

Building a Robust QA System Via Diverse Backtranslation

Stanford CS224N Default Project

Nikhil Raghuraman

Department of Computer Science
Stanford University
nikhilvr@stanford.edu

Pranav Sriram

Department of Computer Science
Stanford University
prsriram@stanford.edu

Abstract

While question answering (QA) systems have been an active topic of research in recent years, these models typically perform poorly on out-of-domain datasets. Thus, the goal for our project was to build a question answering system that is robust to distributional shift. Utilizing a pretrained DistilBERT model as our baseline, we tested two adaptation methods: backtranslation and few-sample fine-tuning. Backtranslation, which involves translating input data into an intermediate language before translating back to English, is a common data augmentation technique in many NLP tasks. We found that implementing standard backtranslation on out-of-domain training examples yielded significant increases in Exact Match (EM) and F1 scores over our baseline model. We compared these results to several modified backtranslation schemes including a schema where we combined backtranslation with few-sample fine-tuning. Ultimately, we found that combining backtranslation with few-sample fine-tuning did not improve performance over only backtranslation. Our best model achieved an EM of 42.225 and F1 of 59.162 on the test set, and an EM of 38.74 and F1 of 51.19 on the development set.

1 Key Information to include

- Mentor: Rui Yan
- External Collaborators (if you have any): None
- Sharing project: Not sharing project with another class

2 Introduction

Question answering tasks involving reading comprehension are a vital metric by which to evaluate natural language models. While transformer-based question answering systems have demonstrated human-level success on in-domain datasets, their performance suffers when tested on out-of-domain datasets [1]. Hence, the goal for our project was to build a question answering system that is robust to distributional shift and can be applied to more contexts.

We investigated whether backtranslation (translating an input training example into an intermediate language before translating back into English) and few-sample fine-tuning techniques would lead to a robust question answering system when used in conjunction with each other. This inquiry was important because these techniques in isolation have been shown to improve performance on some NLP tasks, but they have not been implemented together on robust QA tasks [2], [3]. Our hypothesis was that combining these different techniques into a single system would lead to improvements over a baseline model. Our novel contributions are as follows:

- We investigated a diverse range of backtranslation strategies, including using many languages as pivot languages, using only languages very different from English as pivot languages, and

translating a portion of a context into many languages before back into English. As far as we are aware, existing literature does not attempt so many forms of backtranslation in the pursuit of robust QA systems.

- We coupled backtranslation with techniques from [3], namely longer fine-tuning and re-initialization of pre-training layers.

We found that backtranslation with pivot languages similar to English yielded strong performance on the development set. Combining backtranslation with the few-sample fine-tuning technique in [3] did not improve performance.

3 Related Work

3.1 Question Answering

Since performance on reading comprehension question-answering tasks can be indicative of a natural language model’s overall understanding of human language, a rich body of research has emerged in recent years studying question answering. A variety of model architectures have demonstrated strong performance on question-answering tasks, including the Bidirectional Attention Flow Model (BiDAF), an LSTM-based architecture [4], and QANet, a convolution and self-attention-based model architecture [2].

Recent works have improved upon these baselines by using large, pre-trained transformer models. BERT, an early transformer model, achieved state-of-the-art performance on the SQuAD v1.1 and v2.0 datasets [5]. BERT’s question-answering baselines were quickly improved upon by models such as SpanBERT, which introduced a more challenging masked language modeling task for pre-training [6]. Recognizing the recent success of transformer-based models for question-answering tasks, we also leverage a pre-trained transformer for our system.

3.2 Robust Question Answering

Though transformer-based question-answering models have achieved strong performance when trained and tested on data from the same distribution, their performance on out-of-domain datasets is much worse [1]. This has led to an active area of research on robust question answering, which seeks to determine how best to build question-answering systems that are robust to a distributional shift and/or adversarial examples. Previous works in this area have leveraged data augmentation via backtranslation [2], [7], explicit negative sampling of no-answer segments during training [7], and adversarial training [8], [9]. For backtranslation specifically, there have notably been varied findings in the literature regarding the success of this approach. [2] found backtranslation helpful, while [7] found it added no substantial gains. For this reason, we explore backtranslation extensively in our work to understand its efficacy.

