Data Augmentation and Ensembling for FriendsQA

Stanford CS224N Custom Project—March 19, 2021

Jazon Jiao
Department of Computer Science
Stanford University
canwen@stanford.edu

Nina Du
Department of Computer Science
Stanford University
ninacdu@stanford.edu

Abstract

FriendsQA is a challenging QA dataset consisting of 10,610 questions, based on multi-person dialogues from the TV series *Friends*. We augmented its training data using back-translation, and proposed a novel method to effectively find answers in paraphrased contexts, by comparing the sum of word embeddings. Combining data augmentation with ensembling of BERT-large models, we pushed state-of-the-art F1 / EM scores on FriendsQA from 69.6 / 53.5 to 72.08 / 54.62.

1 Key Information to include

- Mentor: Andrew Wang
- External Mentor: Yi Cheng (yicheng98@163.com)

2 Introduction

Rapid progress on Question Answering (QA) has been made in recent years. However, widely used benchmarks on QA, such as SQuAD[1], Natural Questions[2] and NewsQA[3], mostly consist of passages from Wikipedia or other online sources, yet this is only one category of human languages. One other crucial aspect of languages comes in the form of everyday conversations, and understanding them is equally important for better machine comprehension on human languages.

In this paper, we explore the FriendsQA dataset, first presented by Yang and Choi [4]. FriendsQA is a question answering dataset that contains 1,222 dialogues and 10,610 open-domain questions based on transcripts from the TV show *Friends*. It is the first dataset that challenges span-based QA on multiparty dialogue with daily topics. An example scene and associated QA's is described below. Each utterance is either a "note", or a (speaker, utterance) pair recording what a speaker has said; "uid" refers to utterance id. Here, each question has 2 "gold" answers that may disagree.

uid	**Speaker**	Utterance
0	**#Note#**	*Central Perk, the gang is there, Phoebe is returning from the bathroom.*
1	**Phoebe**	That's like the tenth time I've peed since I've been here!
2	**Monica**	That's also like the tenth time you told us.
3	**Phoebe**	Yeah, oh I'm sorry, it must be really hard to hear!
4	**Ross**	Pheebs, did ... you want a cookie?
5	**Phoebe**	Thank you so much.
6	**Rachel**	So uh, Pheebs, honey, how are those mood swings coming?
7	**Phoebe**	I haven't really had any yet.
8	**#Note#**	*Monica, Joey, and Chandler all shake their heads.*

Question	**Gold Answers**	uid	index$_{start}$	index$_{end}$
1. Where is the gang all at ?	**Central Perk**	0	3	4
	, the gang is there	0	5	9
2. How many times has Phoebe been to the bathroom at Central Perk ?	**tenth time**	1	4	5
	the tenth time	1	3	5

3 Related Work

3.1 QA on multi-party dialogues

The previous best score on the FriendsQA dataset is obtained by Li and Choi [5]. Their paper introduces a new approach to transformers that learns hierarchical representations in multiparty dialogue to tackle machine comprehension on FriendsQA dataset. They proposed two pre-training tasks, utterance-level masked LM and utterance order prediction, to improve the quality of both token-level and utterance-level embeddings generated by the transformers. It then introduces a new multi-task learning approach, the joint inference between token span prediction and utterance ID prediction, to fine-tune the language model for span-based QA. The approach shows improvement of 3.8% and 1.4% over the previous state-of-the-art transformer approaches, BERT and RoBERTa on FriendsQA, achieving F1 scores of 63.1 and 69.6 on these two models respectively.

Notably, the authors modified BERT to first predict the utterance id in which the answer lies, and then predict the exact span within the utterance. In our project, we concatenated all utterances and tried to predict the answer span directly, without the intermediate step of predicting uid.

Another study on FriendsQA is presented by Liu et al. in their paper about graph-based knowledge for QA [6]. They devise a relational graph convolutional network (R-GCN) on dialogues using edges to represent relationships between entities, which is especially important for resolving relationships in a dialogue setting.

Apart from FriendsQA, the only two other datasets about QA on multi-person dialogues that we know are Molweni [7] and DREAM [8]. However, Molweni is too new for researchers to have explored in-depth, whereas for DREAM, the questions come in the form of multiple choice, and thus it is in essence different from FriendsQA.

