BERT-based Transfer Learning with Synonym Augmentation for Question Answering

Alvin Deng
The University of Texas at Austin
alvin.q.deng@utexas.edu

Evan Shrestha
The University of Texas at Austin
evanshrestha@utexas.edu

Abstract

This paper proposes to extend recent progress made in natural language processing with popular transfer learning models such as BERT to question answering tasks. Our approach combines a flavor of BERT, DistilBERT, with synonym-based question augmentation to detect answers in a given context. Our experiments show that our approach achieves significant improvements over classical techniques in Stanford Question Answering Dataset (SQuAD) 1.1.

1 Introduction

This paper tackles the open-book factual question answering task using Wikipedia as the knowledge base. Question answering is a significant pillar of natural language processing and has gone through significant research over the last few years. Large models focused on transfer learning have recently allowed for improvements across a wide array of tasks, beating out classical techniques by significant margins.

Furthermore, data augmentation techniques for training regularization have been prevalent in computer vision tasks but were historically more difficult to apply to text problems. However, the introduction of modern lightweight natural language augmentation techniques has shown promise.

In this paper, we show how applying developments introduced by BERT can offer notable performance improvements over previous methods. Additionally, we introduce question augmentation through synonym substitution to regularize the training for robustness.

2 Related Work

Open-book question answering is a common task in natural language processing. DrQA (Chen et al., 2017) aimed to address the task through a Retriever-Reader mechanism, in which a Retriever model employs a search mechanism based on bi-gram TF-IDF representations to return relevant documents from a pool of documents and a Reader model detects answer spans within the chosen context. This Retriever-Reader framework has become popular, with further development such as with BERTserini (Yang et al., 2019) and the Multi-passage BERT QA model (Wang et al., 2019).

Further motivation for this paper comes from improvements that have been introduced since the development of DrQA. In particular, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) improved upon many state-of-the-art models at their inceptions and continue to be cornerstones in present-day language modeling tasks. Our approach focuses on BERT, which was designed to provide deep representations that could be fine-tuned with few output layers while providing strong performance in various tasks. BERT introduced a novel masked language modeling which allows for powerful transferable bi-directional training of fine-tuned models. Through knowledge distillation, DistilBERT (Sanh et al., 2019) improved the accessibility of BERT-like performance by reducing the size of BERT by 40% while maintaining 97% of the performance. By dramatically reducing the size of the model, DistilBERT allows for significantly quicker training with much fewer resources.

Additionally, recent research in text augmentation techniques (Wei and Zou, 2019) shows potential through several simple text operations: synonym replacement (SR), random insertion (RI), random swap (RS), and random deletion (RD). Text augmentation results suggest that full-data accuracy can be attainable with up to a 50% reduction in data used.

3 Methodology

In the following section, we introduce our approach, which implements two key changes: (1) the introduction of a DistilBERT representational layer to the model architecture and (2) question augmentation through the use of synonym substitution.

3.1 Model Architecture

Our approach is similar to that used in the Reader portion of the approach found in DrQA. However, we perform a transfer learning process by substituting the initial embedding layer with the final layer of a pre-trained DistilBERT model. From there, we compute aligned question embeddings through a Context2Query layer (Seo et al., 2016) to retrieve a weighted sum of the question and context. We then concatenate these aligned embeddings with the passage embeddings, which are then fed into a Bi-LSTM (Cornegruta et al., 2016) passage encoder, and the question embeddings are fed into a separate Bi-LSTM question encoder. We then compute the attentive sums of the encoded questions to retrieve representational question vectors. These encoded passages and question vectors are then fed into output layers to retrieve probability distributions for the start and end indices in the given context. Finally, we perform a search over the distributions to find the start index with the maximum likelihood and find the ending index with the highest joint probability within a window. The context is then sliced over these indices to predict an answer.

