Building a Robust QA system with Data Augmentation

Stanford CS224N Default Project

Trinity Donohugh
Department of Computer Science
Stanford University
tringd@stanford.edu

Robert Hu
Department of Computer Science
Stanford University
roberthu@stanford.edu

Abstract

Pre-trained neural models such as our baseline model fine-tuned on a BERT based pre-trained transformer to perform nature language question and answering problems usually show high levels of accuracy with in-context data, but often display a lack of robustness with out-of-context data. We hypothesize that this issue is not primarily caused by the pre-trained model's limitations, but rather by the lack of diverse training data that might convey important contextual information in the fine-tuning stage. We explore several methods to augment standard training data with syntactically informative data, generated by randomly replacing the grammatical tense of data, removing words associated with gender, race, or economic means, and only replacing question sentences with synonym words from a lexicon of words. We found that the augmentation method that performed the best was changing the grammer of more and one word in every question. Although it only made less than 1 point increase in the F1 and EM scores, we believe that if we also applied this method to the context and answers training data we would be able to see even more significant improvements. We were also surprised the method of removing associates with gender, race, or economic performed relatively well given that we removed a lot of words from the dataset.

1 Introduction

Our hypothesis going into this project for why question answering systems struggle with out-of-domain datasets is that neural models learn on one type of language syntax and thus struggle to see correlations between questions asking about the same context/ topic but written in another grammatical tense or language syntax. Therefore, our goal for this project was to train a fully-functional neural baseline model and experiment with the idea of robustness via different methods of data augmentation.

From our experimentation with three different methods of data augmentation: changing questions grammar tense, removing context words that are related to gender, ethnicity or class, and replacing questions words with synonym words, we found that changing questions grammar tense was the most effective method, although did not make large tangible improvements, it did show promise. Thus, it pointed us in the direction of changing more words in the question to it's past tense structure and it showed even more improvements. Thus, we think that a future additional improvement that would help make huge improvements may be to change the questions to all past tense but then changing the contexts to all present tense or even changing every word to different tenses randomly.

Our reasoning behind the grammar method is because words from different languages translated between languages will often times "mess up" grammar since grammar is different between different languages. Thus, we believed that by changing the grammar we change the syntax of the sentence, allowing our model to be able to understand and respond to questions in unfamiliar syntactical form and still be able to draw correlations between the out of domain question and the original training data.

Our reasoning behind removing words associated with gender, class, and ethnicity was because we believe that some of the correlations the model learn to draw are between certain words that are often associated together due to societal norms. Thus, we believed that by removing them in the training data, our model would be able to draw correlations between out of domain questions with training examples shown. Since we forced our model to think outside of societal norms we believed it would be able to find interesting, and nuanced correlations that are foreign to us. This method ended up doing better than we had hoped after we added more words to our collection of "social class" words. Although, still not the most optimised, we believe that if we asked it an even more random set of questions, it would have out performed the traditional grammar method.

Lastly, our reasoning behind the synonyms method was because we had seen a couple papers use this method to augment their data with promising results. Thus, we believed that it would be a good benchmark for us. However, this method performed poorly for us, likely due to the fact that we did not replace every word and our synonyms library was a little out of date and unable to find synonyms for many words. Since this was only a benchmark for us, we focused our efforts in testing the social class augmentation method since we felt that it was the most novel and interesting method that could introduce new ways of thinking about data augmentation with potentially positive societal impacts.

