RobustQA: Benchmarking Techniques for Domain-Agnostic Question Answering System

Stanford CS224N Default (Robust QA track) Project
Mentor: Davide Giovanardi
Late Day Sharing:
Yijie (2 days) Zhu (4 days) Zelin (4 days) → 1 day for milestone & 2 days for report

Yijie Sun
Department of Statistics
Stanford University
yijiesun@stanford.edu

Zhu Shen
Department of Statistics
Stanford University
zhushen@stanford.edu

Zelin Li
Department of Statistics
Stanford University
jameszli@stanford.edu

Abstract

The goal of our project is to build a question answering system that is robust to out-of-distribution datasets. Motivated by the Apple Inc. team's approach at the 2019 MRQA workshop [1], we paraphrased both the in-domain and out-of-distribution training sets by back-translating each query and context pair to multiple languages using architectures that include a two-layer neural machine translation (NMT) system and pretrained transformers. By finetuning the DistilBERT baseline on these augmented datasets, our best model achieved 51.28 F1 and 35.86 EM on the development set and 59.86 F1 and 41.42 EM on the test set.

1 Introduction

Despite all the hype about large pretrained transformers like BERT and ROBERTA, recent studies have suggested that domain-adaptive pretraining, followed by a more restrictive finetuning on task-relevant data may yield considerable performance gains [2]. This approach, however, is resource intensive and cannot be applied to the cases where only few examples of the target domain are available.

To make things even worse, the neural networks tend to capture minuscule correlations in the text, leading to drastically different result when the input is slightly disturbed, which poses a significant challenge for the Question Answering (QA) task due to the large discrepancy between the training and test corpus. In fact, models that outperform human on SQuAD [3] are found to be overfitting and generalize poorly to out-of-distribution datasets [4].

To address these issues, we approach the robustness problem by augmenting both the in-domain (*ind*) and out-of-distribution (*ood*) training data. For each query and context pair (q, c) in the training set, a query paraphrase q' and a context paraphrase c' was generated by first translating (q, c) to a pivot language using either the 2-layer `seq2seq` NMT system or pretrained transformers and then translate it back to English.

By adding these label preserving invariances to the finetuning procedure, we hope to reduce the learned features specific to the *ind* data, while increasing the number of the *ood* data so that our QA model can generalize more broadly.

2 Related Work

Existing works on the topic of improving robustness of the QA system diversify into multiple directions. Jacobs et al. [5] approaches this problem through the Mixture-of-Experts (MoE) method, where each of the datasets has its own DistilBERT model [6] with a multi-layer perceptron model as the gating function to decide the mixture weight of each of the dataset model. Zhang et al. [7]

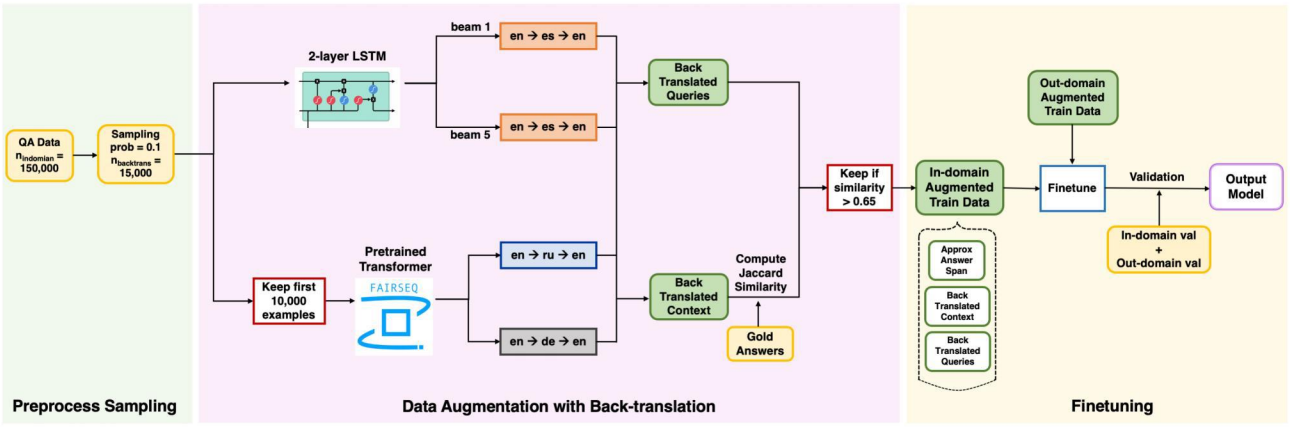

Figure 1: Approach Overview

instead focuses on using few-sample finetuning, where they explore the effects of hyper-parameters, such as learning rate and number of gradient update steps, on increasing accuracy. The most relevant work to our approach is Apple Inc. team's paper at the 2019 MRQA workshop [1], where they explored three data augmentation and sampling techniques: negative sampling, augmentation through back-translation and weighted sampling motivated by active learning. We specifically implement the back-translation method in our project.