3.2 Augmentation

In an attempt to regularize and increase the lift of the training data, we apply synonym replacement (SR) to a question generation process. We use `nlpaug`[1], a package for generating synthetic text data. We apply a probabilistic question augmentation process to artificially increase the size of our training data by randomly substituting words for their synonyms. These synonyms are generated by WordNet (Miller et al., 1990), a commonly used lexical database. These generated sentences are appended to the training set prior to training, and the model learns from this modified pool of questions.

4 Dataset and Evaluation

4.1 Dataset

We focus on Stanford Question Answering Dataset (SQuAD) 1.1 (Rajpurkar et al., 2016). SQuAD provides a large amount of contextual text with related questions crowdsourced from Wikipedia articles. SQuAD consists of over 100,000 questions with the answers labeled in corresponding articles and provides for a rich source of factual information commonly used to train question answering models. Human performance on the dataset achieves an F1 score of 86.8%.

4.2 Evaluation Metrics

We use two primary metrics to evaluate the performance of our approach: F1 score and exact match.

F1 score. The F1 score is a measurement of accuracy computed as the harmonic mean of precision and recall. Specifically, we tokenize the predicted answers and ground truth labels and compute the F1 score as follows:

$$\text{F1 score} = 2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (1)$$

The F1 scores for all predictions are then averaged to retrieve the final F1 score (macro-averaged).

Exact Match. Exact match measures the proportion of predicted answers that match exactly with their corresponding labeled answers.

4.3 Training

To train the model, we first tokenize the questions, both augmented and unaugmented, and answers with the BERT tokenization process. Specifically, we use the DistilBertTokenizer and DistilBERT base model (`distilbert-base-uncased`[2]) provided by Hugging Face through the `tokenizers`[3] and `transformers`[4] packages. We need to use the corresponding tokenization process (equivalent to the standard BERT tokenization) to leverage the pre-trained DistilBERT model. The hidden dimension size of the internal Bi-LSTM layers is set to 256, and we use a batch size of 192. The model is trained on a Tesla V100 GPU for 10

[1] https://github.com/makcedward/nlpaug

[2] https://huggingface.co/distilbert-base-uncased

[3] https://github.com/huggingface/tokenizers

[4] https://github.com/huggingface/transformers

epochs with early stopping for approximately 70 minutes.

5 Analysis

To address questions from Section 1, we conduct a series of experiments to validate our claims and to show some interesting properties of our approach. Section 5.1 explores the effectiveness of synonym augmentation in improving the robustness of various models. Section 5.2 compares the model performance between the baseline models and our approach. Finally, Section 5.3 provides an ablation study to measure the individual contribution of components (e.g., Synonym Augmentation) to the model performance.

5.1 Synonym Augmentation

To understand the impact of synonym augmentation on the robustness of the model performance, we employ a hyper-parameter search on the probability of augmenting the question text ranging from 0.1 to 0.5. Table 1 compares these probabilities on the development set of the SQuAD dataset. Surprisingly, we observe a marginal performance degradation as we increase the augmentation probability. The model suffers about -2.47% and -2.03% drop in EM and F1, going from a probability of 0.1 to 0.5. One possible reason for this regression is that the synonym replacement augments the grammatical structure of the question. It might confuse the question embedding so that the wrong context is being served to the downstream layers. Another explanation from the research community suggests a similar pattern when applied to large training sets, possibly due to the strong generalization strength already provided by substantial unaugmented training sets (Wei and Zou, 2019).

5.2 Baseline Comparison

To measure the effectiveness of our approaches, we compare our model with two baselines with the same model architecture, except the embedding layer is either randomly initialized or GloVe-based (Pennington et al., 2014). To simplify the comparison, we choose the model with a synonym augmentation probability of 0.1 to compare with the baselines since it has the best performance shown in Table 1. Table 2 presents the model performance on the development set of the SQuAD dataset. With the novelty of masked language models, DistilBERT as the embedding yields significant improvement in both EM and F1 (+27.79% and +19.44% respectively compared to GloVe embedding).

5.3 Ablation Studies

We run a series of ablation experiments to better understand each component's performance contribution (e.g., DistilBERT). Results are shown in table 3 and discussed in detail next.

