Improving the Robustness of Question Answering Systems to Question Paraphrasing

Wee Chung Gan
Department of Computer Science
National University of Singapore
gan_weechung@u.nus.edu

Hwee Tou Ng
Department of Computer Science
National University of Singapore
nght@comp.nus.edu.sg

Abstract

Despite the advancement of question answering (QA) systems and rapid improvements on held-out test sets, their generalizability is a topic of concern. We explore the robustness of QA models to question paraphrasing by creating two test sets consisting of paraphrased SQuAD questions. Paraphrased questions from the first test set are very similar to the original questions designed to test QA models' over-sensitivity, while questions from the second test set are paraphrased using context words near an incorrect answer candidate in an attempt to confuse QA models. We show that both paraphrased test sets lead to significant decrease in performance on multiple state-of-the-art QA models. Using a neural paraphrasing model trained to generate multiple paraphrased questions for a given source question and a set of paraphrase suggestions, we propose a data augmentation approach that requires no human intervention to re-train the models for improved robustness to question paraphrasing.

1 Introduction

With the release of large-scale, high-quality, and increasingly challenging question answering (QA) datasets (Rajpurkar et al., 2016; Nguyen et al., 2016; Joshi et al., 2017; Reddy et al., 2018), the research community has made rapid progress on QA systems. On the popular SQuAD dataset (Rajpurkar et al., 2016), top QA models have achieved higher evaluation scores compared to human. However, since the test set is typically a randomly selected subset of the whole set of data collected, and thus follows the same distribution as the training and development sets, the performance of models on the test set tends to overestimate the models' ability to generalize to other unseen test data. It is thus important for QA models to be evaluated on other unseen test data for a better indication of their generalization ability.

Context: ... commentators had debated whether the figure could be reached as the growth in subscriber numbers elsewhere in Europe flattened.
Original Question: What was happening to subscriber numbers in other areas of Europe? **Prediction:** flattened
Paraphrased Question: What was going on with subscriber numbers in other areas of Europe? **Prediction:** growth
Context: ... According to the Second law of thermodynamics, nonconservative forces necessarily result in energy transformations within closed systems from ordered to more random conditions as entropy increases.
Original Question: What is the law of thermodynamics associated with closed system heat exchange? **Prediction:** Second law of thermodynamics
Paraphrased Question: What is the law of thermodynamics related to closed system heat exchange? **Prediction:** nonconservative forces

Figure 1: Examples of brittleness to paraphrasing. Both examples show an initially correct prediction turning into a wrong prediction after small changes in the question.

In this paper, we explore QA models' robustness to question paraphrasing. Our motivation stems from the observation that when a question is phrased in a slightly different but semantically similar way, QA models can output a wrong prediction despite being able to answer the original question correctly. Figure 1 shows two such examples. Sensitivity to such paraphrasing needs to

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6065–6075
Florence, Italy, July 28 - August 2, 2019.

be improved for better reliability of QA models on unseen test questions.

We focus on the SQuAD QA task in this paper. SQuAD was created by getting crowd workers to create questions and answers from Wikipedia paragraphs. SQuAD serves as a benchmark for QA systems, taking as input a question and a context to predict the correct answer. Two evaluation metrics are used: exact match (EM) and F1. Since an answer must be a span from the context, most models output a probability distribution for the start and end token separately, and constrain the end token to be after the start token.

Despite the availability of SQuAD 2.0 (Rajpurkar et al., 2018) which requires models to additionally decide whether a question is unanswerable, we focus on the original version of SQuAD (Rajpurkar et al., 2016). This is due to the simpler task of the original SQuAD which allows us to concentrate on robustness of models to question paraphrasing.

We created two paraphrased test sets by paraphrasing SQuAD questions so as to evaluate the robustness of models to question paraphrasing. Using a neural paraphrasing model trained to generate a paraphrased question given a source question and a paraphrase suggestion, we created a non-adversarial paraphrased test set from SQuAD development questions which is subsequently verified by human annotators. We also created an adversarial paraphrased test set by re-writing the original question using words in the context near a confusing answer candidate of the same type as the correct answer. Both test sets lead to significant decrease in the performance of QA models.