3.2 BERT and DistillBERT

DistillBERT is a 40% smaller, 60% faster, distilled version of the original BERT model [9]. BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. It obtains excellent results on eleven natural language processing tasks, including pushing both SQuAD v1.1 question answering Test F1 and SQuAD v2.0 Test F1 to 1.5 point and 5.1 point absolute improvement respectively. As a general-purpose pre-trained version of BERT, DistillBERT leverages knowledge distillation during the pre-training phase in order to reduce the size of a BERT model. By using a teacher-student network while reducing the token-type embeddings and optimizing the operations in the Transformer architecture, DistillBERT reduces the size of a BERT model by 40% yet still retains 97% of the language understanding capabilities.

3.3 Data Augmentation

As FriendsQA is a unique and relatively small dataset, we strove to improve model performance by enriching the dataset with data augmentation methods.

3.3.1 Back-translation

Back-translation is an effective method to improve neural machine translation (NMT) with monolingual data [10]. Although back-translation has been widely used to improve the same translation task[11] or intrinsic paraphrase evaluation [12] [13], researchers in the QANet paper [14] are the first to propose the application of back-translation to enrich traning data for the QA task. In the paper, researchers combine their new QA architecture, QANet, with back-translation and observe that back-translation can bring non-trivial improvement in terms of accuracy.

3.3.2 EDA

Another data augmentation technique we tried to apply on our QA task is Easy Data Augmentation (EDA) proposed by Wei and Zou[15]. It is originally a data augmentation technique on text classification tasks. EDA consists of four operations: synonym replacement, random insertion, random swap and random deletion.

However, we found that EDA did not perform well on our tasks. Since utterances are most often only a few words long, insertion/swapping/deletion often fundamentally change sentence meanings, while replacement doesn't perform as effectively as back-translation.

4 Approach

4.1 Data Preprocessing

The FriendsQA dataset [4] is in many ways similar to SQuAD. We transformed its JSON files to match the format of the SQuAD dataset; for this project, we simply concatenated all speakers, utterances, and descriptions of one scene into one paragraph to serve as our context, for example:

```
[Scene: A 747 somewhere over the North Atlantic, Monica and Chandler are
sitting in first class, depressed.] Monica Geller: Y'know, maybe it's
best that we never got to do it again. Chandler Bing: Yeah, it kinda makes
that - that one night special. Y'know, technically we still are over
international waters ...
```

4.2 Data augmentation via back-translation

We used 8 Helsinki-NLP OPUS-MT models [16] for back-translation between English and Chinese, German, Spanish and French, respectively. We did only one iteration of back-translation, such as English → German → English. (Doing more iterations would mean back-translating based on back-translations, such as English → German → English → German → English...)

We produced 4 new training datasets (one for each intermediate language) where all contexts and questions are replaced with back-translated versions. (Answers are not back-translated, as discussed in Section 4.3.3. Also, Dev and Test Sets are not augmented).

Note that the inputs to the translation model are single utterances instead of entire contexts. This is beneficial since the translations have better accuracy for shorter inputs. We then concatenate the translated utterances into a context, as specified in Section 4.1.

The translations generated by the model are often "direct translations" that are ungrammatical in the target language; however, when translated back to English, they often produce well paraphrased results. See the following Table for an example.

4.3 Finding answers in paraphrased text

For span-based QA tasks, once a context is paraphrased, the answer to a question may no longer be contained in the context. We need to find a span in the paraphrase, a_{new}, that most resembles the original gold answer, denoted a_{orig}.

Note that for FriendsQA, we only need to find a_{new} in the same utterance as a_{orig}, which narrows down the search considerably. This is because a gold answer cannot span over two or more utterances, and each gold answer is associated with the utterance that contains it.

4.3.1 Character-based comparisons

In the QANet paper [14], this is done by a simple heuristic: take the candidate with the highest character-level 2-gram score with respect to the original answer. At first, we tried a similar method, computing the F1 score between the characters of a candidate and a_{orig}, and taking the candidate with the highest F1 score.