Tokenization: Since leveraging the DistilBERT model provided by Hugging Face requires us to use their tokenization, we try to apply the same tokenization to models with randomly initialized embedding to observe its contribution to the model performance. Unfortunately, the model with GloVe embedding had some compatibility issues with Hugging Face's tokenization, and we were unable to obtain its results. However, the comparison between the randomly initialized embedding indicates that the tokenization does not significantly lift EM and F1.

Synonym Augmentation: Similar to findings shown in Section 5.1, we have also observed a minor performance drop across different combinations of embedding and tokenization techniques. One interesting observation is that the DistilBERT-based model without the synonym augmentation performs the best out of all models in this ablation study.

Embedding: Given the effectiveness of pre-trained transformers, we have observed the most performance boost (+30.8% in EM and +21.8% in F1 compared to GloVe-based model)[5] by using DistilBERT as the embedding compared to other approaches such as random initialization and GloVe.

6 Conclusion

We show that applying pre-trained language models such as DistilBERT can provide significant performance gains to previous approaches. Additionally, we explore the potential of text augmentation through synonym replacement on questions, although resulting improvement is yet to be shown. Further exploration on the topic could examine more complex strategies, possibly utilizing language models such as GPT-2 (Radford et al., 2019) for more advanced grammatical substitutions. We expect a combination of transfer learning and strong text augmentation has further potential to significantly improve upon classical techniques.

[5] We referencing the result from DistilBERT as embedding without synonym augmentation

Embedding	Tokenization	Synonym Aug. Prob.	Exact Match (EM)	F1
DistilBERT	Hugging Face	0.1	**61.43 ± 0.21**	**72.30 ± 0.18**
		0.2	61.14 ± 0.17	72.14 ± 0.18
		0.3	60.18 ± 0.56	71.29 ± 0.40
		0.4	59.46 ± 0.27	70.67 ± 0.24
		0.5	59.91 ± 0.04	70.83 ± 0.04

Table 1: Hyper-parameter search with various augmentation probabilities on the development set of the SQuAD dataset. Evaluation metrics are averaged with 3 runs on different seeds.

Embedding	Tokenization	Synonym Aug.	Exact Match (EM)	F1
Random Weights	MRQA		37.70 ± 0.29	50.99 ± 0.23
GloVe (6B/300D)	MRQA		48.07 ± 0.36	60.53 ± 0.35
DistilBERT	Hugging Face	✓	**61.43 ± 0.21**	**72.30 ± 0.18**

Table 2: Comparison between baseline models and proposed model on the development set of the SQuAD dataset. Evaluation metrics are averaged with 3 runs on different seeds. The synonym augmentation probability is 0.1.

Embedding	Tokenization	Synonym Aug.	Exact Match (EM)	F1
Random Weights	MRQA		37.70 ± 0.29	50.99 ± 0.23
Random Weights	MRQA	✓	37.05 ± 0.29	50.45 ± 0.25
Random Weights	Hugging Face		36.82 ± 0.36	50.60 ± 0.33
Random Weights	Hugging Face	✓	36.31 ± 0.10	49.77 ± 0.19
GloVe (6B/300D)	MRQA		48.07 ± 0.36	60.53 ± 0.35
GloVe (6B/300D)	MRQA	✓	47.24 ± 0.30	59.69 ± 0.30
DistilBERT	Hugging Face		**62.88 ± 0.04**	**73.77 ± 0.30**
DistilBERT	Hugging Face	✓	61.43 ± 0.21	72.30 ± 0.18

Table 3: Ablation study on the influence of various components (e.g., DistilBERT, synonym augmentation) on the performances of the development set of the SQuAD dataset. Evaluation metrics are averaged with 3 runs on different seeds. The synonym augmentation probability is 0.1.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Savelie Cornegruta, Robert Bakewell, Samuel Withey, and Giovanni Montana. 2016. Modelling radiological language with bidirectional long short-term memory networks. *arXiv preprint arXiv:1609.08409*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*.