There are also several other papers that purely look at the effect of back-translation. Edunov et al. [8] investigate the effect of sampling or noisy synthetic data compared to data generated by greedy or beam search. Poncelas et al. [9] examine the effect of back-translated data size on performances of the NMT systems. We extract partial methods from these papers and combine them to create our own method in implementing data augmentation for finetuning purpose.

3 Approach

3.1 Baseline

The baseline model we used is the direct output from the finetuned DistilBERT model [6] provided by the course staff member. We trained the model on Microsoft Azure and achieved an Exact Match (EM) score of 33.246 and an F1 score of 48.432 on the *ood* validation set.

3.2 Preprocessing

Due to computing resource constraint, we uniformly sampled 10% of the *ind* training examples (15K out of 150K total) and used all *ood* training examples (381 total) for paraphrasing. To conform to the input data format expected by the machine translation systems, the contexts were first split into sentences using `spaCy` [10]. We removed blank lines and special encoding tokens such as `\u20x` to avoid packing empty tensors during the translation. We also added questions marks to the end of queries if they did not exist already. Otherwise, the model would fail to recognize the end of the line and would keep repeating the last words. Meanwhile, we recorded the context sentence where the gold answer a is located in order to facilitate the search for approximate answer span after back-translation.

3.3 NMT

For both NMT models (English-to-Spanish and Spanish-to-English translation), we used a `seq2seq` network consisting of a Bidirectional LSTM Encoder and a Unidirectional LSTM Decoder with global multiplicative attention.

The training corpus was sourced from the English-Spanish TED talks dataset (20MB) available on the CS 224N Winter 2020 website [11], which contains 216,617 parallel sentences. Instead of adopting Byte Pair Encoding as the Apple Inc. team, we performed tokenization using `SentencePiece` since

it is whitespace agnostic and gives more stability. Using domain knowledge in linguistics, the size of the source language (English) vocabulary was set to 8,000 and that of the pivot language (Spanish) was set to 13,000.

Using the paper by Luong et al [12] who built a English-Vietnamese NMT system using a TED talks dataset of similar size (133K parallel sentences), we experimented with various hyperparameter values and set learning rate = 7.5e-4, patience = 2, embedding size = 512 and hidden size = 512.

For fast iterations of testing, we started with the 1-layer LSTM implementation in CS 224N Assignment 4 [13]. However, the 1-layer NMT model did not yield satisfying results: We observed lots of `<unk>` tokens and non-translatable sentences due to the discrepancy between the vocabulary used in the TED talks training corpus and the QA dataset which is extracted mainly from News articles and Wikipedia. To increase the expressivity of the NMT model, we adapted the code to build a 2-layer LSTM network with modifications to facilitate running experiments on English-Spanish translation (Table 1 shows the improvement compared to the 1-layer network).

Once trained, the NMT model was used to translate the queries and context from the QA dataset. After each translation, we parsed the paraphrases and dropped the entire example if any sentence in the query or the context is non-translatable.

Table 1: NMT Architecture Comparison

Model	Language	Train Perplexity	Val Perplexity	BLEU
1-layer NMT	English $\rightarrow$ Spanish	6.58	8.36	29.02
	Spanish $\rightarrow$ English	5.44	6.82	29.94
2-layer NMT	English $\rightarrow$ Spanish	3.80	6.78	33.76
	Spanish $\rightarrow$ English	3.62	6.16	35.29

3.4 Transformer

We still observed empty translations and numerous `<unk>` tokens in the back-translated examples. These problems can be ascribed to the small training corpus, significant differences in training sources and inadequate flexibility of our NMT model. Considering the time and computing resources to train larger-scale NMT systems, we leveraged the pretrained transformers by `FAIRSEQ`. Specifically, we utilized the winning model of the WMT19 Shared News Translation Task built by Facebook AI Research team [14] in two language pairs: English (EN) $\leftrightarrow$ German (DE) and English (EN) $\leftrightarrow$ Russian (RU). The training corpus of these transformers come from the News Crawl, Common Crawl and Wikipedia, which is rather similar to our QA training data sources. The DE-EN translation task dataset has 38,690,334 examples (9.71GB) and RU-EN dataset has 38,492,126 examples (3.86GB). Due to the slow translation speed, we only included the first 10K examples of the sampled 15K examples for back-translation using transformer.