Original	Scene: Phoebe is entering carry a large box, Monica is mopping the ceiling. ...
Back-translated	Scene: Phoebe entering with a big box, Monica is rubbing the roof. ...
(Translation)	场景:菲比进入携带一个大盒子,莫妮卡正在擦屋顶。
Question	How is Monica getting the banana off of the ceiling?
Gold Answer	Monica is mopping the ceiling.
Old Method	entering with a big box, Monica is *(F1 = 0.690)*
New Method	Monica is rubbing the roof. *(NMSE = 0.135)*

Table 1: An example of back-translation and two different methods of finding answers in the paraphrase, from Scene ID s03-e16-c07. Note that although the back-translated sentence is reasonable, the Chinese translation is not grammatical, as it uses the same word order as English. The correct word order in Chinese for "Phoebe enters with a big box" would be "Phoebe with a big box enters".

This character-level comparison works for the most part, but it fails to capture word meanings, and is ineffective for many well-paraphrased texts.

One would naturally think of comparing sentence embeddings. However, sentence embeddings could be problematic, since it would be hard to determine the exact span of the answer. If sentence embeddings reflect how sentences are similar to each other in meaning, then nearby spans may have similar embeddings. For the paraphrase in the table above, we could imagine that the algorithm may take, say, *"box, Monica is rubbing the roof."* or *"Monica is rubbing the"* as the answer.

4.3.2 Innovation: Sum of word embeddings

To this end, we propose an innovative way to calculate the similarity scores between a candidate span and a_{orig}. The idea is to incoporate the sense of "number of words" when comparing phrases, by taking the sum of word embeddings and then calculate the error. Please see Appendix A.1 for a graphical illustration for this technique.

Let $v_{candidate}$ be the sum of all word embeddings in a candidate span, and v_{target} for the sum of all word embeddings of a_{orig} [1]. The error is given by

$$\text{NMSE} = \frac{\text{Mean Squared Error}(v_{candidate} - v_{target})}{\sqrt{\text{length of } a_{orig}}} \tag{1}$$

Lower error means higher similarity. The MSE is divided by the square root of gold answer length (i.e. number of words) to obtain "Normalized MSE" (NMSE). This is done to compensate for the disadvantage of long sentences that are paraphrased well, and is useful for the "discard threshold" discussed below.

We go through each $O(n^2)$ possible spans in the paraphrased text, and take the span with the lowest error as the paraphrased answer. If a question has no answer with NMSE below 0.2, then we consider it to be "no answer found", and the question is discarded. This threshold is chosen empirically by inspecting the output. The following table gives the number of questions discarded and augmented for each back-translation language (each row sums to 9791, the number of traininig examples).

Language	# Discarded	# Augmented
Chinese	1510	8281
German	452	9340
Spanish	796	9107
French	684	8995

Although this method can effectively identify long paraphrased answers (see Table 1 on the previous page for a comparison with the previous method), it still sometimes fail when the answer texts are very short, since the NMSE is quite unstable when comparing just one or two words.

4.3.3 Back-translate the gold answer?

In an effort to improve accuracy, We also tried to back-translate the gold answer, and compare each candidate span to not only the gold answer, but also the paraphrased answer.

However, we found out that sometimes a bad candidate can achieve a high score (low error) if the corresponding paraphrased answer is also badly translated. One way this can happen is when the paraphrased answer cuts off half of the gold answer. As such, we chose to compare the candidates spans only to the gold answer.

4.4 Ensembling

Ensembling is a popular way to boost performance of BERT (and any machine learning model) [17] [18]. For our task, the score of each candidate answer span is the sum of the `start_logit` and `end_logit` values in the final layer of `BERTForQuestionAnswering`. Then, for each candidate answer, we add up the score given by each model (each model is assigned equal weight).

[1] We adapted cs224n Assignment 1 code and loaded word embeddings from "glove-wiki-gigaword-300". If a word is not in the vocabulary, it is assigned a random word embedding with random weights from -1 to 1.

5 Experiments

We adapted the RobustQA default project code [2] for our experiments.

5.1 Data

The input data has the same format as SQuAD, as described in Section 4.1. Due to the input size limit of BERT, contexts with length greater than `max_length=384` are split into multiple contexts, with overlaps between them. Since the average context length in FriendsQA is 499.5 tokens, this means most of the contexts need to be split (for comparison, the average context length for SQuaD is 260.3 tokens). The longest context needs to be split into more than 10 segments. This splitting is automatically done for us by the RobustQA code.