2 Related Work

We decided to focus on the area of data augmentation to enable robustness of the Distil-Bert model because data augmentation has shown success in the field of computer vision. For example, we saw in the AutoAugment paper[1], they used a search algorithm to figure out which agumentation technique was best for that image and applied the technique for each image. The result they got was a top-1 accuracy of 83.5% on ImageNet. In addition, another paper in this space did something similar but more simple by just randomly erasing a select rectangular region of an image with random values[2]. What resulted was improvements on object detection and person re-identification. This inspired our data augmentation approach of removing words related to gender, class, and ethnicity. We did not do this randomly but we felt that given we are approaching a different problem in this case of doing natural language processing we felt that we did not necessarily need to randomise the data removal and instead decided to try a novel and innovative approach to tackle robustness in a question and answering system. We also took inspiration from a paper still in the computer vision space that used back-translation on the labels for the training images[3]. This method they used was able to make substantial improvements on a range of computer vision tasks and achieve an error rate of only 4.2 when trained on 25,000 image examples and just 5.43 on 250 image examples. Given that by back-translating their training image labels, which is text data, they were able to see such substantial improvements in the accuracy of their model, we felt that this was a very promising method for us to try and apply to natural language processing problems. We also found that the data augmentation method for computer vision problems across all three papers showed improvements on the model being able to identify images that were out of the context of the training data. Thus, this made us think that this would be a good method to apply to the robustness problem in natural language processing as it worked in another similar field.

Therefore, we decided to look at a couple papers that applied the method of data augmentation in the natural language processing space. In doing so, we also found some great results from people who did different experiments in this space. A paper that we were particularly inspired by, especially for our grammar tense augmentation method (discussed below), was a paper on multilingual augmentation applied to natural language process[4]. The essentially translated all the training data from English to another language and then back to English and trained the model on the newly translated English. Their hope was that this method would create the most broad understanding of the input text and they were able to see significant improvements in their model's BLEU scores, especially when translating the training data to Danish. With this information, we hypothesised that a reason this method of data

[1]Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le. AutoAugment: Learning Augmentation Strategies from Data. 2018.

[2]Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, Yi Yang. Random Erasing Data Augmentation. 2017.

[3]Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, Quoc V. Le. Unsupervised Data Augmentation for Consistency Training. NeurIPS, 2020.

[4]Matthew Ciolino , David Noever , and Josh Kalin. Multilingual Augmenter: The Model Chooses. 2021.

augmentation worked so well in this paper was because by doing back-translation, the text was able to include nuance linguistical and syntactical differences in the training data and thus allow the data to give the model a broader understanding of language structure. This inspired us to apply our grammar tense replacement method because we believed that one major difference in languages are the ways grammar is written and understood across different languages. Thus, we believed that by changing the grammar of our training data, we are able to see similar improvements to the back-translation method, whilst trying something new.

The last paper[5] that inspired our synonym replacement method was one discussing the used a combination of synonym replacement, random insertions, random swaps, and random deletions in order to augment their training data. They found that by replacing their text with synonyms, they were able to see improved performance for both conventional and recurrent neural networks. In addition, they found that by training their model with this data augmentation approach, they were able to see the same accuracy as the normal data set using only 50% of the available training data. This really inspired us as we saw this large improvement as a very promising method. Given that this was a proven method that works for data augmentation with NLP tasks, we decided that we would try this method as well as a benchmark to test and compare with our two more alternative and novel approaches of removing words relating to social class and replacing words with the same word in a different tense.

Lastly, our method of removing any words associated with gender, race and social economic background was very much inspired by many recent conversations and studies around the societal biases that are increasingly being build into natural language models, as discussed by our guest lecturer Yulia Tsvetkov. One paper[6] that particularly struck us was one that discussed how NLP models often time not just propagates bias in training data, they even amplify it. Thus, we believed that if we could improve robuestness whilst rooting our bias in our training data, they than would be an ideal solution. This was obviously very idealistic but we wanted to at least give it a try to see what effects it would give us. In addition, we decided to include words associated with race and social class as well because we believed that a lot of the time certain events and contacts would be more heavily associated with certain races and classes and thus if we are able to root that out then that would be ideal for enabling our baseline model to have a more broad understanding of the training data and thus be able to answer a wider variety of questions, particularly ones out of the contexts of the training data. We were truly inspired by the large amount of work in this space, particularly looking at how we could root out bais in traditional NLP models without removing valuable correlations for the model to draw between examples such as women and mother and men and father. Although, we believed that by augmenting a data set that is entirely rid of all mentioned of "social phenomenons" and structures, we would be able to see even more unique result. We understood that our results could ultimately return unrealistic answer, it was also a good step. One additional note is that we tested our trained model on common question that are not rid of bias and thus this would affect the outcome of our method of removing all social structures from the training data. We have one question for the future of bias researchers and it is whether bias is often time not being tackled because the accuracy measure often times included the biases that make up the training data, just some food for thought! :)

3 Approach

Our main approach for the three different methods of data augmentation we discussed is creating a different function for each of method of data augmentation, running our test data through that function to create a new test data file before training the new test data on our baseline model.