3.5 Approximate answer span

Since the true answer span might be lost in the paraphrased context, we had to compute an estimated answer span after back-translation. We could locate the paraphrased context sentence containing the true answer rather easily since we recorded their positions in the preprocessing step. Then we iterated over all continuous subsets of words and find the span that is the most similar to the original gold answer. To characterize the pairwise similarity, we considered the Jaccard similarity measure J adopted by the Apple 2019 MRQA paper, which can be expressed as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

for two word sets A and B. However, we ended up using the `Generalized Jaccard` [15] because it is more robust to misspelling - a rather frequent situation in our case due to the subword-level embedding. By default, `Generalized Jaccard` uses the Jaro similarity function, which is defined as:

$$sim_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$$

where $|s_i|$ is the length of string i, m is the number of matching characters and t is half the number of transpositions.

To ensure the quality of the augmented samples, we investigated the distribution of `Generalized Jaccard` scores between the estimated answer spans and the original gold answers (see Figure 2), and we decided to filter out examples with scores below 0.65. In Table 2, the post-Jaccard filtering sample size column shows that at most $1/3$ examples were filtered out, leaving us with enough examples for finetuning.

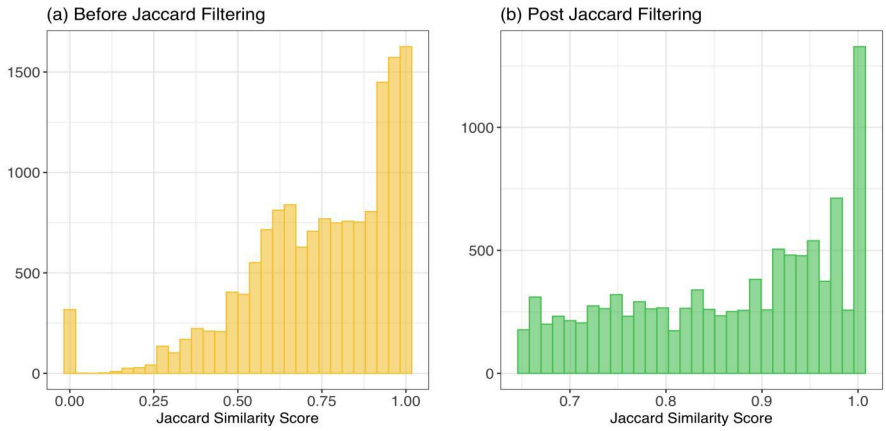

Figure 2: Before and Post Filtering Jaccard Score Distribution

4 Experiments

4.1 Data

We used two sets of data to complete the project. The first set of data is provided by the course staff, where the *ind* preprocessed datasets, SQuAD, NewsQA, and Natural Questions, were used to train the distilled version of BERT as our baseline model. We used the *ood* datasets train/dev splits as part of our finetuning process. The second set of data was the English-Spanish TED talks dataset used to train our NMT system, as described in section 3.3.

Table 2: QA Datasets Overview

Dataset	Train	Dev	Test
ind			
SQuAD [3]	50000	10,507	-
NewsQA [16]	50000	4,212	-
Natural Questions [17]	50000	12,836	-
ood			
DuoRC [18]	127	126	1248
RACE [19]	127	128	419
RelationExtraction [20]	127	128	2693

4.2 Evaluation method

We measured the QA task performance based on two evaluation metrics: Exact Match (EM) Score and F1 Score. The EM score measures the number of answers that are exactly correct, and the F1

Table 3: Back-translation Result

Experiment Configuration	Post-backtrans Sample Size	Post-Jaccard Filtering Sample Size	Query BLEU	Context BLEU
		ind		
NMT beam 1	14999	10231	24.07	18.87
NMT beam 5	14857	10546	28.43	25.07
NMT beam 1, no <unk>	14999	10232	25.07	23.05
NMT beam 5, no <unk>	14857	10553	28.55	28.17
Transformer DE	10000	9240	47.04	58.91
Transformer RU	10000	8827	36.55	50.33
		ood		
NMT beam 1	381	261	27.50	26.55
NMT beam 5	380	293	31.66	29.99
NMT beam 1, no <unk>	381	260	28.18	29.07
NMT beam 5, no <unk>	380	293	31.74	32.10
Transformer DE	381	366	52.87	56.46
Transformer RU	381	347	39.72	44.60

score captures the harmonic mean of precision and recall to measure whether the chosen answer are actually part of the true answer.

