some amount of noise at training time to offset annotation error. However if a model can perform well even with a high level of noise, we should be wary of what the model has learned. In our first dataset ablation experiment, we evaluated how various amounts of noise at training time affected model performance.

		% of Random Labels		
Dataset	Baseline	10%	50%	90%
SQuAD	78.5	77.1	73.9	32.1
TriviaQA	46.8	36.6	10.9	0.0
NQ	70.6	68.1	60.5	19.4
QuAC	20.3	16.4	1.8	0.3
NewsQA	56.3	50.8	30.2	2.0

Table 6: F1 scores of answered questions decrease as models are fine-tuned on increasingly noisy data.

To introduce noise to the training sets, we randomly selected 10%, 50%, or 90% of the training examples that were answerable and updated the answer to a random string from the same context and of the same length as the original answer. We ensured that the random answer contained no overlaps with the original answer. For simplicity, we did not alter impossible examples. An example of a random label is in the second row of Table 5.

We fine-tuned new models on increasingly noisy training sets and evaluated them on the original clean test sets. The results are in Table 6 in terms of F1 scores and reported only for answerable questions. On training sets where 10% of the examples have random labels, all of the models see a drop in F1 scores. SQuAD, NQ, and NewsQA achieve over 90% of their baseline score, showing robustness to a reasonable level of noise. TriviaQA and QuAC take larger F1 hits (achieving only 78% and 81% of their baselines), suggesting that they are less robust to this type of noise.

As the amount of noise increases, most F1 scores drop to nearly 0. SQuAD and NQ, however, are suspiciously robust even when 90% of their training examples are random. SQuAD achieves 41% of its baseline and NQ achieves 27% of its baseline with training sets that are 90% noise. Although the SQuAD and NQ models achieve high F1 scores on their test sets, this shows that parts of the test sets are answerable without needing to learn reading comprehension from correct examples in the training set.

4.2.2 Shuffled Context

Dataset	Baseline	Shuffled Context
SQuAD	75.6	70.5
TriviaQA	58.7	38.7
NQ	73.5	64.5
QuAC	33.3	32.4
NewsQA	60.1	48.2

Table 7: F1 scores decrease, but not dramatically, on test sets with shuffled context sentences.

Written text usually follows a logical structure, so we would expect reading comprehension tasks to rely on a structure where facts are presented in a meaningful order. Do models exploit this structure, or do they perform just as well with randomly arranged sentences? Our second dataset ablation experiment investigates how models perform when the sentences in the context are shuffled.

For each context paragraph in the test set, we split the context by sentence, randomly shuffled the sentences, and rejoined the sentences into a new paragraph. The original answer text remained the same, but the answer start token was updated by locating the correct answer text in the shuffled context. An example is in the third row of Table 5.

We used our models fine-tuned on the original training sets and evaluated on the test sets with shuffled contexts. The results are in Table 7. TriviaQA sees the largest drop in performance, achieving only 66% of its baseline, followed by NewsQA with 80% of its baseline. SQuAD and QuAC, on the other hand, get over 93% of their original baselines even with randomly shuffled contexts. TriviaQA and NewsQA have longer contexts, with an average of over 700 words, and so shuffling longer contexts seems more detrimental. However these results show that for many questions, models do not need to learn content structure to correctly predict the answer, and sentence position does not seem to play much of a role.

4.2.3 Incomplete Input

Dataset	Baseline	First Half	First Word	NER
SQuAD	75.6	36.4	22.8	30.0
TriviaQA	58.7	45.8	31.8	25.2
NQ	73.5	61.4	50.3	35.9
QuAC	33.3	25.2	22.4	17.2
NewsQA	60.1	43.6	26.3	11.3

Table 8: F1 scores decrease on test sets with incomplete input, but models can work with as little as one word.

QA dataset creators and their crowd workers spend considerable effort hand-crafting questions that are meant to challenge a models ability to understand language. But are models using the questions? In previous work, Agrawal et al. (2016) found that a Visual Question Answering (VQA) model could get good performance on the test set with just half the original question. We applied Agrawal et al. (2016)'s approach of using incomplete questions to our datasets.

We created two variants of each test set: one containing only the first half of each question, and one containing only the first word of each question. The answer expectations were not changed. Examples are in the fourth and fifth rows of Table 5.