Furthermore, a number of promising approaches have emerged for learning from a limited number of examples in domains other than the primary, in-domain dataset. For example, [3] suggested fine-tuning BERT models for more iterations and re-initializing the top pre-trained layers that are most specialized to the pre-training task. [10] noticed performance improvements after continuing to train the masked language modeling pre-training objective even on unlabeled data from their downstream tasks. To our knowledge, though, none of these approaches has been considered yet for building robust question answering systems. To address this gap, we seek to consider these and other fewshot-learning techniques in our work to determine if they apply to question answering.

4 Approach

We explored several approaches for training a robust question-answering system. For all approaches, we first fine-tuned a pre-trained DistilBERT model on a large in-domain dataset. We then performed a final round of fine-tuning on our small out-of-domain dataset. We hypothesized that separating our training for both in-domain and out-of-domain data would allow our model to first develop strong question-answering performance (from the in-domain data) and then adapt completely to the out-of-domain data. Below, we describe our model architecture, training details, and adaption approaches in more detail.

4.1 Model Architecture

Noting the success of pre-trained transformer models for downstream tasks such as question answering [5], [6], we utilized a pre-trained DistilBERT model for our task [11]. DistilBERT has a very similar architecture to the original BERT model, albeit with half as many layers. Additionally, DistilBERT was pre-trained using a knowledge distillation technique [12], [13] in which knowledge from a large transformer model is compressed into the smaller DistilBERT architecture. Simultaneously, DistilBERT was trained on the traditional masked language model loss typical of transformer models. We refer the reader to [11] for further details.

4.2 Loss Function

We performed our fine-tuning using the DistilBertForQuestionAnswering pre-trained model available on huggingface.co. For every token in the input to the model, the model outputs two logits for the likelihood of that token being the start and end of the correct answer span. The DistilBertForQuestionAnswering provides the following loss function:

$$J = -\log p_{start}(i) - \log p_{end}(j) \quad (1)$$

In 1, p_{start} and p_{end} are functions representing start and end probabilities for all tokens. i is the ground-truth start token for the answer span and j is the ground-truth end-token. Restated, this equation is a summed cross-entropy loss for the start and end probabilities.

4.3 Baseline

We utilized a DistilBERT model fine-tuned first on in-domain data and then out-of-domain data as our baseline.

4.4 Adaptation Approaches

4.4.1 Data Augmentation using Backtranslation

Our first approach was to perform heavy backtranslation on the limited out-of-domain data in order to generate paraphrases and vastly expand our training data size, [2] and [7] previously explored backtranslation for robust question answering, though [2] found it beneficial for performance and [7] found the performance benefits to be minimal. Because of this lack of consensus, we were interested to see whether we could achieve performance gains using backtranslation.

Our backtranslation technique involved the following procedure. We first split every training context into 3 portions: sentences before the answer, the answer sentence, and sentences after. We independently backtranslated each portion with probability $p = 0.9$ by randomly selecting one of several languages (depending upon the experiment; see section 5) as a pivot language. Doing so allowed us to increase our out-of-domain training data size.

One natural roadblock with this approach is that the answer in the backtranslated context may have different verbiage or appear in a different location as compared to the original context. To mitigate this, we computed character-level bigrams of both the original answer and the backtranslated answer sentence. Using these bigrams, we computed Jaccard similarity scores between spans of tokens and the original answer. Below, we define the Jaccard similarity calculation. Let A be the set of bigrams in the span of original text and B the bigrams in the span of answer text. The Jaccard similarity score [14] is defined as follows:

$$J = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

After calculating Jaccard scores for all possible spans in the backtranslated answer sentence, we selected the span of text with the highest Jaccard score. If no span had a score of at least 0.45, we discarded the entire backtranslated context. Both [2] and [7] utilized character-level bigrams to determine the new answer span, and [7] utilized a Jaccard similarity score, which is what led us to this approach.