We hypothesize that exposing a model to various ways of asking the same question during training will improve its robustness to question paraphrasing. To this end, we use the trained paraphrasing model to introduce additional training examples containing paraphrased training questions to augment the original training data for retraining.

The contributions of this paper are as follows:

- We introduce a *novel* method to generate diverse paraphrased questions by guiding the model with paraphrase suggestions.
- We release two paraphrased test sets[1] using SQuAD development questions for evaluation of QA models' robustness to question paraphrasing. The non-adversarial paraphrased test set consists of 1,062 questions paraphrased with slight perturbations from the original questions. The adversarial paraphrased test set consists of 56 questions paraphrased using context words near a confusing answer candidate.
- We show that all three state-of-the-art QA models that we experimented with, including one that outperforms human on SQuAD, have worse performance on the non-adversarial paraphrased test set even though they are semantically and syntactically similar to the original questions. All three QA models have drastically lower performance on the adversarial paraphrased test set.
- We show that it is possible to improve the robustness of QA models to paraphrased questions for both paraphrased test sets, using a fully automatic approach to augment the training set and retraining the model on the augmented training set.

[1]The two test sets are available at https://github.com/nusnlp/paraphrasing-squad.

2 Paraphrase-Guided Paraphrasing Network

In this section, we introduce our method to train a neural network that is able to take as input a source question together with a paraphrase suggestion (a word or phrase) to generate a paraphrased question. To do so, we require a training dataset where each training example is of the form (source question, paraphrase suggestion, target question). Since we want the generated paraphrase to contain the paraphrase suggestion provided, the suggestion given during training must be part of the target question. We elaborate on the construction of our training dataset in Section 2.2.

By training our model to make use of a paraphrase suggestion to paraphrase a source question, we are able to leverage a database of word and phrasal paraphrases (Section 3.1.1) to generate multiple paraphrases for a given SQuAD question. This is useful for the creation of the non-adversarial paraphrased test set (Section 3.1) and additional training data for improvement on this test set (Section 4.2.1). This model is also useful for training data augmentation for improvement on the adversarial paraphrased test set (Section 4.2.2).

2.1 Model Architecture

We use the transformer model from Vaswani et al. (2017) which is an encoder-decoder architecture that relies mainly on a self-attention mechanism. We extend the decoder using the copy mechanism of See et al. (2017) which allows tokens to be copied from the source question. This is achieved by augmenting the probability distribution of the output vocabulary to include tokens from the source question.

The input to the encoder is the concatenation of a paraphrase suggestion and the source question separated by a special token: "<suggestion> <sep> <source question>", tokenized using the subword tokenizer SentencePiece by Kudo and Richardson (2018).

2.2 Dataset Preparation

We use a combination of the WikiAnswers paraphrase corpus (Fader et al., 2013) and the Quora Question Pairs dataset[2] for training. The two questions in a question pair in the Quora dataset are typically very similar in meaning. In contrast, the WikiAnswers paraphrase corpus tends to be noisier but one source question is paired with multiple target questions. This allows the model to be trained to output different target questions depending on the paraphrase suggestion given. A combination of these two datasets thus provides a balance between good paraphrasing and using a paraphrase suggestion to generate a paraphrase.

2.2.1 Obtaining Source and Target Questions

WikiAnswers dataset: This paraphrase corpus contains over 22 million question pairs. We use only a small portion of this dataset so as not to overwhelm the Quora dataset. We only keep a question pair if each question is at least 7 tokens long, since training on longer sentences is more helpful. We also attempt to filter out erroneous question pairs by removing all question pairs with paraphrase similarity scores below 0.7 using a pre-trained model by Wieting and Gimpel (2018). Then, we randomly sample source questions to obtain about 350,000 question pairs.

Quora dataset: For the Quora dataset, we use a pair of questions as two training examples by including both source question to target question and vice versa in the training set, i.e., we include QuestionA → QuestionB and QuestionB → QuestionA in the training set. A total of about 280,000 training examples come from the Quora dataset.