We evaluated models fine-tuned on the original training set on the incomplete test sets. The results are in the First Half and First Word columns of Table 8. F1 scores decrease on test sets with increasingly incomplete input, but surprisingly, all of the models can correctly predict examples with as little as the first word. In the most extreme case, NQ achieves 68% of its baseline F1 score with only the first word. These results show that not all questions in test sets require full question understanding for the model to make correct predictions.

To test how this is possible, we create a naive named entity recognition (NER) baseline using spaCy[5] to see how biased the datasets are at selecting the first entity of a given type. If the sentence started with who, we returned the first person entity in the context, for when, we returned the first date, for where, the first location, and for what, the first organization, event, or work of art. The results

[5] `https://spacy.io`

are reported in the NER column of Table 4. With the exception of NewsQA, we are able to achieve over 40% of the baseline performance on all other datasets with an NER system that does not do any reading comprehension.

4.3 How well do models handle question variations?

The previous section found that models can perform well on test sets even as seemingly important features for question answering are stripped from datasets. This section considers the opposite problem: Can models remain robust as features are added to datasets? To analyze this, we run two experiments where we add filler words and negation to test set questions.

4.3.1 Filler Words

Dataset	Baseline	Filler Words
SQuAD	75.6	69.5
TriviaQA	58.7	56.5
NQ	73.5	67.6
QuAC	33.3	31.2
NewsQA	60.1	54.8

Table 9: F1 scores slightly decrease on test sets where a filler word is added to the question.

If a QA model is understanding questions, it should handle semantically equivalent questions equally well. While previous works have shown poor performance on QA datasets with paraphrased questions (Ribeiro et al., 2018; Gan and Ng, 2019), we take an even simpler approach of introducing filler words that do not affect the rest of the question and test the robustness of the models.

For each question in the test set, we randomly added one of three filler words (*really*, *definitely*, or *actually*) before the main verb, as identified by spaCy's POS tagger. An example is shown in the sixth row of Table 5. The answer expectations were not changed.

Table 9 shows the results of models fine-tuned on their original training sets and evaluated on their filler word test sets. All models drop between 2 to 5 F1 points. Although these drops do not seem like much, these results show that even such a naive approach can hurt performance. It is no surprise that more sophisticated paraphrases of questions cause models to fail. The SQuAD model in particular had better performance when 50% of the training set was randomly labeled (73.9) than when filler words were added to the test set (69.5), suggesting that some models have learned to become robust to less consequential features.

4.3.2 Negation

Dataset	Baseline	Negative	Negative on Original
SQuAD	75.6	97.9	2.0
TriviaQA	58.7	55.5	42.0
NQ	73.5	40.0	68.9
QuAC	33.3	70.7	16.1
NewsQA	60.1	24.3	52.3

Table 10: SQuAD outperforms other models on test sets with negative questions.

Negation is an important grammatical construction for QA systems to understand. While negative questions are less common than positive questions, they can be more damaging. After all, giving the same answer to a question and its negative (Who invented the telescope? vs. Who didnt invent the telescope?) can frustrate or mislead users. We evaluated how sensitive models are to negation in questions. In particular, we tested if models understand negation, or if they skip negative words and continue to provide the original answer.

We negated every question in the test set by mapping common verbs (i.e. *is*, *did*, *has*) to their contracted negative form (i.e. *isn't*, *didn't*, *hasn't*) or by inserting *never* before the main verb of the sentence, as identified by spaCys POS tagger. We used *never* for the sake of simplicity, since *not* often requires an auxiliary and a verb tense change in English. The expected answer of all the negative questions were changed to be impossible. An example is in the last row of Table 5.

We used the models fine-tuned on their original training sets and evaluated them on the negated test sets. The results are in Table 10. The Baseline column shows the results of the model on the original test set. The Negative column shows the results of the model on the negative test set expecting no answer as the correct prediction. The Negative on Original column shows how often the model returns the original answer ignoring the negation. We see that SQuAD outperforms all the other models in both correctly not answering negative questions and giving its original answer when given a negative question less than 3% of the time. Other models return the original answer to the negative question between 48% and 94% of the time.

