Increasing Robustness of DistilBERT QA System with Few Sample Finetuning and Data Augmentation

Stanford CS224N Default Project

Alina Chou
Department of Computer Science
Stanford University
alinac@stanford.edu

Anjali Sukhavasi
Department of Computer Science
Stanford University
asukhava@stanford.edu

Abstract

NLP systems often cannot accurately generalize information beyond their training domain because they learn superficial correlations between words rather than understanding their meaning. In order to build a model that demonstrates language understanding with minimal exposure to the domain, we must build a robust model that can quickly adapt to new domains. Our project attempts to build upon the existing DistilBERT model, a distilled, more manageable, scalable, and environmentally-friendly version of BERT, to increase language understanding by training and testing the model on a reading comprehension task. We leveraged few sample finetuning and data augmentation via back-translation and synonym replacement in our attempt to improve reading comprehension performance on out of domain data . While finetuning yielded promising results, we found that data augmentation produced negligible results compared to the baseline.

1 Key Information to include

- Mentor: Yian Zhang

2 Introduction

While humans can quickly learn and generalize new words to new contexts or domains by learning the true meaning of words rather than learning correlations between words, NLP systems often cannot accurately generalize information beyond their training domain because they learn superficial correlations between words rather than understanding their meaning. The mathematical structure behind common NLP models is rooted in the probability of the word(s) co-occuring within a context based on training data that contains those words in those contexts at a certain frequency. Therefore, it is difficult to teach a model to generalize beyond what it has learned during training. However, in the real world all users, and by extension, all user interactions are unique. Therefore, in order to competently serve the needs of users in the real world, a model's ability to understand language must extend beyond the words it has trained on or "knows" to language it has not seen before.

Our research seeks to specifically improve the existing DistilBERT model, a smaller, more energy and time-efficient BERT model with similarly high performance, which was created using knowledge distillation. In doing so, we hope to build a robust model that is both environmentally friendly and more scalable to smaller devices such as smartphones, where a competent language-understanding model could become more accessible to a broad majority of potential users.

The task we will test and train our model on is reading comprehension; that is, given a passage (also known as the context) and a question (also known as a query), the QA system must correctly

identify if the answer to the question is contained within the text, and if so, return the span of text that answers the question. In this manner, we can measure a model's ability to understand the text, and by extension, natural language in general. An example of an input is a passage on Tesla's life story and a question about where he set up his first lab; the correct output would be the excerpt from the passage detailing the location of Tesla's first lab.

We implemented few sample finetuning and data augmentation in an attempt to increase robustness on out-of-domain data. Few sample finetuning was implemented by adjusting hyperparameters to increase fewshot accuracy, which is the accuracy of a model based on a small sample size.

We implemented data augmentation in two separate ways: via back-translation using different pivot languages and via synonym substitution. data augmentation has shown to be incredibly successful in computer vision models and tasks, wherein a change in a single pixel of image data, or a shift in the orientation or coloring of the image mostly does not change the content of the image but strengthens the model's ability to handle variances in data. Conversely, a change in single word or in the structure of a sentence can greatly alter the implicit and explicit meanings of a sentence.

Data augmentation via back-translation is the practice of translating the source sentence to another language, referred to as the "pivot" language, and then translating the sentence back to the source language to increase the diversity of sentence structure and word usage in the training data. We study the effects of implementing data augmentation via back-translation using three pivot languages from different language families: Russian, German, and Chinese from the Slavic, Germanic, and Sino-Tibetan language families, respectively. In doing so, we seek to study the effect of using different languages with distinct synctactic and grammatical structures on augmenting data through back-translation. That is, sentences back-translated through German may employ completely different synctactic structures and word choices than words back-translated through Russian which may be different from those back-translated through Chinese. To these ends, data that is augmented via back-translation in one language may make the model more robust than using data that is augmented via another language. For example, since English is closer to German than it is to Chinese, data augmented via back-translation to Chinese may be more variable and produce more robust training than data augmented via back-translation to German.

Data augmentation via synonym substitution chooses words that are not stop words from the sentence and replaces each of them with a random synonym. Inspired by data augmentation in computer vision and similar to the data augmentation via back-translation method described above, this method increases the variability of the training data so as to avoid training the model on weak correlations, thereby increasing robustness.

3 Related Work

The DistilBERT research has shown that it is possible to reach similar performances on many downstream-tasks using significantly smaller language models that are pre-trained with knowledge distillation, resulting in lighter and faster models which are less computationally costly. When modeling human language with computers and applying deep learning methods to train models, one of the most important factor that needs to be considered is the environmental impact. To maintain high accuracy, the models require exceptionally large computational and energy resources and are costly to train and develop as a result.[1] DistilBERT seeks to reduce the carbon footprint and associated costs of training larger models, while maintaining high performance. Another relevant problem in NLP is making language models more accessible and scalable. Since the paper includes a mobile testing conducted on an iPhone, the authors sufficiently demonstrate that the model can perform well on mobile devices with smaller amounts of memory, and is more relevant for the general public. This reflects how powerful and accessible NLP models can become in the future.