[2]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

2.2.2 Obtaining Paraphrase Suggestions

WikiAnswers dataset: For each source and target question pair, we use word alignments that come with the dataset to match words and phrases from the source to target question to obtain phrase alignment pairs. The alignment pairs are filtered to keep phrases that occur in the target question but are not in the source question. Given a source and target question pair, we thus have a set of possible paraphrase suggestions to choose from. We show an example in Figure 2.

Since most source questions have multiple target questions in this dataset, given one source question q and all of its corresponding target questions $t_1, t_2, ..., t_k$, we thus have k sets of possible paraphrase suggestions $S_1, S_2, ..., S_k$. From each set of possible paraphrase suggestions S_i, we select one suggestion $s_i \in S_i$ to construct a training example (q, s_i, t_i). We constrain the selection such that all paraphrase suggestions chosen are unique, i.e., $\forall i, j (i \neq j \Rightarrow s_i \neq s_j)$. This is to ensure that there are no duplicate (q, s_i) input pairs in the training dataset which will result in the model being trained on different targets given the same input.

Furthermore, to enable the model to paraphrase even without a suggestion given, some paraphrase suggestions are randomly selected to be replaced with a special empty token.

Quora dataset: Since the Quora dataset does not come with word alignments, we first use TextRank (Mihalcea and Tarau, 2004) to obtain question keywords from both source and target questions. Then, the paraphrase suggestion is the highest ranked key phrase in the target question that is not in the source question. We do not allow stopwords to be selected as a paraphrase suggestion. Similarly, a random subset of the paraphrase suggestions is replaced with the special empty token. We show an example of obtaining paraphrase suggestions for this dataset in Figure 3.

2.3 Implementation

We train our paraphrasing model using the implementation by OpenNMT (Klein et al., 2018), following the hyper-parameters of Vaswani et al. (2017). We lowercase all data for training and create a tokenized vocabulary of size 8k from SentencePiece (Kudo and Richardson, 2018).

Question	Word Alignments	Phrase Alignments	Candidate Suggestions
Source	what nutrients do green peppers have in them ?	(what, what) (green, a green) (have in them, contain) ...	a green, pepper, contain
Target	what nutrients does a green pepper contain ?		

Figure 2: An example of finding possible paraphrase suggestions for a source and target question pair from the WikiAnswers dataset. Since there can be multiple target questions for a given source question, we ensure that there are no duplicates in the suggestions chosen for the same source question.

	Question	Keywords	Candidate Suggestions	Selected Suggestion
Source	how can i find out how many devices are connected to my wifi?	wifi, connected, many devices, devices, find	wifi network, network, know	wifi network
Target	how can i know how many devices are connected to my wifi network?	wifi network, network, wifi, connected, many devices, devices, know		

Figure 3: An example of obtaining a paraphrase suggestion for a source and target question pair from the Quora dataset. Keywords from the questions are obtained from TextRank.

Since our model is not directly comparable to other neural paraphrasing models in the literature, we do not perform automatic evaluation and instead leave the evaluation of our model's performance to Section 3.1.2, where we employ human annotators to evaluate the paraphrasing quality of our model on SQuAD questions.

3 Paraphrasing SQuAD Questions

In this section, we discuss the creation of two paraphrased test sets using SQuAD development questions for the evaluation of the robustness of QA models to question paraphrasing.

3.1 Non-Adversarial Paraphrased Test Set

We use the trained paraphrasing model from Section 2 to create a non-adversarial paraphrased test set. We employ human annotators to ensure the quality of the questions for this test set, which also serves as evaluation for our paraphrasing model.

In contrast to methods that query the model to create adversarial examples, this dataset is created in a completely model-independent way designed to provide a better indication on performance during actual use.

3.1.1 Paraphrasing Process

To obtain paraphrase suggestions for input to our paraphrasing model to paraphrase SQuAD questions, we rely on the paraphrase database PPDB (Pavlick et al., 2015), which is an automatically extracted database consisting of millions of paraphrase pairs. The paraphrase pairs can contain a single word or multiple words. PPDB comes in 6 different sizes, with larger sizes having greater coverage but are less accurate.