For the method of changing the grammatical tense of the input randomly, we used a random number generator to chose which word in the sentence will be changed in a range between index 0 and the last word in the sentence [1], then we used the pattern.en English Linguistics library to find the word in a different tense, replace it in the sentence and add it to a the dataset to be processed for training. We repeated this until the entire training dataset's contexts had been done.

[5]Jason Wei, Kai Zou, EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. IJCNLP, 2019.

[6]Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. Mitigating Gender Bias in Natural Language Processing: Literature Review. 2019

For the method of replacing words with relation to gender, race, or economic means, we run a for loop over every word in every sentence in context, checking if the word is in our socialclass.txt file of gender, race, or economic means vocab. This file is included in our code, feel free to take a look at the words we ended up compiling and choosing :)! We will create our dictionary lexicon with data on gender words gathered from this Github project, data on race words gathered from this paper on terms relating to ethnicity, and data on economic means words from SMART Vocabulary cloud tool from the Cambridge Dictionary.

For the last method of replacing question sentence words with synonym words, we can simply use a for loop to go through every word in every sentence and using the PyDictionary Module to find the synonym words.

After the milestone, we had hoped to improve on our scores by combining all three methods described to augment the dataset and then send that dataset into pre-processing for training. However, after trying to optimise the time complexities of our data augmentation methods, we were still unable to reduce the data augmentation time for the datasets. Thus, we ended up opting to improve on our grammar tense method since that performed the best the first time. We improved on the grammar tense method by instead of choosing one word in each question sentence to change, we would randomly choose the number of words to change in that sentence and change all those words, then repeat for the next sentence until we have updated every question in that dataset.

We used the standard Baseline Model from the default project, explained in the guideline doc and cloned from MurtyShikhar/robustqa Github repository. In short, it fine-tunes DistilBERT (a smaller, distilled version of the original BERT model[2]) on all the training data.

4 Experiments

We tried the three different methods of data augmentation as well as made improvements to the grammar tense replacement method with the hope of improving our overall performance.

4.1 Data

For the standard original dataset we are comparing with, we used,

Dataset	Question Source	Passage Source	Train	dev	Test
	in-domain datasets				
SQuAD [3]	Crowdsourced	Wikipedia	50000	10,507	–
NewsQA [4]	Crowdsourced	News articles	50000	4,212	–
Natural Questions [5]	Search logs	Wikipedia	50000	12,836	–
	oo-domain datasets				
DuoRC [6]	Crowdsourced	Movie reviews	127	126	1248
RACE [7]	Teachers	Examinations	127	128	419
RelationExtraction [8]	Synthetic	Wikipedia	127	128	2693

For language grammatical data we used the Pattern.en English Linguistics library. We used the conjugation method to change words from present tense in the question to past tense.

For words on gender, race, and economic means, we crowdsourced and created our own library from a range of sources including a Github project on gender words, a paper with terms relating to ethnicity, and the SMART Vocabulary cloud tool from the Cambridge Dictionary for economic means words. We created a socialclass.txt file containing all the words we used and check if words in the training context dataset contained the word, if so, the word would be removed from the context dataset.

For synonym words, we used the PyDictionary Module. We replaced random words in the questions dataset with it's synonym by checking for it's synonym with the PyDictionary words look up function.

4.2 Evaluation method

Our evaluation method was purely quantitative, we used two metrics to measure our different data augmentation method's improvements compared to the baseline model. The two metrics we used are the **Exact Match (EM)** score and **F1** score.

4.3 Experimental details

We used the same baseline model to run every single data augmentation iteration. We ran our experiments by running different combinations and iterations of our data augmentation methods on our training data sets before putting the data through the baseline model.