4.3 Experimental details

4.3.1 Beam size

Motivated by the research conducted by Edunov et al. [8] who showed that the searching approach used in the NMT decoding step can significantly influence the translation performance, we experimented with two searching approaches: greedy search and beam search with size of 5. We would like to investigate how different beam sizes could affect our back-translation results and thus affect the downstream QA task performance.

4.3.2 Replacement of <unk> tokens

The back-translated queries and context often output <unk> tokens. These <unk> tokens, however, represent different words and may convey different meaning across sentences. To prevent the QA model from being distracted to predict the <unk> tokens, we did a version of the back-translated query and context, where the <unk> tokens were replaced with empty spaces to test whether this could improve the QA learning.

4.3.3 Back-translation

In total we generated six augmented dataset using variants of the back-translation model including: (1) NMT model with greedy search (beam size = 1), (2) NMT model with beam search (beam size = 5), (3) NMT model with greedy search (beam size = 1) and no <unk>, (4) NMT model with beam search (beam size = 5) and no <unk>, (5) Transformer DE and (6) Transformer RU.

The training time for each NMT models was approximately 8 hours in total. The translation time was roughly 20 hours for greedy search NMT and roughly 28 hours for beam-5 NMT. For Transformer DE and Transformer RU, the translation time was around 36 hours.

For each augmented dataset, we compute the BLEU scores for both the query and the context (measured at the sentence-level) before vs. after the paraphrasing as a measure of the translation quality. The results are summarized in Table 3.

4.3.4 Finetuning

For each of the six variants, we loaded the baseline checkpoint and finetuned the QA model on both the augmented *ind* (roughly 10K examples) and *ood* data (roughly 380 examples). Evaluated on

the validation set for every 100 steps, each finetuning process took approximately half hour. The hyperparameters were set to default with batch size = 16, lr = 3e-05, and number of epoch = 3.

In particular, both models with no <unk> token performed much worse than their counterparts. We suspected this might be because the <unk> token functions as a sentinel which marks the length and position of the missing subword. Once replaced, the model cannot distinguish such placeholders from a regular space which represents the separation of the words. As a result, the meaning and grammatical structure of the sentence become fragmented. In all subsequent experiments, we dropped these 2 variants with the <unk> token replaced.

In addition to the individual models, we also tried ensembling NMT beam 1, NMT beam 5, Transformer DE and Transformer RU. Since these augmented data were back-translated from the same set of training data, we sampled 1/4 from each in our ensemble to avoid finetuning to very similar examples repeatedly.

To improve models' performance on *ood* data, we first validated the model on the *ood* validation set. Given that we also augmented some *ind* training examples, it is reasonable to validate also on the *ind* validation set. It is worth noting that the *ind* validation set is much larger than and is likely to dominate over the *ood* validation set during the finetuning validation process. Therefore, we chose to sample 1.4% of the original size of the *ind* validation data and combined with all of the *ood* validation data so that the *ind* validation size (27,555 $\times$ 1.4% $\approx$ 386 examples) roughly equal to *ood* validation size (382 examples). Introducing the *ind* validation could presumably act as a from of stabilization and prevent the model from choosing the best checkpoint solely based on the noise from the rather limited *ood* validation data.

4.4 Results

Our best model is obtained by finetuning on the augmented data generated by Transformer RU only and validating on the union of 1.4% *ind* and all *ood* validation data. It achieved 59.86 F1 and 41.42 EM on the test set and 51.28 F1 and 35.86 EM on the development set, ranked respectively at the 27th and 26th position on the leaderboard by the time of writing. Unlike what we have hoped for, ensembling did not help with the QA performance. The ensemble model achieved 58.92 F1 score and 40.46 EM on the test set and 48.26 F1 score and 32.98 EM on the development set, which we suspect is because the NMT systems lag well behind the Transformer. The QA performance of other experiments on the development set can be found in Table 4.