First, we obtain all n-grams (up to 6-grams) from the source question and remove unigrams that are stopwords. Next, we search the PPDB (XL size) for paraphrases of the remaining n-grams with equivalence score above 0.25. This gives us a set of paraphrase suggestions for the model to generate paraphrased questions. We use a threshold of 0.25 for a balance between having a larger set of paraphrase suggestions and having a less noisy set of suggestions.

After paraphrase generation, we perform post-processing to remove semantically dissimilar paraphrases. Similar to filtering question pairs from the WikiAnswers corpus, we use the pre-trained model by Wieting and Gimpel (2018) to obtain paraphrase similarity score for the generated questions and keep only those scoring above 0.95. This is required due to noisiness of the paraphrase suggestions obtained from PPDB and to ensure that a larger number of paraphrased questions are semantically similar to the original question.

We summarise the paraphrasing process in Fig-

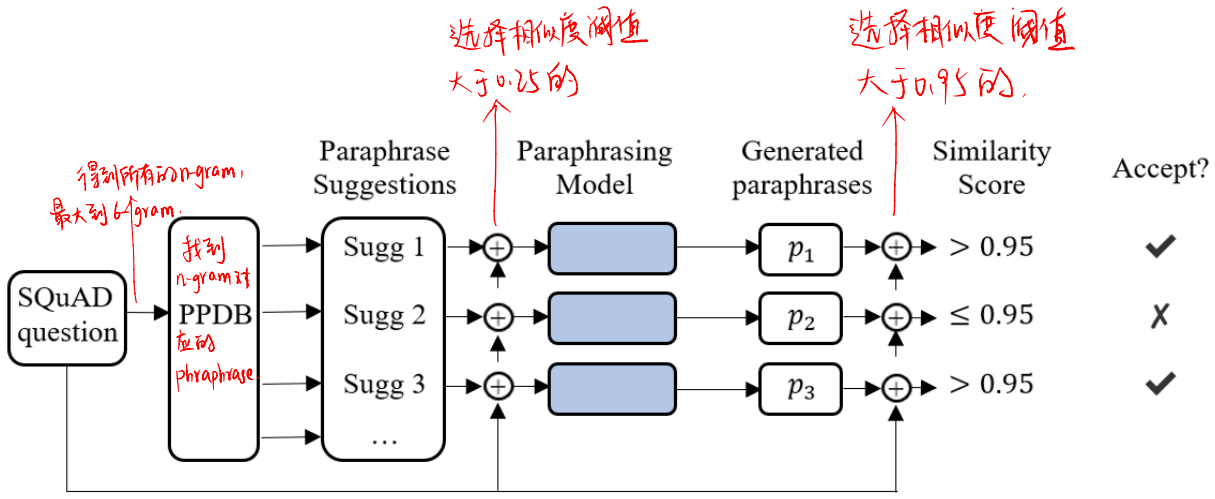

Figure 4: Process to paraphrase SQuAD questions. We first use PPDB to obtain paraphrase suggestions before passing both the original question and the suggestions to our paraphrasing model to generate paraphrases. A generated paraphrase is accepted if its similarity score with the original question is above 0.95. ⊕ refers to the use of the original SQuAD question and the previous output as inputs to the next step.

Original Question
the european court of justice cannot uphold measures that are incompatible with what?
Paraphrased Questions
1. the european court of justice cannot uphold a number of measures that are incompatible with what?
2. the european court of justice cannot uphold measures that are inconsistent with what?
3. the european court of justice cannot uphold measures which are not compatible with what?
4. the european court of justice has not been able to uphold measures that are incompatible with what?

Figure 5: Examples of generated paraphrases.

ure 4 and show four example paraphrases generated by our model from the same question in Figure 5.

3.1.2 Human Evaluation

To evaluate the quality of the automatically generated paraphrases, we employ human annotators from Amazon Mechanical Turk (AMT) to rate the semantic equivalence and fluency of the paraphrased questions.

We paraphrase questions from the SQuAD development set and randomly select 3,000 generated paraphrases, containing between 2 and 3 paraphrased questions for each original question. For each pair of questions, we ask 2 annotators from AMT to state how well they agree with the following two statements, on a scale of one to five (strongly disagree, disagree, neutral, agree, or strongly agree):

1. The paraphrased question has the same meaning as the original question (i.e., both the paraphrased and the original question are expected to yield the same answer).