5.2 Evaluation method

The 3 papers on FriendsQA all use 3 metrics for evaluating the model's overall performance. In decreasing order of strictness, they are: Exact Match (EM), Span Match (SM, which is equivalent to F1), and Utterance Match (UM).

The UM score is the percentage of predicted answers that lie within the same utterance (uid) as "gold"—as long as an answer falls on the same line of text as "gold", it's counted as correct. We did not use Utterance Match because we concatenated all utterances as the context to each question.

We evaluate the EM and F1 scores for our models, and used RobustQA project code for calculating them. This means the gold and predicted answers are normalized (punctuations are removed, etc.) before comparisons are made.

5.3 Base models

A baseline we implemented was from the "question-answering" pipeline of the Hugging Face `transformer` Python module [19]. This pipeline could perform generic question answering with a very simple user interface. It managed to get F1 / EM scores of 45.4 / 33.4 on the Dev Set.

It turns out that the pipeline uses a pretrained model available in the `transformer` module called `distilbert-base-uncased-distilled-squad`, i.e. DistilBERT fine-tuned on SQuAD. In our later experiments, we would continue to fine-tune this model on FriendsQA.

A counterpart model is available for BERT-large, named `bert-large-uncased-whole-word-masking-finetuned-squad`. It achieves F1 / EM scores of 60.9 / 42.9 when directly evaluated on FriendsQA dev set. This model serves as the starting point of our BERT-large fine-tuning experiments.

Fine-tuning based on SQuAD-finetuned models is important, since as [20] showed, one of the best ways to boost BERT performance is through "Transferring via an Intermediate Task", which means first training the model on a relevant annotated dataset before fine-tuning on downstream tasks.

5.4 Experiment details

Hyperparameters: We use the default learning rate of `3e-5`, and batch size of 16; in our case, increasing the learning rate would decrease performance, and increasing the batch size has little influence over training outcomes. For all experiments presented here, the number of epochs is set to 1. Performance almost always decreases if there is a second epoch, because the model quickly overfits on our small dataset.

Validation and testing: We followed the procedure of training on the training set, validating on the dev set during training, and evaluating on the test set after training is complete. The model that gives the highest F1 score across all validations during training is saved; the saved model is used later for ensembling.

We later realized that for the default SQuAD project, students are only allowed to evaluate on the test set 3 times, whereas we tested after completing each round of training. In retrospect, a better practice would be to save the test set for evaluation of our final models. Still, we did not use the test set to tune or select our model, but only used it for evaluation.

[2] `https://github.com/MurtyShikhar/robustqa`

5.5 Results

Before showing our results, it's worth noting that FriendsQA has 10 times fewer dev and test questions than SQuAD, so small differences in the dev and test results are more likely to be due to statistical fluctuations.

5.5.1 Fine-tuning on augmented data

#	Training Data	DB (dev)	BL (dev)	BL (test)
#0	None	45.40 / 33.38	60.86 / 42.89	-
#1	en	57.57 / 39.42	67.01 / 48.91	-
#2	en + de	57.20 / 40.19	68.43 / **50.83**	68.16 / 50.29
#3	en + es	58.08 / 40.89	68.20 / 49.87	68.96 / 51.04
#4	en + fr	57.73 / 40.10	**68.57** / 50.39	**70.12 / 52.21**
#5	en + zh	57.99 / 39.76	67.00 / 49.14	68.63 / 50.04
#6	en+zh+de+fr+es	56.79 / 38.97	-	-

Table 2: F1 / EM scores of fine-tuning on augmented data. In the "Training Data" column, "en" stands for the original data, "de" stands for data back-translated on German, "es" for Spanish, "fr" for French, and "zh" for Chinese. "**DB**" stands for DistilBERT (fine-tuned on SQuAD), while "**BL**" stands for BERT-large (fine-tuned on SQuAD).

As the table shows, models jointly trained on the augmented datasets (#2 - #5) does outperform models trained on the original dataset only (#1), which proves the effectiveness of our back-translation techniques. The BERT-large model in #4, with an F1 score of 70.12 on the test set, was our first (and smallest) model to beat the state-of-the-art F1 score of 69.6.

For #6 we trained the model on the original dataset along with all 4 augmented sets. It is not clear to us why it performed poorly. A plausible reason is the model is given many near-identical data, and so quickly overfits. As such, we did not repeat this experiment on BERT-large.