Though [2] and [7] both explored backtranslation for question answering, both utilized only a single pivot language. In this work, we undertake a more thorough exploration of different forms of backtranslation, including selecting among many languages similar to English for translation, selecting among languages that are very different from English, translating into multiple languages, and translating the question in addition to the context. These different experiments are a novel contribution to the area of robust question answering. Our hypothesis was that more diverse training data through stronger backtranslation with many languages helps, as long as the data is still equally valid. We describe all experiments and results in more detail in section 5.

We implemented all backtranslation code independently. We also utilized nltk library [15] to split contexts into sentences and the BackTranslation Pypi library for backtranslation.

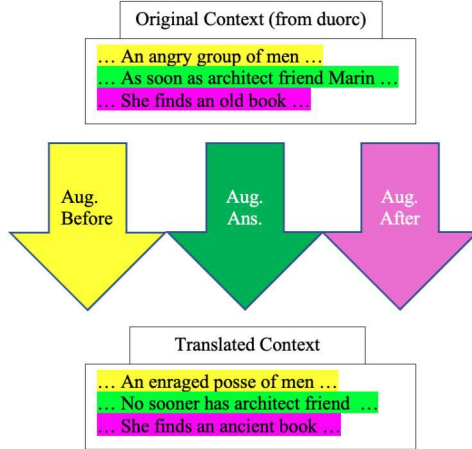


Figure 1: An example of a back-translated output. The context before the answer, the answer sentence, and the context after the answer are all back-translated independently. In all, there are up to three translations per context.

4.4.2 Selected Reinitialization of DistilBERT Layers

Following the hypothesis posed by [3] that higher layers of a BERT pre-trained model closer to the final output are overly specialized to the pre-training task (in our case, the masked language model task), we decided to explore deliberately re-initializing the top layer of our DistilBERT model to allow our model to more easily specialize to question answering. [3] explored re-initializing from 1 to 6 of the top layers of BERT. Since the DistilBERT model has fewer layers than BERT, we re-initialized only the topmost layer. To do so, we set bias terms to 0 and randomly initialized weight terms using the distribution $N(0, 0.02^2)$, as in [3]. We report the results for this technique in section 5.

4.4.3 Extended Fine-tuning Time

Finally, we attempted to fine-tune our DistilBERT model for longer on the in-domain dataset. We originally trained our baseline model for 3 epochs on the in-domain dataset. According to [3], this closely adheres to widespread wisdom on how best to fine-tune BERT models. However, [3] found that deviating from this wisdom and simply fine-tuning for longer leads to performance improvements. As a result, we attempted an experiment in which we fine-tuned on in-domain data for four epochs.

5 Experiments

We conducted nine experiments, each of which fine-tuned our pre-trained DistilBERT model [11].

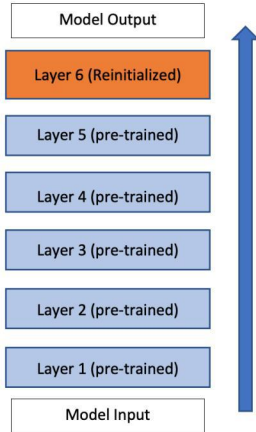


Figure 2: A diagram of our re-initialization process. We re-initialized the layer of the pre-trained model closest to the output.

5.1 Backtranslation Experiments

We conducted three preliminary experiments. Experiment 1, the provided baseline, consisted of DistilBERT fine-tuned on only in-domain data before being evaluated on an out-of-domain dataset. Experiment 2 consisted of the baseline model with additional fine-tuning epochs on only the out-of-domain train data. Experiment 3 consisted of the baseline model fine-tuned on both the original out-of-domain data and backtranslated versions of our out-of-domain data. In Experiment 3, for each context, our model randomly selected an Indo-European language relatively similar to English as a pivot language. The possible pivot languages were French, German, Spanish, Dutch, Italian, Russian, Swedish, and Norwegian. In all, we generated 10 backtranslated examples for each existing out-of-domain example and inserted them into our existing out-of-domain dataset. Moving forward, we refer to Experiment 1 as our baseline and Experiment 3 as our backtranslate baseline.

Afterwards, we conducted four further experiments exploring different backtranslation variations. Experiment 4 increased the number of backtranslated examples generated from each out-of-domain input example. This hyperparameter was increased from 10 to 30. In Experiment 5, we backtranslated all questions (in addition to the contexts).

For Experiment 6, we tested two-step backtranslation. Specifically, we first backtranslated an input example portion using a random pivot language. We then backtranslated again with another random pivot language. The output of this second backtranslation was added to our dataset. Each input context had a probability $p = 0.9$ of being two-step backtranslated.

Finally, for Experiment 7, we tested backtranslation with pivot languages linguistically far/different from English (all previous experiments used only the Indo-European languages). The languages utilized were Chinese, Japanese, Arabic, Turkish, Hungarian, Korean, Hebrew, and Tamil. All of these languages are outside the Indo-European language family. Additionally, the US State Department considers all eight of these language to be Category III (languages with significant linguistic and/or cultural differences from English) or Category IV languages (languages which are exceptionally difficult for native English speakers) [16].

We conducted Experiments 4 through 7 with the hypothesis that increased out-of-domain dataset diversity would lead to a more robust QA system. For example, we hypothesized that translating into languages very different from English would lead to that were substantially different from the original contexts. However, we were also interested to see if any of these experiments obscured the original meaning of example contexts and worsened performance.

5.2 Experiments Inspired by Fewshot Learning Approaches

Lastly, we performed a few experiments inspired by [3]. Rather than perform these experiments in isolation, we attempted them using our most successful backtranslated dataset in which we generated

30 backtranslated examples per original example on Indo-European languages. For Experiment 8, we re-initialized the top layer of our pre-trained DistilBERT model before undergoing our two fine-tuning rounds on our in-domain data and cached backtranslated data. For Experiment 9, we fine-tuned our DistilBERT model for 4 epochs rather than 3 on the in-domain data before fine-tuning on out-of-domain data.

5.3 Data

To evaluate performance, we fine-tuned a DistilBERT transformer on 3 large in-domain training datasets and 3 small out-of-domain training datasets. The system was then evaluated on 3 out-of-domain dev and test datasets. The input to our model was a context-question pair, and the output was the start and end of the answer from the context.

Experiment 1 was trained on the `indomain_train` folder (which consisted of the `nat_questions`, `newsqa`, and `squad` datasets). Each of these three datasets consisted of 50,000 train examples. Experiment 2 utilized the trained model from experiment 1 and fine-tuned the model on the `oodomain_train` folder (which consisted of data from the `duorc`, `race`, and `relation_extraction` datasets). These three datasets each consisted of 127 train examples. Finally, experiment 3 utilized the trained model from experiment 1 and fine-tuned on our augmented (backtranslated) `oodomain_train` folder. For backtranslation, a multiplier factor of 10 was used meaning that there were $(10 + 1)(127) = 1397$ train examples for each out-of-domain dataset.

For experiment 4, we used a multiplier factor of 30 for backtranslation. Thus, this resulted in $(30 + 1)(127) = 3937$ train examples for each out-of-domain dataset. Experiments 5 through 7 used the same number of examples as Experiment 3. Experiments 8 and 9 reused the cached backtranslated examples from Experiment 4.

5.4 Evaluation method

For all of our experiments, our evaluation metrics were EM and F1 scores. All experiments were evaluated on the `oodomain_val` folder (which contained 126 examples from DuoRC, 128 examples from RACE, and 128 examples from the Relation_Extraction datasets).

5.5 Experimental details

In all of our experiments, we utilized an Adam Optimizer with weight decay with a batch size of 16 and a learning rate $\alpha = 3 * 10^{-5}$. For experiment 1 (baseline), we fine-tuned for 3 epochs on our in-domain train dataset (which took close to 18 hours). For experiment 2, we fine-tuned the model from experiment 1 for 10 additional epochs on the out-of-domain train dataset. Finally, for experiment 3, we backtranslated 10 new examples for each out-of-domain train example. We then fine-tuned on this combined dataset for 12 epochs. Notably, we utilized a different number of epochs for experiments 2 and 3 because of the smaller number of examples for experiment 2. For each experiment, we retained the checkpoint with the highest validation F1 score. We chose these training lengths by observing the points at which the model started to overfit to the training data.