2. The paraphrased question is written in fluent English.

For better annotation quality, we employ two annotators to annotate each paraphrased question and require the annotators to have at least 99% approval rate with at least 1,000 approved HITs.

The evaluation results are shown in Figures 6 and 7, where we plot the number of annotations against the scores assigned by the annotators, which are between 1 (Strongly Disagree) to 5 (Strongly Agree). 78.1% of the generated paraphrases are judged to be semantically equivalent and 78.6% are judged to be fluent, where annotators agree or strongly agree to questions 1 and 2 respectively.

3.1.3 Test Set Creation

We only include a generated paraphrased question into the test set if both annotators agree or strongly agree that the paraphrased question and the original question are semantically equivalent. To ensure that no question is over-represented, if there are multiple accepted paraphrased questions from an original question, we randomly select only one of the paraphrased questions to be included in the test set. A total of 1,062 paraphrased questions are produced.

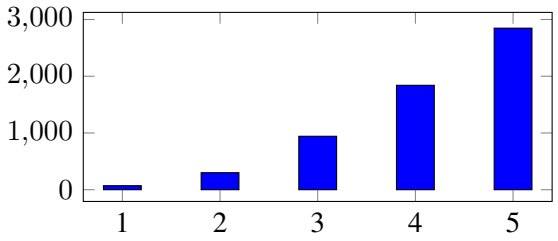

Figure 6: Semantic equivalence ratings

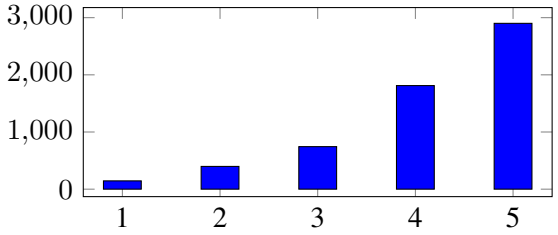

Figure 7: Fluency ratings

3.2 Adversarial Paraphrased Test Set

Motivated by the observation that QA models trained on SQuAD tend to perform string matching to return an answer of an appropriate type near a region of significant word overlap between the context and the question (Jia and Liang, 2017; Rondeau and Hazen, 2018), we create a test set to exploit this weakness of the models. In the context of question paraphrasing, we can simply paraphrase the question by using words in the context near a wrong answer candidate of the same type to generate a natural adversarial example.

We show in Figure 8 an example of producing such a paraphrased question. Since the correct answer "2009" is a year, we locate another year "1963" in the context and use the nearby context words "been televised" to paraphrase the original question.

We perform such paraphrasing manually by going through question and context pairs from the SQuAD development set and re-writing the question using context words near a confusing answer candidate if such a candidate exists and there are suitable nearby context words for use in paraphrasing. We create a total of 56 paraphrased questions for the adversarial test set.

Context: 826 Doctor Who instalments have been televised since 1963 ... Starting with the 2009 special "Planet of the Dead", the series was filmed in 1080i for HDTV ...
Original Question: In what year did Doctor Who begin being shown in HDTV? **Prediction:** 2009
Paraphrased Question: Since what year has Doctor Who been televised in HDTV? **Prediction:** 1963

Figure 8: An example of paraphrasing question using context words (underlined) near a confusing answer candidate to generate a natural adversarial example.

4 Experiments on QA Models

We conduct experiments on three state-of-the-art QA models: BERT (Devlin et al., 2018)[3], DrQA[4] (Chen et al., 2017), and BiDAF[5] (Seo et al., 2016). BERT, in particular, outperforms human on the SQuAD task.

4.1 Evaluating Performance on the Two Paraphrased Test Sets

For each paraphrased test set, we compare the performance of the three QA models on the original questions from the SQuAD development set and the corresponding paraphrased questions.

4.1.1 Non-Adversarial Paraphrased Test Set

The performance of the QA models on the original and paraphrased questions for the non-adversarial paraphrased test set is given in Table 1.

