Question Answering with BERT and Answer Verification

Kevin Culberg
Department of Computer Science
Stanford University
Stanford, CA 94305
kculberg@stanford.edu

Abstract

The task of reading comprehension with unanswerable questions challenges a models ability to both correctly predict an answer span as well as determine if the question can even be answered. Most solutions involve adding placeholder values to the output prediction to predict a question is unanswerable if those placeholders score high enough as the start and end locations of the answer span. I propose combining a method of answer verification with a model architecture that has demonstrated success at question answering. This two stage model combines the strength of answer prediction with the ability to determine if the predicted answer is a valid answer to the question asked. Models with a second stage performed worse on the metrics of EM and F1 on the SQuAD 2.0 dataset, but did show an improvement in correctly determining if questions were unanswerable. Overall it is likely that the addition of a second stage with shared weights caused instability during training resulting in the decreased performance.

1 Introduction

The problem of machine reading comprehension is difficult because it requires understanding enough about language to be able to answer a given question with a short string from a contextual paragraph. The addition of unanswerable questions to the problem further increases the difficulty because now the technique needs to be able to determine that a given question is even answerable as well as where the answer is inside of the paragraph. Improvement on this problem signals advancement in many areas of natural language processing that could also be improved with similar techniques. This makes the problem of reading comprehension useful from the standpoint of improving the performance of natural language processing networks. Additionally, this problem has numerous practical applications. Personal assistant technology has become a more common part of everyday life, but these devices often return information in short snippets. No one wants to wait for their audio activated device to recite an entire Wikipedia article related to their question. Instead, users prefer receiving the exact answer in as few words as possible. Advancement in the problem of reading comprehension is important because it signals growth across the field of natural language processing and has real benefits on practical applications.

Current networks are very good at recognizing where the answer is inside of a contextual paragraph, but still frequently fail to identify whether or not a question can be answered. I propose a model that combines the current state of the art of Pre-trained Contextual Embeddings (PCE) with answer verification in order to improve the ability to recognize unanswerable questions. Specifically, I intend to determine if the addition of an answer verification network improves performance on identifying unanswerable questions and if this addition increases or decreases performance on answerable questions.

2 Related work

The task of reading comprehension through question answering has seen tremendous advancement following the use of PCE in models. These models are able to predict an answer to a question given only the text of the question and a paragraph of text that may or may not contain the answer. If the question is not answerable with a span of text from the given paragraph then the model should predict that the question is unanswerable. Devlin et. al. [1] introduced a new design for a language representation model called the Bidirectional Encoder Representations from Transformers (BERT) that uses both left and right context in all layers compared to previous models that only used one direction in the self-attention layers of the transformer. This bidirectional context allows for increased understanding of any token in a sequence by handling cases where tokens that modify or alter the context fall before or after a that token. This model also has great flexibility and can act as a pre-trained base layer for any NLP task by providing an embedding for the input. Additional layers can be combined with BERT depending on the task which allows for flexibility of the output while decreasing the number of weights that need to be trained from scratch. BERT achieves superior performance to other similar models such as OpenAI GPT on a wide range of NLP benchmarks and at the time achieved state of the art results on eleven NLP tasks.

The key technique that BERT is built on is a model architecture known as the transformer introduced by Vaswani et. al. [4] Transformers improve on previous model architectures by removing the reliance on recurrent steps and instead using a self-attention. Self-attention can be calculated with matrix operations allowing it to be parallelized which can greatly speedup calculations. Their proposed model architecture also featured multi-headed attention which allowed for learning multiple different context spaces for use in computing attention scores. This further increases the flexibility of the model and allows for models that can be more easily interpretable by a human.

In 2018, Hu et. al. introduced a method of answer verification to improve performance on question answering tasks. [2] In their paper, they proposed three different architectures by which to verify answers predicted for a given question. These architectures applied different methods of either feeding in the input to the answer verifier as a long sequence, encoding the question and answer sequences independently, and a hybrid method. These models did not benefit from the transformer architecture and instead relied on bidirectional LSTM layers and GloVe embeddings. At the time of publication this technique achieved state of the art on the SQuAD 2.0 dataset with an F1 score of 74.2.

3 Approach

I propose a two stage network consisting of answer prediction and verification in order to improve the ability to determine if a question has no answer, see Figure 1. The first stage uses BERT with a head layer that is responsible for predicting the start and end location for the answer span from the context paragraph. The second stage acts as an answer verification step. The inputs to this stage are the same question input from the first stage concatenated with the predicted answer sequence extracted from the context paragraph. The second stage contains the same BERT base model as the first stage but instead feeds into a head layer that predicts a binary classification if the answer is valid. This method of answer verification was partially inspired by work by Hu et al.[2]

A standard baseline for this NLP task and the one used for comparison is BERT-base with a simple head layer to predict an answer as well as whether the question is answerable. BERT-base consists of 12 transformer blocks, a hidden size of 768, 12 self-attention heads. The head layer consists of a single linear layer also with a hidden size of 768. Further description of the model can be found in Devlin et al.[1] A prebuilt implementation of the baseline was used and is available at https://github.com/huggingface/pytorch-pretrained-BERT.