Table 4: QA Task Result on Development Set

	Validate on *ood* Only		**Validate on *ind* + *ood***	
	F1	**EM**	**F1**	**EM**
NMT beam 1	49.09	34.82	48.12	32.98
NMT beam 5	49.98	34.29	48.44	33.25
NMT beam 1, no <unk>	48.10	32.72	-	-
NMT beam 5, no <unk>	49.48	32.98	-	-
Transformer DE	50.51	36.65	48.09	32.98
Transformer RU	51.28	35.86	**51.28**	**35.86**
Ensemble	50.50	36.65	48.26	32.98

1.Bolded model was used for submission to test leaderboard
2.Ensemble model used the augmented training set which was sampled from augmented data generated by NMT beam 1, NMT beam 5, Transformer DE and Transformer RU with equal weights

5 Analysis

5.1 Comparison of back translation models

As demonstrated by Table 3, Transformer DE and RU outperform all NMT models in terms of both the percent of translatable queries and the translation quality (as measured by the BLEU score). This aligns with our expectation since the transformer architecture, equipped with multi-headed attention

and position representation, is inherently more powerful than the 2-layer LSTM network. The training corpus used by both transformers also come from a similar data source as the QA dataset and is much larger in size, as noted in section 3.4.

By inspecting the augmented datasets generated from the back-translation models, we note that transformers are also better at preserving the grammatical and logical structure of the sentence. The NMT systems, nonetheless, fail to translate most named entities like person, location and dates, which are often the gold answers of a question (see example below):

> *Context:* ***Lionel Messi*** *continued his remarkable scoring streak ...*
>
> *NMT systems:* ***Linel Messi*** *continued his remarkable score of scored with ...*
>
> *Transformers:* ***Lionel Messi*** *continued his remarkable scoring streak ...*

5.2 Effect of BLEU score

As discussed above, a low BLEU score as in the case of NMT beam 1 (around 27.0) and NMT beam 5 (around 30.0) might signal a poor translation quality, which ultimately lad to lower performance on the QA task. In fact, beam 1 with no `<unk>` token performed worse than the baseline when finetuned on *ood* validation data only, while the other 3 variants only slightly outperformed the baseline.

However, a high BLEU score isn't necessarily better: If the augmented examples are too similar to the original examples, they don't add much value to the learning of the QA model. This is likely what happened with the Transformer DE, which achieved a BLEU score of above 50 on both the query and the context, but only had fair performance on the QA task.

Overall, there seems to be a sweet spot in between where the model is able to preserve the translation quality while introducing a fair amount of noise to the data. In particular, the Transformer RU, with a query BLEU of 39.72 and a context BLEU of 44.60 achieved the best development set performance among all models.

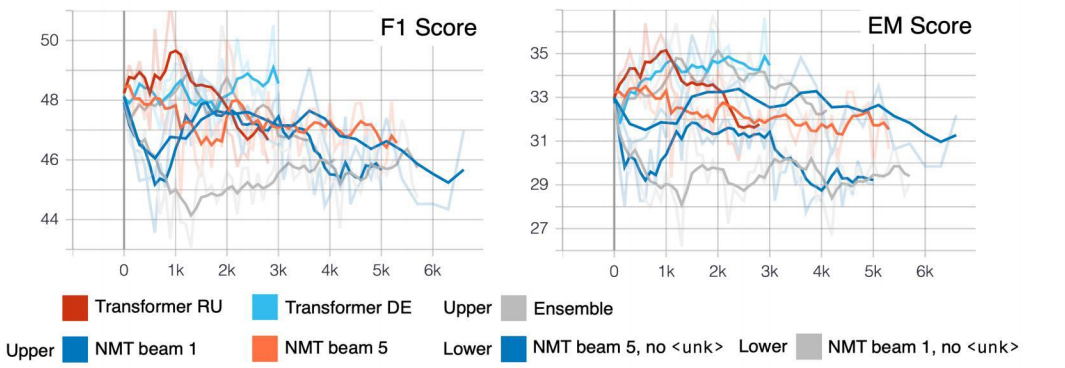

Figure 3: Model Performance on Development Set

5.3 Choice of pivot language

While we attribute Transformer RU's outstanding performance mainly to its well-ranged BLEU score, we cannot disentangle the effect of the choice of the pivot language.