Despite the paraphrased set being semantically similar, and no model querying is performed to intentionally locate weaknesses of the QA models, all three models suffer a significant drop in performance. This highlights the brittleness of the trained models to question paraphrasing.

4.1.2 Adversarial Paraphrased Test Set

We compare the performance of QA models on the original and paraphrased questions for the adversarial paraphrased test set in Table 2.

[3]We used the PyTorch re-implementation available at https://github.com/huggingface/pytorch-pretrained-BERT

[4]We used the re-implementation focusing on the reader module available at https://github.com/hitvoice/DrQA

[5]We used the original implementation available at https://github.com/allenai/bi-att-flow

Model	EM Score		F1 Score	
	Orig Q	Para Q	Orig Q	Para Q
BERT	83.62	79.85	90.78	87.63
DrQA	67.33	65.25	76.25	74.25
BiDAF	67.80	63.84	76.85	73.51

Table 1: Performance of QA models on the original questions (Orig Q) compared to non-adversarial paraphrased questions (Para Q).

Model	EM Score		F1 Score	
	Orig Q	Adv Q	Orig Q	Adv Q
BERT	82.14	57.14	89.31	63.18
DrQA	71.43	39.29	81.02	48.94
BiDAF	75.00	30.36	81.55	38.30

Table 2: Performance of QA models on the original questions (Orig Q) compared to adversarial paraphrased questions (Adv Q).

The adversarial paraphrased test set is able to exploit the reliance of QA models on string matching to cause drastic decrease in the models' performance. BiDAF demonstrated the weakest resilience to such a deliberate attack with a decrease of 43.25 F1, while BERT and DrQA suffered a decrease of 26.13 F1 and 32.08 F1 respectively. This sharp drop in performance highlights a serious flaw in QA models trained on the SQuAD dataset: if we ask a question that matches the context words near a confusing answer candidate, we are likely to get a wrong answer.

4.2 Re-Training Using Training Data Augmentation

Our evaluation suggests that the original training dataset does not contain sufficiently diverse question phrasing. This leads to the models not learning to respond correctly to various ways of asking the same question.

A natural way to improve the robustness of QA models to question paraphrasing would thus be to expose them to more diverse question phrasing. We attempt to achieve this by using our paraphrasing model to paraphrase the training set of questions.

4.2.1 Non-Adversarial Paraphrased Test Set

For improvements on the non-adversarial paraphrased test set, we use the same approach described in Section 3.1.1 to *automatically* generate paraphrased questions from the training set of questions and keep paraphrased questions with similarity score above 0.9. This acceptance threshold is lower than that used in Section 3.1.1 in order to create more diverse paraphrased questions as training data (as a result, these questions are expected to be noisier). No human annotator is employed to check the semantic equivalence of the paraphrased questions and the original questions.

Model	EM Score		F1 Score	
	Before	After	Before	After
BERT	79.85	80.89	87.63	88.62
DrQA	65.25	67.33	74.25	75.00
BiDAF	63.84	66.20	73.51	75.94

Table 3: Performance on the non-adversarial paraphrased test set before and after re-training.

Model	EM Score		F1 Score	
	Before	After	Before	After
BERT	84.02	83.76	91.00	90.88
DrQA	69.04	68.74	78.38	77.86
BiDAF	67.67	67.49	77.46	77.10

Table 4: Performance on the original development set before and after re-training.

We randomly sample 25,000 paraphrased questions to be used as additional training data. We retrain all three QA models using the original training data and the additional 25,000 paraphrased questions. The performance of the three QA models on the paraphrased test set before and after retraining is shown in Table 3.

Even though the augmented training dataset is noisy (since not all generated questions are true paraphrases), all QA models still show improvement on the paraphrased test set after retraining. Furthermore, re-training causes only a negligible drop to the performance of QA models on the original development set, as shown in Table 4.

4.2.2 Adversarial Paraphrased Test Set

In contrast to using PPDB to obtain paraphrase suggestions for the neural paraphrasing model, we now require the paraphrase suggestions to be from the context of the associated question.