While the DistilBERT paper mentions that the model is finetuned to increase accuracy compared to the original BERT model, implementing few sample finetuning in our research might help build robustness to OOD datasets of varying sizes by increasing fewshot accuracy to decrease volatility in response to smaller datasets. Few sample finetuning has been previously studied on the BERT model, but not on the DistilBERT model. [2] One of the motivations in initially implementing

few sample finetuning on BERT was that the large model did not seem able to adapt to smaller datasets. However, since the DistilBERT model is considerably smaller than the BERT model, we seek to study the effects of few sample finetuning on fewshot accuracy.

Data augmentation via synonym replacement has previously shown promising results, but was mostly tested on text classification tasks and we seek to study whether it has similarly promising results on the reading comprehension task.[3] Data augmentation via back-translation has been implemented (using French as the pivot language) on the reading comprehension task with subtly promising results.[4] Our project intends to study the effect of implementing different pivot languages from other language families to observe its effect on increasing robustness.

4 Approach

4.1 Baseline

The baseline model that is given to us for this default project finetunes DistilBERT on all training data. Our loss function is a sum of the cross-entropy loss or the negative log-likelihood loss for the start and end locations and is represented by $loss = -\log p_{start}(i) - \log p_{end}(j)$. The loss function is minimized during training using the AdamW optimizer. We implement the loss function in `train.py`.

4.2 Few Sample Finetuning

For few sample fine tuning, we explore the effect of important hyperparameters such as learning rates, number of gradient update steps, number of layers to freeze, and number of batches. We modify the values of each of the hyperparameters with ones that generate high performance in Zhang et al.'s research paper [2].

As for learning rates, ADAM rescales it by $\frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t}$, with $\beta_1, \beta_2 \in [0,1)$ representing the exponential decay rates for the moment estimates. We tune the value to maximize the evaluation scores, $5e-5$. For number of gradient update steps, we experiment with 3, the default value, as well as 4, experimenting whether more epochs will increase the robustness of the model. For number of layers, the paper suggests that with more layers, the performance will not necessarily increase. The best values for number of layers varies across datasets. We experimented with fine tuning on 4, 6, and 12 layers. Finally, for the number of batches, we tried batch size of 4, 8, and 32 aside from the default size of 16, following the hyperparameter setup of Lee et al [5].

4.3 Data Augmentation via Back-Translation and Synonym Substitution

For data augmentation, given an input of $x = (q, p)$ with a label y, an augmented example is generated to be $x' = (q', p')$, where q' and p' are paraphrases of q and p respectively. In order to obtain the paraphrases of q and p, we implement both back-translation (first translating the original language into a pivot language, then translate back to the original language to produce a paraphrased version) and synonym substitution (words in the language are replaced with their synonyms). As for back-translation, we use German, Russian, and Chinese as pivot languages chosen from different language families. For synonym substitution, we replace words in the original context and question with WordNet's synonyms.

5 Experiments

5.1 Data

Our QA model trains on three in-domain reading comprehension datasets - Natural Questions [6], NewsQA[7], and SQuAD[8]. Natural Questions consists of 307,000+ real queries to the Google search engine and an answer that is taken from the relevant Wikipedia page. NewsQA consists of 100,000+ crowdsourced questions and answers consisting of spans of news article text from CNN/Daily Mail. SQuAD is a reading comprehension dataset consisting of 100,000+ crowdsourced questions over a set of Wikipedia articles. The model is then evaluated on three out-of-domain

datasets - DuoRC[9], RACE[10], and RelationExtraction[11]. DuoRC uses movie descriptions from two sources to formulate 180,000+ question-answer pairs, ensuring minimal lexical overlap between the question and answer to test true understanding. RACE consists of roughly 100,000 question-answer pairs from English reading comprehension tests for middle and high school aged Chinese students, allowing us to compare the model's reading comprehension skills against expected human performance on the task. RelationExtraction compiles common knowledge-base relations based on natural-language questions, further testing whether the model is able to pick up on the relationships between subjects in the "reading" and correctly respond to questions on these relations.

5.2 Evaluation Method

We measure performance by EM and F1 scores [12]. F1 scores measure the precision of words chosen for the answer that are actually part of the answer. EM, or exact match, scores calculate the number of answers that are exactly correct with the same start and end index.