Among all languages investigated, German is the most similar to English in that they not only both belong to the Germanic linguistic family, but also use the same 26-letter alphabet and assume a similar grammatical structure. Although the Spanish orthography includes 4 extra letters "ch", "ll", "ñ", and "rr" compared to the English alphabet, the spelling of many words are still similar to their English counterparts. Russian, however, uses the Cyrillic alphabet consisting of 10 vowels, 21 consonants, and 2 signs. It also has a more flexible word order: Although the SUBJECT-VERB-OBJECT structure is considered dominant, the SUBJECT-OBJECT-VERB is also valid. Except for the foreign words, the spelling basic Russian vocabulary are vastly different from the other languages considered in this project, as demonstrated by the example below:

English: *person*

German: *person*

Spanish: *persona*

Russian: человек *(chelovek)*

Therefore, it could be the case that the large linguistic distance between Russian and English compared to German or Spanish, making it a better pivot language choice because it can sufficiently disturb the inputs and thus contribute to the overall robustness of the QA model.

5.4 Error analysis

To understand what our best model, Transformer RU manages to solve and its limitations, we analyzed and compared performance of all models under different scenarios.

Question: What was the score in semifinal game of Wales-Ireland?
Context: ... Les Bleus avenged their 2007 semi defeat by the English on home soil with a 19-12 victory in Auckland, setting up a last-four clash with Wales – who went through after beating Celtic neighbors Ireland 22-10 ...
Answer: 22-10
Transformer RU augmented model prediction: 22-10
Other model prediction: 19-12

Analysis: *As discussed in Section 5.1, the NMT models fail to translate most person names and numbers, which can be illustrated by the above example. Not only does Transformer RU successfully restore the right numbers, the QA model tuned on this augmented dataset also understands the inner logic of the sentence. Specifically, the QA model with Transformer RU augmentation correctly matches the scored points with the corresponding teams; 22-10 is the score for the game Wales-Ireland, while 19-12 is the score for the game England-France.*

Question: What is the writer's attitude toward madness?
Context: It seems that great artists and scientists often suffer from mental problems. Both Einstein and Dickens had mental illness ... bad or just difficult to understand, but their discoveries have improved the world we live in. It seems that a little creative madness is good for us all.
Answer: little creative madness is good for us all
All model predictions: good for us all

Analysis: *All models are able to capture some information in the true answer but lose nuance attributes and definitions. In this example, all models are able to output the correct attitude "good for us all" but fail to qualify the "madness" as "a little creative" in their predictions.*

6 Conclusion

In this paper, we implemented back translation as a method of data augmentation. By paraphrasing each query and context pair in the sampled in-domain and out-of-distribution training sets, we generated augmented datasets using one of the 6 variants including NMT beam 1 (with and without `<unk>` tokens), NMT beam 5 (with and without `<unk>` tokens), Transformer DE and Transformer RU, as well as an ensemble model with equal splits. Our best model, the Transformer RU, achieved 51.28 F1 and 35.86 EM on the development set and 59.86 F1 and 41.42 EM on the test set (respectively ranked 26th and 27th on the leaderboard). The results from our extensive set of experiments not only show that backtranslation helps boost the QA performance, but also that the backtranslation architecture and training corpus matter as they affect the translation quality. If given more time and computing resources, we would like to run experiments using the control variate method to isolate the effect of pivot language in backtranslation on the downstream task performance. If indeed the Transformer RU outperforms other variants due to the larger linguistic distance between Russian and English, then languages like Chinese may work even better because it uses the pitch of a phoneme to determine word meaning and does not use spaces to separate words.

References

[1] Zhucheng Tu Chris DuBois Shayne Longpre, Yi Lu. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. In *Machine Reading for Question Answering*, 2019.

[2] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks, 2020.

[3] Konstantin Lopyrev Percy Liang Pranav Rajpurkar, Jian Zhang. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[4] Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. Learning and evaluating general linguistic intelligence, 2019.

[5] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[7] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning, 2021.

[8] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale, 2018.

[9] Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. Investigating backtranslation in neural machine translation, 2018.

[10] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[11] Cs 224n winter 2020. `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/index.html`. Accessed: 2021-02-25.

[12] Christopher D. Manning Minh-Thang Luong. Stanford neural machine translation systems for spoken language domains, 2015.

[13] Cs 224n winter 2020 assignment 4. `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/assignments/a4.pdf`. Accessed: 2021-02-25.

[14] Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. Facebook fair's wmt19 news translation task submission, 2019.

[15] AnHai's Group, 2017.

[16] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset, 2017.

[17] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, 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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019.

[18] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension, 2018.

[19] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations, 2017.

[20] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics.