Data Augmentation application and ensemble learning in question answering using SQuAD 2.0 dataset

Sichen Zhong
University of California, Berkeley
School of Information
sichenzhong@berkeley.edu

April 9th, 2022

Abstract

The *Stanford Question Answering* (SQuAD 2.0) dataset has been developed to enable researchers to design models better tackle the reading comprehension tasks. This paper has focused on the fine-tuning pre-trained transformer-based model, augmenting the dataset, and further ensemble the models using either an unaugmented dataset or an augmented dataset. Our proposed method for ensembling augmented data based on RoBERTa model achieves comparable performance to state-of-art small model performance by achieving an EM score of 80.56 and an F1 score of 83.81 on the development dataset.

1 Introduction

The application of natural language processing in reading comprehension has been an arising field. Question answering is indeed a commonly seen challenge used in the reading comprehension tasks. The Stanford Question Answering Dataset as the most widely used dataset including both answerable and unanswerable question datasets has been seen as the reading comprehension benchmark dataset. As answers are extracted based on the context and question statement, the focus on the data augmentation on the question statement will further improve the machine's accuracy in finding the exact match for the answer. Other than augmentation to the training dataset, the ensemble of model prediction is another apparatus where multiple diverse models predict the answer either using different training datasets or different algorithms. In this paper, we focused on transformer-based architectures such as BERT, ALBERT, and RoBERTa model ensembling in combination with the unaugmented and augmented Squad 2.0 dataset.

2 Dataset

The Stanford Question Answering dataset with adversarial unanswerable questions is a dataset containing 100,000+ questions that are either answerable or unanswerable about the paragraphs that are crowdsourced on a set of Wikipedia articles. [8] Based on the exploratory data analysis, there's roughly a third of questions that are unanswerable in the training dataset and roughly a half of questions that are unanswerable.

3 Background

Inspired by Corbeil and Ghadivel [1], a back-translation approach for data augmentation has been used as the paraphrase-based technique. From what we observed in the SQuAD dataset, the answer was extracted from the context, thus augmenting context is not a suitable technique. Instead, data augmentation applied to the question statement would be a better choice. Multiple data augmentation methods could be applied to the question statement such as back translation, synonym substitution, word embedding substitution, and more. Edward Ma has created a library that can generate synthetic data for improving model performance without manual effort. [5] In this paper, transformer-based architecture has been focused on the study. Ensemble modeling is another way to improve the transformer model without adding too much model complexity.

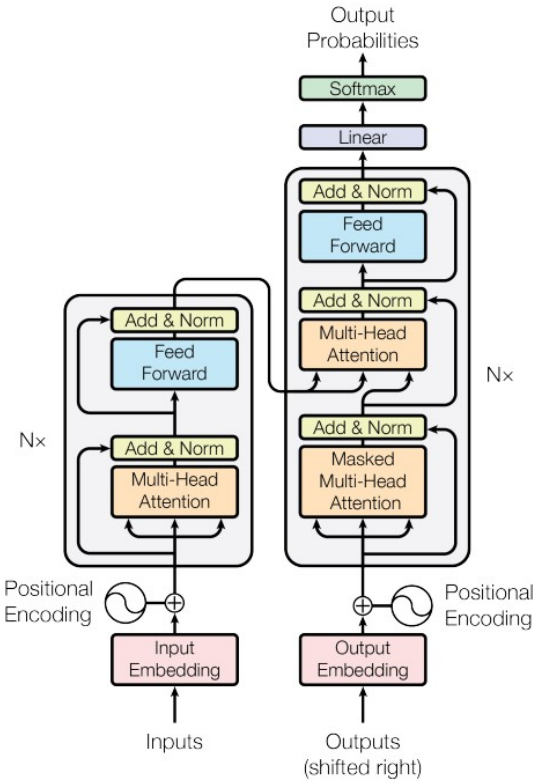

Figure 1: Transformer model structure[9]

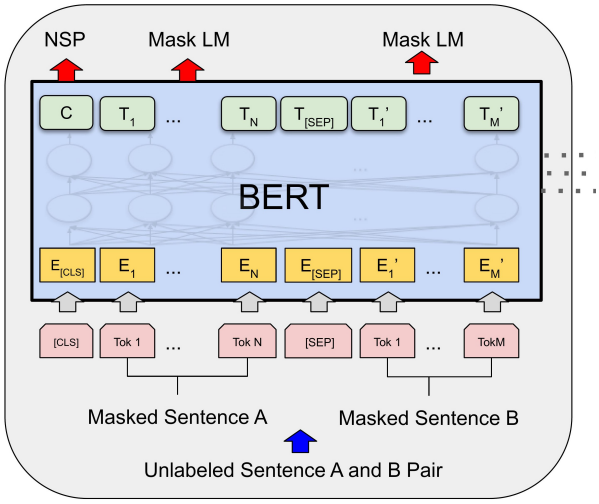

Figure 2: Pre-training procedure for Bert

3.1 Related Work

BERT Bidirectional Encoder Representations from the Transformers model are designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. [2] Bert-base-cased has been chosen in our experimental setup which has 12-layer, 768-hidden layers, 12-heads, and 110M parameters. In the fine-tuning step for the BERT model, input is first tokenized and embedded using WordPiece embeddings into token embeddings, segment embeddings, and position embeddings. The first token for each sentence is determined to be [CLS] as a special classification token. A [SEP] special token is being added to separate the context and question in the sentence. The embeddings are then passed into the encoder. Figure 1 has shown the encoding structure for the transformer, whereas Figure 2 showed the Bert model tokens sent into encoders. The encoder consists of L encoder layers, each containing two sublayers of a multihead self-attention layer and a feed-forward network.

ALBERT A lite BERT model is introduced to utilize the parameter reduction techniques to lower the usage of memory and increase the training speed. Factorized embedding parameterization break the large vocabulary matrix into two small matrices. Cross-layer parameter sharing is another technique used to share the parameter used while the depth of the network growth, which prevent the parameter growing out of the scope. [3]

RoBERTa A robustly optimized BERT model has further improved the training strategy and training dataset to improve the downstream task performance of the BERT model. Various size and domains corpus have been used in pre-training and fine-tuning to optimize the performance. [4]

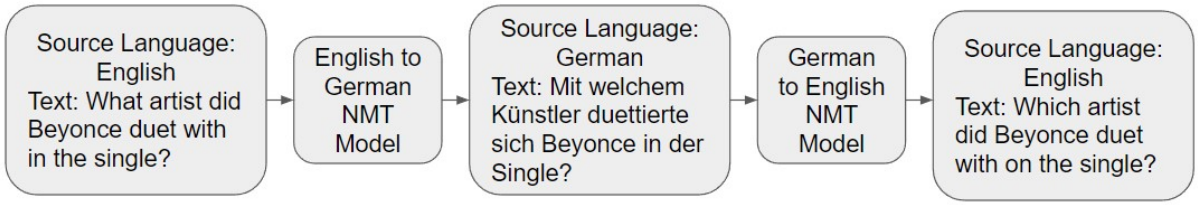

Figure 3: Example question statement back translated from English to German

4 Methods

The experimental design has been split into three parts. First, the introduced three models have been fine-tuned in the original SQuAD 2.0 dataset and ensembled together to compare the EM and F1 performance. Then data augmentation in different techniques has been used. Then the augmented data have individually gone through each mentioned model with the fine-tuned parameter in previous sections. The prediction is then ensembled for further improvement. Last, the combination of two techniques of data augmentation is applied to the training data set to see if further improvement can be achieved.

Fine-tuning For models mentioned above, the models are fine-tuned in the learning rate(LR), number of epochs(E), max sequence length(MSL), and training batch size(BS). Eventually, optimized hyperparameters have been applied to the pre-trained models and using the SQuAD 2.0 for downstream training and evaluation.

Data Augmentation In this paper we have applied 4 individual techniques and 1 combination techniques on the question statement which are random back translation substitution for words, random synonym substitution, context word embedding substitution using BERT model, word embedding substitution using word2vec model, and the combination of back translation and synonym substitution. Google translate api has been used for the back translation task from English to German and back to English as example shown in Figure 3. In the actual back translation augmentation, in order to reduce the computation length, only 0.5% of words in the question statement has been randomly selected and back translated. Nlpaug library developed by Edward Ma has been used for the other data augmentation tasks.[5] As RoBERTa has been the best out of three fine-tuned model, all augmented dataset has trained downstream and eveluate using fine-tuned RoBERTa model.

Ensemble modeling The ensemble learning has used the max voting technique to vote for the most appeared results when the unaugmented dataset is used.[7] Then different combinations of augmented datasets are ensembled to boost the performance based on the RoBERTa model.

5 Results and discussion

5.1 Fine-tuning

Bert-base-case pre-trained model has been used in this study and further fine-tuned the hyperparameters to achieve the optimum performance. As the experimentation on a different combination of the learning rate, the number of epochs, max sequence length, and training batch size, we determined that the learning rate of 4×10^{-5}, 3 Epochs, the max sequence length of 384, and training batch size of 16is determined to be the optimal performance. As the hyperparameters are being tuned, the EM has been improved from 70.81 to 72.55 and the F1 has been improved from 73.95 to 75.94 compared to the default setting for the pre-trained model. By fine-tuning the hyperparameters, the linguistic structure from the first 8 layers is preserved but affects only a few layers, and is specific to in-domain examples which contributes to the improvement in performance.[6] Similar hyperparameters fine-tuning approach has been applied to ALBERT and RoBERTa models to achieve the optimized baseline performance with an unaugmented SQuAD 2.0 training dataset. The metrics used to determine the performance are an exact match and F1 score, where the exact match is the straightforward character pairs, and F1 is defined as equations below.

$$F1 = \frac{2 \times precision \times recall}{(precision + recall)}$$

where,

$$precision = \frac{truepositive}{(truepositive + falsepositive)}$$

$$recall = \frac{truepositive}{(truepositive + falsenegative)}$$

As the variants of the BERT model, ALBERT and RoBERTa after fine-tuning all have shown around 7 scores higher EM and F1 compared to BERT. As shown in Table 1, RoBERTa has been focused on for the next round of experiments for data augmentation as it outperformed in the overall exact match and F1 score, although ALBERT has better no answer exact match and F1 performance.

5.2 Data Augmentation

As discussed in the methods section, 5 different techniques of data augmentation have been applied to the original SQuAD 2.0 dataset. From Table 2, we were able to see although their overall EM and F1 have inversely affected by the different techniques, some have maintained a relatively satisfying performance using the unaugmented data and a few have improved in the questions with the answer. For back translation, we were able to find from the example question statement shown in Figure 3 that the back-translated sentence has changed from 'What' question to 'Which' question. This could affect the encoder finding the potential answer. Also, we see that the word2vec word embedding augmentation based on similarity has increased the F1 score for questions with an answer of over 1 score, but not performing well in the questions without an answer. Also, the combination of the back translation and synonym substitution for random words has boosted the performance on the questions with an answer. This could lead to our further experimentation using the ensemble method to combine the benefit of the dataset augmentation without including the inverse impact.

5.3 Ensemble Modeling

In this section of the study, the ensembled combination of the unaugmented dataset and augmented dataset has been shown in Table 3 for Ensemble models 1, 2, and 3. The augmented dataset is specifically chosen based on previous augmentation performance to purposely boost all metrics regardless the question has an answer or not. The improvement in the EM and F1 has been observed in all ensemble model predictions. Another experimentation to ensemble only using the potential improving augmented dataset has been performed to significantly improve the EM and F1 for questions with the answer but inversely affected the EM and F1 for the questions without an answer. This could be addressed because there's no answer existing in the context thus the question statement being augmented will make the model getting the token harder. Therefore the assumption is that ensembling all previous decent performed ensemble models will further improve the model performance. The final ensemble model 5 has a further improvement in the performance in the question without an answer which leads to an overall performance boost in the overall EM and F1. This result is coherent with previous analysis and techniques and the ensembled model 5 does not seem to overfit the development set.