Experiment 4 was fine-tuned on out-of-domain data for 50 epochs, and the remaining were trained for 20 epochs. However, we ended these experiments early once we noticed the model had begun to overfit to the training data.

In every experiment, we retained the checkpoint with the highest F1 score on the validation set.

5.6 Results

See our experimental results below. Based on these results, we decided to submit our model from Experiment 4 to the RobustQA test leaderboard. Experiment 4’s strong performance in both EM and F1 score suggest that more backtranslation as opposed to less is generally helpful. We attained an EM score of 42.225 and a F1 score of 59.162 with this model on the test set.

Our results somewhat lined up with our expectations. Like [2], we found that backtranslating questions hurts performance. Backtranslating questions seemed to obscure critical question meaning information leading to worse results. Interestingly, while backtranslating with "harder" languages

Experiment #	EM Score	F1 Score
Experiment 1 (Baseline)	31.68	47.10
Experiment 2 (Baseline + Finetune)	35.08	49.42
Experiment 3 (Baseline + 10x Backtranslate + Finetune)	38.74	50.79
Experiment 4 (Baseline + 30x Backtranslate + Finetune)	38.74	51.19
Experiment 5 (Baseline + Question Backtranslate + Finetune)	38.48	50.14
Experiment 6 (Baseline + Two-Step Backtranslate + Finetune)	37.70	50.97
Experiment 7 (Baseline + Hard Language Backtranslate + Finetune)	37.96	51.07
Experiment 8 (Re-initialized DistilBERT + 30x Backtranslate)	38.22	49.99
Experiment 9 (DistilBERT + Longer finetuning + 30x Backtranslate)	38.48	51.21

resulted in a higher F1 score than our backtranslate baseline (experiment 3), it also decreased our EM score. We hypothesize this is because "harder" languages may result in the loss of more linguistic information due to imperfect translations, thus decreasing the probability of getting an exact match. However, the increased variation due to translations may help increase our model's overall performance on new data, as captured by our F1 score.

Two-step backtranslate had a worse EM score than our backtranslate baseline. This is likely because too much meaning is lost in the successive backtranslations leading to the output example to become functionally useless and mainly serve as noise for the model as opposed to useful augmented data.

Finally, we were surprised that Experiments 8 and 9 lead to minimal or no improvements. We believe that Experiment 8 was unsuccessful due to the DistilBERT containing fewer total layers than the BERT model used by [3]. Since our DistilBERT model was altogether smaller, the top layer of the model may have contained crucial and transferrable linguistic information, rather than only parameters highly specialized for the pre-training task. Lastly, Experiment 9's lack of success suggests that three epochs actually were sufficient for fine-tuning our DistilBERT model. Again, this is likely because our DistilBERT model was smaller than the BERT model used by [3] and hence required fewer epochs to train fully.

6 Analysis

We looked at predictions where our baseline and/or final model outputted incorrect answers. Below, we analyze a few of these examples:

6.1 Complex Multi-Answer Questions

- **Relevant Context:** "... Pete used to spend his holidays travelling the world, visiting the pyramids in Egypt or scuba diving in the Caribbean. Nowadays he prefers to spend his holidays and weekends making his house look more beautiful. Like hundreds of thousands of other British people, he has discovered the joy of DIY (Do It Yourself), which means if there are any things that need fixing around the house, he will try to do the job himself. As he showed me the new kitchen he put together by himself and the newly painted walls, I asked Pete where he got his inspiration from. ... I guess it is not really surprising that DIY programs are so popular. Two common sayings in Britain- 'an Englishman's home is his castle' and 'there's no place like home'-show how important our houses are to us."
- **Question:** Which might NOT be shown in a DIY program on TV?
- **Baseline Answer (Exp. 1):** new kitchen
- **Our Answer (Exp. 4):** how important our houses
- **True Answer:** scuba diving
- **Analysis:** This question is extremely broad and doesn't have a clear correct answer. While the true answer in this case was "scuba diving," there are a number of other valid answers such as "pyramids." While our baseline model's answer is completely incorrect (a new kitchen could definitely appear on a TV DIY program), our model gives an answer that doesn't make sense either. The extreme open-ended nature of the question likely makes it difficult for both our baseline and backtranslated model.