5.5.2 Ensembling

We took the BERT-large models fine-tuned on the augmented FriendsQA datasets, and ensembled these models. Ensembles outperform single models, and the more models we have, the better results we get.

Models	Dev F1 / EM	Test F1 / EM
#2 + #3	69.25 / 51.10	-
#2 + #3 + #4	69.63 / 51.10	-
#2 + #3 + #4 + #5	70.41 / 51.86	71.41 / 53.71

5.5.3 More training data and more ensembling

Just before the project is due, we fine-tuned another BERT-large model on a combination of 5 datasets: FriendsQA, German back-translated FriendsQA, NewsQA and Natural Questions (taken from the RobustQA project), and Molweni [7]. Ensembling this model with #2 + #3 + #4 + #5 gives a F1 / EM score of **72.08 / 54.62** on the Test Set, pushing the previous top F1 / EM scores by 2.5 / 1.1, respectively. The table below compares our results to previous works on FriendsQA.

Paper	Model	UM	F1 (SM)	EM
Liu et al., 2020 [6]	$\text{BERT}_{\text{base}} + \text{Graph}$	74.1	65.5	48.1
Li and Choi, 2020 [5]	$\text{RoBERTa}_{\text{base}}$	**82.7**	69.6	53.5
ours, 2021	ensemble of 5 $\text{BERT}_{\text{large}}$	N/A	**72.1**	**54.6**

6 Analysis

In this section, we inspect the output of the 4-model ensemble (#2 + #3 + #4 + #5, with an F1 score of 71.41) on the Test Set. In each example, an excerpt of the relevant context (with utterance ids) is shown, followed by the Questions and Answers, and finally the top predictions and associated scores given by each of the 4 models.

The following table gives the number of correct/incorrect answers by whether the models agree. “Models agree” means all 4 models output the same top answer, while “models disagree” means at least 1 model gives a different top answer than the others. For comparison, the best-performing single model (#4) got 621 answers correct.

	models agree	models disagree	total
correct	384	261	645
incorrect	111	445	556

We manually went through the 111 answers where the models agree but turns out incorrect, and divided them into 4 categories depending on whether the gold answers themselves are correct and the predicted answers include certain portion of contents of gold answers. The following table shows the numbers of answers in each category. “gold c” or ‘gold w” indicates whether we believe that gold answers are correct (“c”) or wrong (“w”); “pred +” or “pred -” indicates whether gold answers are included (“+”) in the predicted ones or not (“-”). Almost half of the predicted answers are marked incorrect due to either omission of certain parts of the gold answers or inclusion of unimportant details, not because the prediction is inherently “wrong”.

gold c, pred +	gold c, pred -	gold w, pred +	gold w, pred -	total
54	38	14	5	111

Unlike SQuAD which has 3 gold answers per question, FriendsQA only provides one or two gold answers per question. We believe that this limitation might lead to more subjectivity in prediction, which can affect the judgement of the correctness and result in many “gold c, pred +” predictions.

Below is an instance where there is only one gold answer to a question, and the gold answer is incorrect, while the models agree on a sensible answer.

(**u6**) Rachel: Huh. Except, Phoebe’s not gonna be the one that gets to dress them.
(**u7**) Monica: Because she’s not gonna get to keep the babies.
(**u9**) Monica: Wait a minute! Unless, we give her all gifts she can use after she’s done being pregnant. Like - like umm, regular coffee, Tequila.

Q: *Why does Phoebe not get to dress the babies?*
A: regular coffee

Total	#2	#3	#4	#5	Prediction
50.2	**11.2**	**13.3**	**13.4**	**12.2**	Because she’s not gonna get to keep the babies
46.4	10.5	11.9	12.6	11.3	Because she’s not gonna get to keep the babies.
45.9	10.0	11.3	12.7	11.9	she’s not gonna get to keep the babies
42.1	9.3	9.9	11.9	11.0	she’s not gonna get to keep the babies.
17.3	7.4			9.9	not gonna get to keep the babies
17.1		7.4	9.8		to keep the babies.
17.0		7.5	9.5		keep the babies
10.1				10.1	get to keep the babies.
8.3	8.3				’s not gonna get to keep the babies

An example of the predictions omitting irrelevant details is as follows:

(**u13**) Chandler, Joey: He - he - he got in, he - he got in to San Diego.