We use Flair[6] (Akbik et al., 2018) trained on the Ontonotes dataset[7] which contains 12 named entity classes to label which named entity class, if any, that the answer belongs to. Then, we extract

[6]Pre-trained models available at https://github.com/zalandoresearch/flair

[7]https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf

Model	EM Score		F1 Score	
	Before	After	Before	After
BERT	57.14	69.64	63.18	73.85
DrQA	39.29	41.07	48.94	49.86
BiDAF	30.36	39.29	38.30	47.49

Table 5: Performance of QA models on the adversarial test set before and after re-training.

Model	EM Score		F1 Score	
	Before	After	Before	After
BERT	84.02	83.33	91.00	90.49
DrQA	69.04	67.93	78.38	77.45
BiDAF	67.67	66.23	77.46	76.19

Table 6: Performance on the original development set before and after re-training.

sentences from the context containing named entities of the same type if the named entity contains no overlapping words with the answer.

We perform syntactic chunking on the extracted sentences using Flair trained on the CoNLL-2000 dataset (Sang and Buchholz, 2000). We use the noun and verb phrases from the result of chunking to form the set of paraphrase suggestions for the given question. We ensure that each suggestion obtained contains at least two words and does not overlap with the answer.

After using the paraphrasing model to paraphrase questions from the SQuAD training set using context words as suggestions, we keep only paraphrased questions with paraphrase similarity score above 0.83. This similarity threshold is set lower than the previous selection criterion since we want to allow context words that could be very different from the question words to appear in the generated paraphrase.

We similarly re-train all three QA models with an additional 25,000 paraphrased training examples. The results are shown in Table 5. We see that re-training leads to a significant improvement in the performance of BERT and BiDAF on the adversarial paraphrased test set, although it still falls short of the performance on the corresponding original questions. However, re-training is only able to improve DrQA's performance slightly. In all cases, re-training also only causes a slight decrease in performance on the original SQuAD development set (Table 6).

5 Related Work

We present related work in this section, divided into three sub-topics.

5.1 Adversarial Examples for Question Answering

Jia and Liang (2017) showed that QA models can be confused by appending a distracting sentence to the end of a passage. While this highlighted an important weakness of trained models, the adversarial examples created are unnatural and not expected to be present in naturally occurring passages. In contrast, semantic preserving changes to an input question that lead to returning the wrong answers present more relevant failure cases that occur in practice.

Some previous work used question paraphrasing to create more natural adversarial examples. Ribeiro et al. (2018) made use of back translation to obtain paraphrasing rules that were subsequently filtered by human annotators. Examples of rules obtained include "What VERB → So what VERB" and "What NOUN → Which NOUN". Rychalska et al. (2018) replaced the most important question word identified using the LIME framework with a synonym from WordNet and ELMo embeddings, which was verified by human annotators. These replacements are expected to maintain the meaning of the questions but can sometimes change initially correct answers.

In contrast, we do not restrict ourselves to specific types of paraphrasing when creating the non-adversarial paraphrased test set. Our paraphrasing model can produce paraphrases including but not limited to those in the above two methods. Furthermore, we do not perform any model querying when creating the test set. The ability of our generic approach to decrease the performance of *all* evaluated state-of-the-art QA models demonstrates the need to improve the robustness of current QA models.

The creation of the adversarial paraphrased test set which aims to trick QA models intentionally also contrasts with the approach by Jia and Liang (2017), as the examples created in this work are natural and coherent.

5.2 Neural Paraphrasing Networks

There are a number of neural architectures introduced to automatically generate a paraphrase given an input sentence (Prakash et al., 2016;

Huang et al., 2018; Wang et al., 2018). One conceptually simple approach that does not require a paraphrase corpus is to carry out back translation (Lapata et al., 2017), by first translating the source sentence to a pivot foreign language and back.

Besides single paraphrase generation, the value of generating multiple paraphrases for a given input sentence has also been explored. Gupta et al. (2018) achieved this by using a variational autoencoder (VAE) with a long short-term memory (LSTM) network. Xu et al. (2018) assumed that different paraphrasing styles used different rewriting patterns, which were represented as latent embeddings. These embeddings were used to augment the decoder's hidden state to generate different paraphrases.