5.3 Experimental details

We experiment with few sample tuning and data augmentation via back-translation and synonym substitution. In this section, we will discuss the experiment details on both methods. For both approaches, we experiment with training our model on only in-domain dataset and training on both in-domain and out-of-domain datasets. Specifically, we train finetuned or augmented models on in-domain dataset first and further train the best performing model on out-of-domain dataset by reading weights from the previous best performing training. We initially experiment using a combination of augmented data and few sample finetuning in order to independently observe the effects of implementing one or the other as an improvement to our model. We then combine both implementations to observe the combined effect.

Few Sample Finetuning

In order to compare and obtain the best parameters for the model, we tune the hyperparameters learning rates, number of gradient update steps, number of layers to freeze, and number of batches one parameter at a time while keeping the rest constant. For learning rate, we run with the default value $3e-5$ as well as $5e-5$. For number of gradient update steps, we run with values of 3 (default value) and 4, as we think the model might benefit from having more epochs. For number of layers to freeze, we experiment with 4, 6 (default), and 12 layers. Finally, as for the batch size, we run with 4, 8, 16 and 32 batches as those are the best performing values according to [2].

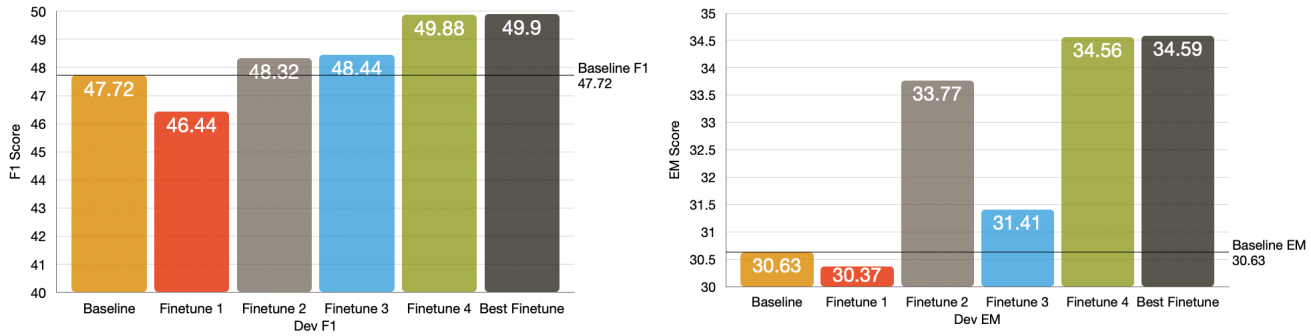
Data Augmentation via Back-Translation and Synonym Substitution

With data augmentation via back-translation, we hope to find whether the approach improves the robustness of the model and whether languages from different language families will influence the results. We experiment with German, Russian, and Chinese, as representations of the West Germanic (same language family as English), Slavic, and Sino-Tibetan language families. After translating both question and context into pivot languages, we then re-translate the pivot languages back to English, which serve as the new training data. On top of back-translation, we also experiment with substituting words with their synonyms, which also ensures equal length of the contexts and questions compared to the original data. We first experiment with using the original parameters as the baseline in order to extract the effects of data augmentation before ensemble the models and experiment with both finetuning and data augmentation.

5.4 Results

Few Sample Finetuning

Table 1 and Figure 1 demonstrates our dev set results after implementing few sample finetuning training on both only the in-domain dataset and in-domain + out-of-domain datasets. Variation indicates the key hyperparameters that we tune. As shown in the results table, our best performing model* from finetuning comes from using a batch size of 16 with 5e-5 learning rate. We evaluate



(a) F1 Score Comparison

(b) EM Score Comparison

Figure 1: Few Sample Finetuning Results Compared to Baseline Model

every 500 steps to save the best models. Using the best finetuning result from in-domain training set, we train the model on out-of-domain training set using weights saved from the in-domain training, which yields further improvements on both the F1 and the EM scores. We also notice with batch size of 32, the model converges the fastest during training – model^ uses minimal training time, converging in 20k steps as compared to the baseline taking 45k steps.

Experiment	Variation	Dev F1	Dev EM	Training Steps
Baseline	Batch size 16 lr 3e-5	47.72	30.63	45k
Finetune 1 (in-domain)	Batch size 4 lr 3e-5	46.44	30.37	146k
Finetune 2 (in-domain)	Batch size 8 lr 3e-5	48.32	33.77	90k
Finetune 3 (in-domain) ^	Batch size 32 lr 3e-5	48.44	31.41	20k
Finetune 4 (in-domain) *	Batch size 16 lr 5e-5	**49.88**	**34.56**	44k
Best Finetune (in-domain + out-of-domain)	Batch size 16 lr 5e-5	**49.90**	**34.59**	44k

Table 1: Few sample finetuning results.