Q: *Where did he get in?*
A1: he got in to San Diego
A2: He-he-he got in, he-he got in to San Diego.

Total	#2	#3	#4	#5	Prediction
47.7	**6.7**	**13.0**	**15.7**	**12.4**	San Diego
45.6	6.5	12.7	14.8	11.6	San Diego.
43.3	5.9	11.6	14.3	11.6	to San Diego
41.2	5.7	11.3	13.4	10.8	to San Diego.
40.6	5.0	11.3	13.6	10.6	he got in to San Diego

In the vast majority of cases, the top answers differ only by whether to include nearby spans. But there are some instances where the models disagree on a fundamental level, and ensembling helped to choose the correct answer based on majority vote. For example:

(**u5**) Monica: Did you just smell my hair?
(**u6**) Peter: Nooo. Uh-huh, no way. What? No.
(**u22**) Monica: All right shut up for a second and let me just see something. Oh, wow!

Q: *At what point did Pete smelling Monica's hair became obvious to her?*
A: Did you just smell my hair?

Total	#2	#3	#4	#5	Prediction
6.2	**0.1**	3.5	-	2.6	Did you just smell my hair?
5.8	-	3.0	-	**2.8**	just smell my hair?
4.9	-	-	**4.9**	-	Nooo
4.4	-	**4.4**	-	-	a second

The following example shows a drawback of concatenating all utterances into one context. Here, the answer consists of one word, "today", and this word occurs two times nearby in the context. The `start_logit` and `end_logit` scores for both occurrences of "today" are high, so the top answer included the context in between two "today"s, spanning two utterances. This would not have happened if we only allow answers to reside in only one utterance.

(**u13**) Peter: So ask me what I did today.
(**u14**) Monica: So what did you do today Pete?
(**u15**) Peter: I bought a restaurant and I would like you to be the head chef.

Q: *When did Pete buy a restaurant?*
A: today

Total	#2	#3	#4	#5	Prediction
45.7	10.4	**11.1**	**14.9**	**9.3**	today. Monica Geller: So what did you do today
43.9	**11.5**	10.5	13.3	8.5	today.
38.3	10.3	8.4	13.4	6.3	today. Monica Geller: So what did you do today Pete?
29.2	9.0	8.2	11.9	-	today. Monica Geller: So what did you do today Pete

7 Future Work

Due to limited time, we only tried very basic hyperparameter tuning. It would be worth exploring the best learning rate, regularization parameters, etc., for our task. Additionally, BERT has a input size limit that is smaller than the average context length of FriendsQA. Although we managed to work around the problem by splitting long contexts, this might potentially harm the model performance. Models like Longformer [21] may be able to better understand long contexts.

8 Conclusion

In this project, we aim to improve model robustness and performance on FriendsQA dataset via data augmentation and ensembling. We generated 4 new training datasets of well-paraphrased contexts and questions through back-translation. We proposed a novel method to find answers in paraphrased text through the use of the sum of word embeddings. When looking for answers in the back-translated context, we compared phrases by taking the sum of word embeddings before the calculation of the normalized mean squared error. This method effectively compensates for the disadvantages of sentences that are paraphrased well but have long length compared to the original ones. We trained BERT on the augmented datasets, and then ensembled BERT-large models, achieving state-of-the-art performance on FriendsQA dataset with F1 / EM scores of 72.1 / 54.6.

9 Acknowledgement

We would like thank Yi Cheng for providing the project idea and offering high-level advice during the course of our project.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. CoRR, abs/1606.05250, 2016.

[2] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. Transactions of the Association of Computational Linguistics, 2019.

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

[4] Yang et al. FriendsQA: Open-Domain Question Answering on TV Show Transcripts. In Association for Computational Linguistics, 2019.

[5] Li et al. Transformers to learn hierarchical contexts in multiparty dialogue for span-based question answering. In Association for Computational Linguistics, 2020.

[6] Jian Liu, Dianbo Sui, Kang Liu, and Jun. Zhao. Graph-based knowledge integration for question answering over dialogue. In International Committee on Computational Linguistics, 2020.