6 Conclusion

In this paper, the focuses are on three part, fine-tuning pre-trained transformer-based models, augmenting dataset, and ensemble the model predictions. Numerous experiments have been designed to achieve the above mentioned tasks. It is shown that the EM and F1 metrics has decent improvement for all pre-trained models by fine-tuning hyperparameters to optimize the model fitting and carefully avoid the overfitting. On the data augmentation technique, it is not have direct improvement on the overall metrics, but some augmentation method have improvement on the questions with an answer. It then comes to use the ensembling method to further improved our model to the metrics of EM at 80.56 and F1 at 83.81.

For future studies, it is believed that instead of 0.5% of question statement being augmented, more augmentation may contribute more to the question that has an answer. Although the question without an answer may be affected, the ensembling method later can help to alleviate the impact. Also, different techniques for ensembling study like weighted voting, bagging, boosting and more may further improve the model performance. Currently the GPU used is 16GB ram provided by Google Colab pro+ which still have the limitation on the training hyperparameter tuning. If more ram could be used in the future study, there's chance to further optimize all model tuned and also learned better using even more layered model.

Model	EM		F1	
		Has ans/No ans		Has ans/No ans
SQuAD 2.0 + BERT	72.55	72.53/72.56	75.94	79.33/72.56
SQuAD 2.0 + ALBERT	79.26	75.86/**82.64**	82.45	82.25/**82.64**
SQuAD 2.0 + RoBERTa	**80.16**	**78.42**/81.90	**83.25**	**84.62**/81.90

Table 1: Performance result for unaugmented data baseline performance

Model	EM		F1	
		Has ans/No ans		Has ans/No ans
Back Translation + R	**79.52**	78.57/**80.47**	**82.46**	84.45/**80.47**
Synonym + R	77.86	78.27/77.46	81.01	81.01/77.46
Context word embedding + R	66.07	77.95/54.23	69.09	84.00/54.23
Word2vec word embedding + R	70.30	79.74/60.89	73.35	**85.85**/60.89
Back Translation + Synonym R	76.88	**79.94**/73.82	79.81	85.81/73.82

Table 2: Performance result for experiments with data augmentation

Model	EM		F1	
		Has ans/No ans		Has ans/No ans
En1: SQuAD 2.0 + BT + E + R Ensemble	80.02	77.19/82.84	83.28	83.73/82.84
En2: SQuAD 2.0 + BT + S + R Ensemble	80.51	77.05/83.95	83.77	83.60/83.95
En3: SQuAD 2.0 + BT + BT&S + R Ensemble	80.41	77.27/83.53	83.69	83.85/83.53
En4: BT + E + S + R Ensemble	78.59	**79.99**/77.19	81.56	**85.95**/77.19
En5: En2 + En3 + En4	**80.56**	76.95/**84.15**	**83.81**	83.48/**84.15**

Table 3: Performance result for experiments with ensembling models, where BT stands for back translation augmentation, E stands for word2vec word embedding augmentation, S stands for synonym augmentation

References

[1] Jean-Philippe Corbeil and Hadi Abdi Ghadivel. "BET: A Backtranslation Approach for Easy Data Augmentation in Transformer-based Paraphrase Identification Context". In: *CoRR* abs/2009.12452 (2020). arXiv: 2009.12452. URL: https://arxiv.org/abs/2009.12452.

[2] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[3] Zhenzhong Lan et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *CoRR* abs/1909.11942 (2019). arXiv: 1909.11942. URL: http://arxiv.org/abs/1909.11942.

[4] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[5] Edward Ma. *NLP Augmentation*. https://github.com/makcedward/nlpaug. 2019.

[6] Amil Merchant et al. "What Happens To BERT Embeddings During Fine-tuning?" In: *CoRR* abs/2004.14448 (2020). arXiv: 2004.14448. URL: https://arxiv.org/abs/2004.14448.

[7] NikhilSrihari. *squad-2.0-ensemble-predictor*. 2019. URL: https://github.com/NikhilSrihari/SQuAD-2.0-Ensemble-Predictor.

[8] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". In: *CoRR* abs/1806.03822 (2018). arXiv: 1806.03822. URL: http://arxiv.org/abs/1806.03822.

[9] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.