Dataset	*n't*	*never*	% of impossible
SQuAD	0.85	0.89	0.05
TriviaQA	0.31	0.48	0.004
NQ	0.37	0.34	0.009
QuAC	0.17	0.17	0.002
NewsQA	0.14	0.06	0.009

Table 11: The fraction of questions in the training set including *n't* or *never* that are impossible. The final column shows of impossible questions, how many include *n't* or *never*

Does the SQuAD model understand negation, or is this a sign of bias? The first two columns in Table 11 show how often a question containing *nt* or *never* was impossible in the training set. SQuAD has a high bias, with 85% of questions containing *n't* and 89% of questions containing *never* being impossible. The final column in Table 11 shows that 5% of impossible questions in SQuAD contain *n't* or *never*, which is a higher proportion than other datasets. SQuADs impressive performance then can be attributed to a bias in the dataset. These results find that no dataset adequately prepares a model to understand negation.

5 Related Work

Our work is inspired by recent trends in NLP to evaluate generalizability and probe what a model has learned. In terms of generalizability, prior work has been done by Yogatama et al. (2019) who evaluated a SQuAD 1.1 model across four datasets, including TriviaQA and QuAC. Talmor and Berant (2019) performed a more comprehensive test across ten QA datasets, including SQuAD 1.1, TriviaQA, and NewsQA. And most recently, the MRQA 2019 shared task (Fisch et al., 2019) used different datasets for training (including SQuAD 1.1, TriviaQA, NQ, and NewsQA), development, and testing to evaluate transferability. In our work, we extend the work on generalizability by including impossible questions and more closely analyzing the related topics of test set similarity and difficulty.

Across different fields in NLP, previous work on probing what a model has learned has found that state-of-the-art models can get good performance on incomplete input (Agrawal et al., 2016; Niven and Kao, 2019), under-rely on important words, (Mudrakarta et al., 2018), and over-rely on simple heuristics (McCoy et al., 2019). In question answering, researchers have found that SQuAD is vulnerable to adversarial attacks (Jia and Liang, 2017; Wallace et al., 2019) and is not robust to paraphrases (Ribeiro et al., 2018; Gan and Ng, 2019). Our work continues exploring what a QA model has learned by comprehensively testing multiple QA datasets against a variety of attacks.

6 Conclusions

In this work, we compared five QA datasets across six tasks and found that even with high F1 scores, many models failed to learn generalizability, expected responses to incorrect or missing data, or the ability to handle variations. These findings reveal shortcomings in both the datasets and current evaluation methods. Based on our work, we make the following recommendations to researchers who create or evaluate QA datasets:

- **Test for generalizability**: Models are more valuable to real-world applications if they generalize. When releasing a new QA model, report performance across all relevant QA datasets without further fine-tuning.
- **Challenge the models**: Evaluating on too many easy questions can inflate our understanding of what a model has learned. Calculate and discard questions that can be solved trivially by high overlap or extracting the first named entity.
- **Be wary of cheating**: Good performance does not mean good understanding. Probe datasets by adding noise, shuffling contexts, or providing incomplete input to ensure models arent taking shortcuts.
- **Include variations**: Language is infinite, so we should prepare models to handle a variety of questions. Consider adding variations such as filler words or negation to existing questions to evaluate how well models have understood a question.
- **Standardize dataset formats**: When creating new datasets, consider following a standardized format, such as SQuAD, to make cross-dataset evaluations simpler.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Dheeru Dua, Ananth Gottumukkala, Alon Talmor, Sameer Singh, and Matt Gardner. 2019. Orb: An open reading benchmark for comprehensive evaluation of machine reading comprehension. In *EMNLP 2019 MRQA Workshop*, page 147.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *EMNLP 2019 MRQA Workshop*, page 1.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel 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 (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle pdftein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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. Association for Computational Linguistics.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. Bert with history answer embedding for conversational question answering. *arXiv preprint arXiv:1905.05412*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Takumi Takahashi, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. 2019. CLER: Cross-task learning with expert representation to generalize reading and understanding. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 183–190, Hong Kong, China. 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. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. Data augmentation for bert fine-tuning in open-domain question answering. *arXiv preprint arXiv:1904.06652*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Mark Yatskar. 2019. A qualitative comparison of CoQA, SQuAD 2.0 and QuAC. 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 2318–2323, Minneapolis, Minnesota. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.