[7] Li et al. Molweni: A challenge multiparty dialogue-based machine reading comprehension dataset with discourse structure. In COLING, 2020.

[8] Sun et al. Dream: A challenge dataset and models for dialogue-based reading comprehension. In Association for Computational Linguistics (ACL), 2019.

[9] Devlin et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In arXiv preprint arXiv:1810.04805, 2018.

[10] et al. Sergey Edunov. Understanding back-translation at scale, 2018.

[11] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. CoRR, abs/1511.06709, 2015.

[12] John Wieting, Jonathan Mallinson, and Kevin Gimpel. Learning paraphrastic sentence embeddings from back-translated bitext. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 274–285, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[13] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Paraphrasing revisited with neural machine translation. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 881–893, Valencia, Spain, April 2017. Association for Computational Linguistics.

[14] Yu et al. QANet: Combining local convolution with global self-attention for reading comprehension. In arXiv preprint arXiv:1804.09541, 2018.

[15] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. CoRR, abs/1901.11196, 2019.

[16] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT), Lisbon, Portugal, 2020.

[17] Yige Xu et al. Improving BERT fine-tuning via self-ensemble and self-distillation. In arXiv preprint arXiv:2002.10345, 2020.

[18] Charlie Xu et al. Applying ensembling methods to bert to boost model performance. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15775971.pdf. CS224n 2019 Final Project.

[19] Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[20] Tianyi Zhang et al. Revisiting few-sample BERT fine-tuning. In arXiv preprint arXiv:2006.05987, 2021.

[21] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. In arXiv preprint arXiv:2004.05150.

A Appendix

A.1 Illustration of Section 4.3.2

Original context: Wow, this is so cool, you guys. The entire city is blacked out!
Paraphrased: Wow, that's cool, you guys. The whole city is dark!
Question: What's happening?

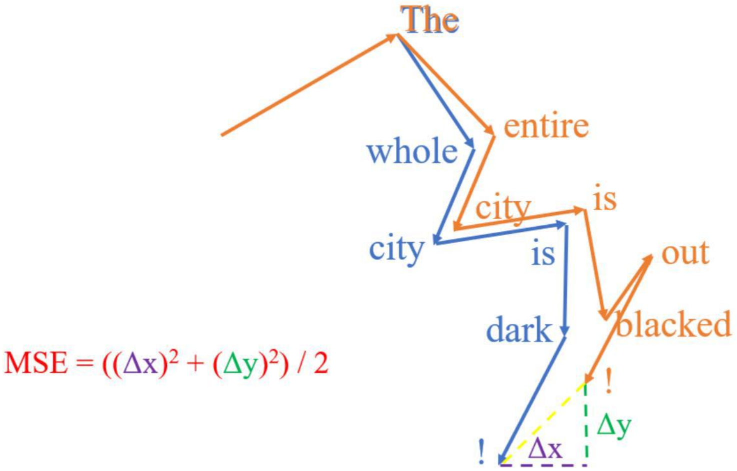

Figure 1: 2D illustration of our sum-of-word-vectors technique for finding the answer in a paraphrased context, used for data augmentation. Orange represents the original answer, and blue indicates the paraphrased answer that we wish to find. The span that gives the smallest MSE is selected as the paraphrased answer.

A.2 Funny back-translations

Back-translations sometimes produce hilarious results that we feel is worth sharing. In the following examples, the first row, **O**, stands for the original sentence; the second row, **B**, stands for the back-translated sentence; and the third row stands for the intermediate translation, with Z, D, F, E representing Chinese, German, French, and Spanish, respectively.

Example 1:

O: How Chandler responds when he learns Joey was locked out?

B: What did Chandler say when he learned about Joey 's incarceration?

Z: 钱德勒得知乔伊被关禁闭后是怎么回应的?

For sentences with repeating words or characters (which is abundant for dialogues), the back-translations often repeat them many more times. Perhaps this is effective data augmentation as it introduces noise into the data!

Example 2:

O: Aaaahhhhhhh.

B: hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh... (514 h’s)

F: Aaaahhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh... (512 h’s)

Example 3:

O: Morning’s here! The morning’s here!

B: Good morning, good morning, good morning, good morning, good morning, good morning, good morning, good morning, good morning, good morning, good morning, good morning.

Z: 早安,早安!