Data Augmentation via Back-Translation and Synonym Substitution

Table 2 shows the back-translation models' dev set performance using the same parameters as the baseline training parameters. Back-translation across different language families generates similar results as the baseline performance (F1/EM 47.72/30.63), while synonym substitution produces little improvement (+0.08/+0.52) compared to the baseline. After further training the best data augmentation model (synonym substitution) on out-of-domain training set, we get dev set F1/EM of 48.28/32.20, which is +0.56/+1.57 improvement compared to the baseline.

Ensemble Model

Finally, we ensemble the model from finetuning and data augmentation and train on both in-domain and out-of-domain training sets, obtaining a dev set F1/EM score of **50.17 / 36.13**,

Experiment	Dev F1	Dev EM	Training Steps
Baseline	47.72	30.63	45k
Back-Translation (German)	47.71	30.62	45k
Back-Translation (Russian)	47.73	30.63	45k
Back-Translation (Chinese)	47.72	30.63	45k
Synonym Substitution	**47.80 (+0.08)**	**31.15 (+0.52)**	45k
Best Data Augmentation (in-domain + out-of-domain)	**48.28 (+0.56)**	**32.20 (+1.57)**	45k

Table 2: Data augmentation via back-translation and synonym substitution results.

which is an improvement of **+2.45 / +5.5** compared to the baseline, and a test set F1/EM score of **58.51 / 41.17**.

6 Analysis

Finetuning results show that the number of layers and learning rate will greatly affect the model's training as well as evaluation performances. Our model generates better results with an increase in learning rate as well as evaluation frequency. The results also demonstrate that the training batch size affects the training time. We notice that the number of training steps is reversely proportional to the batch size – number of training steps decreases linearly with an increase in batch size, as shown in Figure 1. But batch size 16 gives the optimal performance.

The training loss, F1 score, and EM score are approximately the same for the baseline model and the data augmentation model using back-translation (in German, Russian, and Chinese) and synonym substitution indicating little improvement from data augmentation. One potential reason that data augmentation via both back-translation and synonym substitution are not improving the model is that the generated sentences that are of the same meaning and, usually, sentence structure as the original sentence. While data augmentation via back-translation and synonym substitution have proven to improve NMT models that generate text, they might not be useful for our model that has to select the correct span of text for reading comprehension. The information that is fed into the training "answer" and "context" is too similar, which does not lead to a significant difference in answering questions given contexts by selecting spans of text.

However, the data also shows that synonym substitution generates slightly better results as compared to the baseline and via back-translation. An explanation for why synonym substitution performs better than back-translation is that synonym substitution preserves the original length of the text. The substitution does not cause the start and end indices for the context and question to shift, which ensures that the training data selects the correct span of text. Back-translation, on the other hand, might lead to shifting in the selected span of texts in training, which may impair any improvement to training leading to results that demonstrate negligible overall score improvement.

7 Conclusion

Our project builds a model that demonstrates language understanding with minimal exposure to the domain. It builds upon the existing DistilBERT model to increase language understanding by training and testing the model on a reading comprehension task. In our attempt to increase understanding, we leverage few sample finetuning and data augmentation via back-translation and synonym substitution. We train models on both the in-domain dataset and out-of-domain dataset. While finetuning yields promising results of F1/EM score 49.90/34.59 on the dev set, which is an improvement of +2.18/+3.96 compared to the baseline, we find that data augmentation produces negligible results compared to the baseline, with synonym substitution generating F1/EM score 48.28/32.20. The final ensemble model using both finetuning and data augmentation

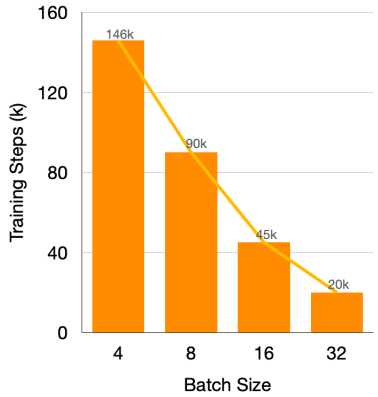


Figure 2: Batch size with training steps.

via synonym substitution trained on in-domain and out-of-domain datasets yields a F1/EM score of **50.17/36.13** on the dev set and **58.51/41.17** on the test set.

References

[1] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Association for Computational Linguistics (ACL)*, 2019.

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

[3] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 2019.

[4] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. 2018.

[5] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. In *ICLR*, 2020.

[6] 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. In *Association for Computational Linguistics (ACL)*, 2019.

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

[8] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *CoRR*, 2016.

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

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

[11] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *CoNLL*, 2017.

[12] Brad Girardeau. Question answering on the squad dataset. In *CS224N*.