We ran the following combinations and iterations of the data augmentation methods: the original grammar augmentation method alone, the social class augmentation method alone, the synonyms replacement augmentation method alone, a combination of both the grammar and social class methods, and an improved version of the grammar method.

The training time increased by about an hour when we ran the synonyms replacement and social class methods but was generally around the same amount of time as the baseline. We believe that the reason for this is because our data augmentation methods make use of sets and dictionaries, enabling look up and data removal time to be low, allowing fast data augmentation process.

The learn rate for the different experiments were also relatively the same since we only made changes to the training data and thus the training process was very much the same. Although, the social class method enabled faster learning rates but only by a small fraction, likely because by removing gendered/ ethnicity/ social class words, the contexts of the questions and answers in the training data become more similar and have more correlations since topics are not grouped into categories that represent our societal norms anymore.

4.4 Results

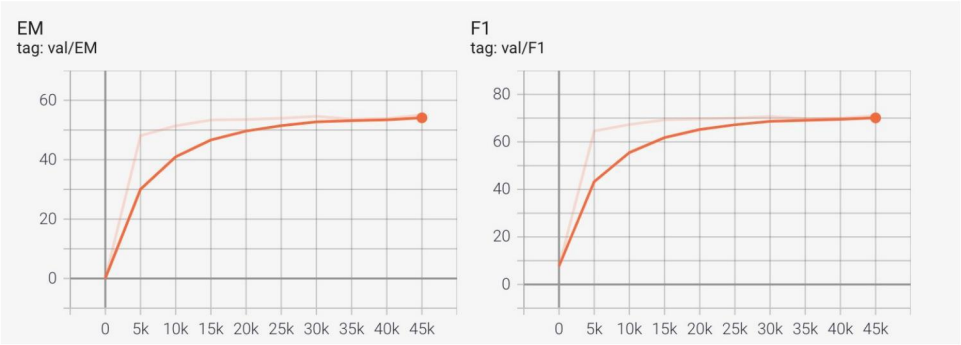

Figure 1: Baseline Tensorboard scores with 0.6 smoothing

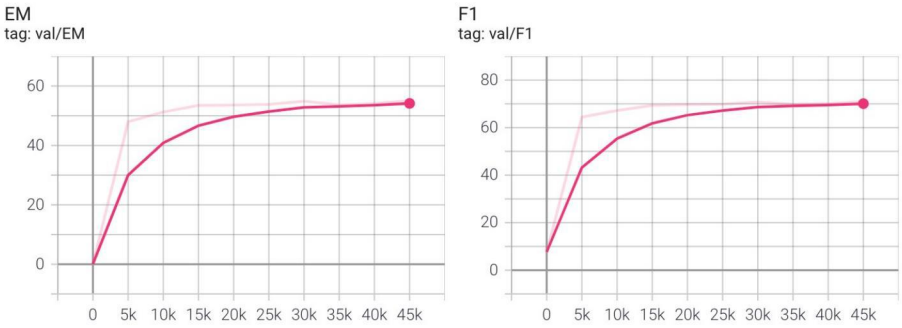

Figure 2: Grammar Tensorboard scores with 0.6 smoothing

While training the grammar model, we found that our scores on Tensorboard performed almost identically, but slightly better than the baseline model, with our model receiving an F1 score of 71.15 and an EM score of 55.27, compared to baseline scores of 71.05 and 55.12. On the test set, our best model (changing grammar) achieved an EM score of 39.335 and an F1 score of 58.051. It performs around the same level as the baseline, with both EM and F1 scores within 1.5 points. The following table compares the validation scores of our different models against the baseline model.

Model	EM	F1
Baseline	33.51	48.60
Grammar	32.15	46.75
Bias	30.84	44.90
Synonyms	28.36	39.70

We improved our grammar score throughout by updating the model to randomize the number of words that we would change the tenses/syntax for in each question, while our former model would only randomly choose one word to change. Although we expected to see some improvement over the baseline, the model performs at almost the same calibre, potentially showing the lack of a drastic effect on the training process even after changing the tenses and syntax of the words.