In contrast to previous work, we introduce a more guided approach to generate diverse paraphrases, by using a paraphrase suggestion together with a source question to generate a paraphrased question. Given k suggestions, our model is thus able to generate up to k paraphrased questions.

5.3 Paraphrasing as an Intermediate Task to Question Answering

Some previous work considers question reformulation as a subtask of question answering. The intuition for doing this is to reduce the space of question paraphrases that the QA model is required to understand. Models trained by this approach are expected to be more robust to various question paraphrases since the model can paraphrase a question to one which it understands.

Dong et al. (2017) first generated multiple paraphrases for a given question and used a neural network to score the quality of each paraphrase. The probability distribution of the answer was then generated for each paraphrased question, which was subsequently weighted by the score of each paraphrased question to compute the overall conditional probability of the answer given the question. Buck et al. (2017) formulated QA as a reinforcement learning problem and introduced a paraphrasing agent trained to paraphrase a question to one that was able to get the best answer from the QA model. Similarly, multiple question paraphrases were generated to obtain multiple answers from the QA model before answer selection was performed.

In contrast to previous work, we consider question paraphrasing as a separate task instead of a subtask. Our approach is conceptually simpler since it only augments the training data to expose models to various question paraphrases and requires no change to the system during test time. Furthermore, the previous approaches require multiple queries to the QA model for a single question, resulting in longer inference time.

6 Conclusion

In this paper, we propose a novel approach to train a neural paraphrasing network to paraphrase questions utilizing paraphrase suggestions. We use the approach to construct a test set of paraphrased SQuAD questions containing questions similar to the original to test models' robustness to question paraphrasing. We also create an adversarial paraphrased test set to test models' reliance on string matching. We show that all three state-of-the-art QA models give poorer performance on the first test set and drastically reduced performance on the second test set. We also show that a completely automatic approach to augment the training data can improve the robustness of the QA models to the paraphrased questions, while still retaining performance on the original questions. Our experiments highlight the need for separate adversarial testing and the importance of improving the robustness of QA models to question paraphrasing for better reliability when tested on future unseen test questions.

There are several possible future directions stemming from this work. As post-processing is required to remove semantically dissimilar paraphrased questions, there is scope for developing better techniques for semantic similarity scoring. There is also scope for better techniques to generate more coherent question paraphrasing when significant question re-writing is required, such as for the situation when we want to paraphrase the question using context words. In addition, we have only considered paraphrasing the question in this paper. Paraphrasing the context is another area to explore but poses significant technical challenge, since it requires altering words over multiple sentences while still retaining the original meaning of the context.

Acknowledgments

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme AISG-RP-2018-007.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *CoRR*, abs/1705.07830.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1870–1879.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1608–1618.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5149–5156.

Shaohan Huang, Yu Wu, Furu Wei, and Ming Zhou. 2018. Dictionary-guided editing networks for paraphrase generation. *CoRR*, abs/1806.08077.

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.

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*, pages 1601–1611.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M. Rush. 2018. OpenNMT: neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pages 177–184.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations)*, pages 66–71.

Mirella Lapata, Rico Sennrich, and Jonathan Mallinson. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 881–893.

Rada Mihalcea and Paul Tarau. 2004. TextRank: bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 425–430.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2923–2934.

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*, pages 784–789.

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.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Marco Túlio 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*, pages 856–865.

Marc-Antoine Rondeau and Timothy J. Hazen. 2018. Systematic error analysis of the Stanford question

answering dataset. In *Proceedings of the ACL Workshop on Machine Reading for Question Answering*, pages 12–20.

Barbara Rychalska, Dominika Basaj, and Przemyslaw Biecek. 2018. Are you tough enough? Framework for robustness validation of machine comprehension systems. *CoRR*, abs/1812.02205.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task chunking. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, pages 127–132.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Thirty-First Conference on Neural Information Processing Systems*.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2018. A task in a suit and a tie: paraphrase generation with semantic augmentation. *CoRR*, abs/1811.00119.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 451–462.

Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018. D-PAGE: Diverse paraphrase generation. *CoRR*, abs/1808.04364.