6.2 Domain-Based Structure

- **Relevant Context:** "... Afterwards, the four's idyll is interrupted by appearance of a Mexican sharpshooter called Chaco (Tomas Milian). Despite initial suspicious, he is welcomed into the group. Chaco displays his shooting skills by shooting rabbits and ducks for their meal for the night. ..."
- **Question:** What is the name of Mexican gunman?
- **Baseline Answer (Exp. 1):** Tomas Milian
- **Our Answer (Exp. 4):** Chaco
- **True Answer:** Chaco
- **Analysis:** In movie reviews, following the first reference to a character, it's very common to include the corresponding actor's name in parenthesis. The baseline has not picked up on this structural information so it appears to get confused and answer with the actor's name instead of the character name. In contrast, our model which has been fine-tuned on some examples from DuoRC seems to have picked up on this underlying structure.

We were also interested in comparing the performance of our baseline and Experiment 4 models against our validation datasets to determine if there are any discrepancies in performance. The below table contains our results:

Experiment #	Duorc EM/F1	Race EM/F1	Relation EM/F1
Experiment 1 (Baseline)	33.33/40.31	23.44/36.76	38.28/64.12
Experiment 4 (Backtranslate)	33.33/39.68	24.22/36.58	58.59/77.16

From this table, we can see that backtranslation led to minimal or no improvements on the duorc and race datasets. However, we notice that it leads to *substantial* improvements on the relation extraction dataset. This leads us to believe that backtranslation as a data augmentation technique is only useful for certain kinds of data. In particular, relation extraction is a question-answering dataset that involves identifying the relationship between different entities given a very short context. Since relationship extraction might depend heavily on word ordering and prepositional word choice, we hypothesize that seeing many paraphrases of a given context is very useful for question answering on the relation extraction dataset. On the other hand, the other datasets, which require higher-level understanding of the entire passages (e.g. race might ask for an overall summary of a passage), do not benefit as much from specific word choice and word ordering diversity.

7 Conclusion

We found that backtranslation is quite effective on QA robustness tasks. With that said, it is important to acknowledge that our experiments were only run on a single small pre-trained transformer model. Thus, it is not guaranteed all of these findings will generalize to other transformer architectures. Furthermore, as mentioned in our analysis, backtranslation only significantly improved results on the relation extraction dataset. Thus, there are clearly limitations surrounding the types of datasets backtranslation works well on.

Increasing the multiplier of our backtranslate operation appears to generally help performance. At the same time, interestingly, two-step backtranslation did not improve results compared to our backtranslate baseline. Similar to the results found in [2], backtranslating questions also does not appear promising for QA tasks.

A critical finding we found was that combining backtranslation with few-sample fine-tuning approaches did not improve performance. This may be due to the fact we utilized a relatively small transformer model, meaning each layer of the model wasn't very specialized.

Backtranslation in languages different from English appears to increase F1 scores but decrease EM scores. We encourage future research to further explore performance of backtranslation on a variety of different languages not just those closely related to English. Future work in this area could involve testing which specific language families are the most promising as pivot languages.

References

- [1] Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online, November 2020. Association for Computational Linguistics.
- [2] 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.
- [3] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning, 2021.
- [4] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension, 2018.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [6] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans, 2020.
- [7] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering, 2019.
- [8] Ziqing Yang, Yiming Cui, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. Improving machine reading comprehension via adversarial training, 2019.
- [9] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training, 2019.
- [10] 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.
- [11] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.
- [12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.
- [13] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’06, page 535–541, New York, NY, USA, 2006. Association for Computing Machinery.
- [14] Wikipedia contributors. Jaccard index — Wikipedia, the free encyclopedia, 2021. [Online; accessed 18-March-2021].
- [15] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O’Reilly Media Inc., 2009.
- [16] U.S. Foreign Service Institute. Foreign language training, 2021.