Our bias/social class model performed much better than we had expected. Since we compiled a list of over 600 terms to exclude (relating to gender, race, or social status/wealth), we anticipated that the score would drop significantly since we were missing so many key words that are commonly used in question answering. The main goal of this model was to present an alternative to traditional NLP models that are prone to generating answers that are biased based on training sets with inherent issues. By removing all words that could potentially result in bias, we created a model that is less biased and more neutral. To our surprise, the model still performed relatively well, and did not lag far behind the baseline and grammar model.

Finally, our synonym model did not perform up to our expectations, and we believe this is the result of the old and outdated API we used to replace words with synonyms. We anticipate that further experiments utilizing this technique while making use of a newer updated API could improve results significantly. In addition, since the model spent over 2 days during pretraining as a result of the outdated API, we were unable to run many tests and perform updates to our model.

5 Analysis

For our grammar/syntax model, we found that small changes in the questions resulted in changes to the way our model responded to questions. For example, while the baseline responds with (Victoria, London), our model responded with (Victoria, London, UK). In some cases, we can see possible negative effects of this grammar change. Fro example, while the baseline answers ("I'm looking forward to the day when more technology will come to my life."), our model responds with (looking forward to the day when more technology will come to my life."), omitting the ("I'm) at the start of the sentence. This could be due to our model changing a non-noun word with its possessive form by adding on an "s" at the end of a word and invalidating "I'm". Another out put of interest is our model obtaining (75 miles northeast of Blainsworth;) compared to the baseline outputting (120 miles northeast of Blainsworth, Nebraska;). The change from 120 miles to 75 miles from the baseline to our model might be a result of our question being framed as asking for a different number present in the sentence or paragraph.

For our bias/social class model, we were able to clearly see the effects of removing all the words in the socialtext file on our results. For example, the output (Miss Murzyn) in the baseline is changed to (Murzyn), clearly showing that the identifier "Miss" is omitted. In another case, the baseline output (dead wife) is outputted as (dead person) from our model. Importantly, the output ("A nurse.") outputted by the baseline was changed to ("doctor.") Although our EM and F1 scores are lower than the baseline for this model, we successfully accomplished our goal for building this model to reduce gendered stereotypes that are inherent in many NLP models.

Finally, our synonym model did not display results that were majorly shocking, mostly either obtaining the same answers as the baseline or diverging substantially. In the cases where the outputs diverged significantly, such as (political resistance even to less far-reaching measures) from the baseline changing to (war), we believe this may be a result of the synonyms being switched in the questions not being compatible with the context of the question being asked.

6 Conclusion

After trying a multitude of different data augmentation methods, we found that the one that had the most improvement from the baseline was the improved grammar augmentation method where instead of just randomly changing one word in the sentence to another tense, we randomly changed up to all words in the sentence. We believe that this method worked best likely because it does something very similar to what the proven back-translation method does and as a result is able to give more improvements.

Although, our method of removing all words related to gender, class and race surprisingly did not entirely flop. In terms of the EM and F1 score, it did worse than the baseline but did not kill the algorithm. We believe that a potential reason for that is because traditional societal biases is also entrenched in the metrics we use to evaluate our NLP models. Thus we believe that in order to truly test if this method worked in a ideal society setting, we would need to introduce new measures of success. We believe that if we want our future NLP models to reflect the world we hope to become rather than the world we currently live in, we are increasingly going to need to push our models to adapt to the ideal goals of a better society. Thus, this may not be the most realistic model for solving basic human problems today, it can be a model that pushes us to do better for the future, to ask questions rid of social biases and free of the constructs and limitations of modern day society.

Additional future work, would be to apply the grammar translation method to not just the question sentences but also the context and answer sentences and to do so randomly. So, rather than just changing the words to one tense, we would randomly chose from a variety of conjugations and singularizations to apply to every word in the data set, creating a set of training data that has grammatical flaws with the hope that it would enable the model to learn broader understanding of texts and draw more nuanced correlations.

A Bibliography

[1] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research 21, 140 (2020), 1–67.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019.

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

[4] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *ACL 2017*, page 191, 2017.

[5] 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.

[6] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In *ACL*, 2018.

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

[8] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115, 2017.*