The proposed model builds off of base BERT with the addition of a transformation layer. This layer takes in as input the question and context paragraph along with the predicted start and end positions of the answer. The layer then extracts the answer from the context paragraph and concatenates the tokens corresponding with the answer to the question tokens. These two strings are separated by a special token similar to how the context paragraph is separated from the question for the first stage input. The output for this layer is then a vector containing only the tokens from the question and

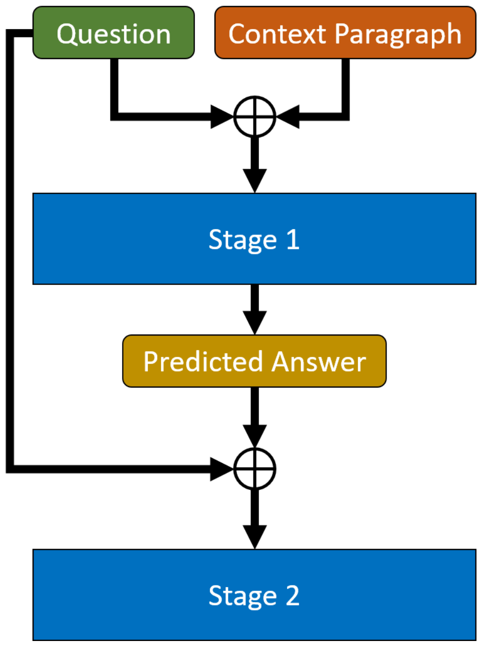

Figure 1: Model diagram depicting the two stages for answer verification and their inputs.

the predicted answer. This is then sent into the same BERT base model from the first stage to be reprocessed to determine if the answer is valid.

The loss function used for the two stage model combines the cross entropy loss for the predicted start and end location of the answer span for the first stage and binary cross entropy loss for the prediction of the question being answerable from the second stage:

$$Loss = -\lambda_1[\sum_{i=0}^{M} \mathbb{1}_i^{\text{poss}}(y_{s,i}\log p_{s,i} + y_{e,i}\log p_{e,i})] - \lambda_2[(1-y_a)\log\sigma(z) + y_a\log(1-\sigma(z))] \quad (1)$$

where $\lambda_{1/2}$ is the loss weight for stage 1 or 2, $y_{s/e}$ is the true start or end position, $p_{s/e}$ is the predicted start or end position probabilities, M is the sequence length, y_a is the true label indicating if the question is answerable, σ is the sigmoid function, and z is the value predicted from the second stage indicating if the question is not answerable. The loss from incorrect answer position predictions is only calculated if the given question is actually answerable. Additionally, the two sources of loss are be weighted independently to control how much influence they have over the updates to the weights in the base BERT layers.

4 Experiments

4.1 Data

The dataset I used for training and evaluating the model is the SQuAD 2.0 dataset.[3] This dataset contains approximately 150,000 questions with half of them being answerable. No additional data was added nor was data augmentation performed.

4.2 Evaluation method

Evaluation of the baseline and custom models was done with exact match (EM) and F1 score using the given answers that accompany the SQuAD 2.0 dataset. EM is a binary score that requires the predicted answer to match the ground truth answer exactly. F1 is less strict and combines the

Table 1: Results on SQuAD 2.0 PCE leaderboard

	Test Set		Dev Set			
Model	EM	F1	EM	F1	Precision	Recall
BERT-base	**59.679**	**63.026**	**59.230**	**62.353**	**76.23%**	57.89%
BERT+AV1	50.871	53.389	53.455	55.525	60.59%	**67.80%**
BERT+AV2	48.318	52.088	48.602	52.548	71.37%	41.32%

precision and recall of the predicted answer compared to the ground truth. F1 score is calculated as F1 $= 2PR/(P + R)$, where P is precision and R is recall.

In addition to overall EM and F1 score, I also evaluated the performance of the answer verifier network by checking the precision and recall on determining the answerability of a question to determine if there is a measurable improvement on unanswerable questions. Precision is the correct number of unanswerable questions predicted out of the total number of predicted unanswerable questions. Improvement in precision shows a decrease in falsely labeling questions as unanswerable. Recall is the correct number of unanswerable questions predicted out of the total number of unanswerable questions. Improvement in this criteria indicates a decrease in the number of unanswerable questions that were incorrectly predicted as answerable.

4.3 Experimental details

Two implementations of the BERT+AV (BERT with answer verification) model where trained and evaluated along with BERT-base. BERT-base only contained stage 1 with no answer verification features. BERT+AV1 was trained from only the non-fine tuned weights and contained a dropout layer after the second stage with a dropout chance of 0.1. BERT+AV2 was trained on top of the fine tuned weights from BERT-base, but did not use dropout in its second stage. All models were trained on the SQuAD v2.0 training set for three epochs with a learning rate of $3e^{-5}$ and a batch size of 3. Training time for the models with answer verification took approximately three hours longer per epoch.

4.4 Results

The performance for the BERT-base model is as expected based on other BERT models submitted to the SQuAD 2.0 leaderboard, see Table 1. It was believed that the addition of answer verification to the model would improve performance on EM and F1 scores. However, this is not the case with the two models that feature a second stage scoring far below the BERT-base model. Both models reused weights from the first stage for the second stage task of answer verification. This likely pulled the weights in two different directions during the training process as they were updated to better predict answer positions as well as better predict question answerability. This conflict caused decreased stability during training and resulted in worse performance overall. Of the two models with answer verification BERT+AV1 performed the best. This model was trained from the same starting point as BERT-base which likely allowed the network to better specialize for the new added task of answer verification. This network also featured dropout for the second stage which improved performance on correctly determining a question's answerability which can be seen in its precision and recall scores.

By focusing only on the model's ability to correctly predict answerable questions it is possible to see how much the addition of answer verification can improve the model's performance. BERT-base achieved a precision of 76.2% and a recall of 57.9% on the dev set. This means that the model has a low rate of predicting false positives (incorrectly predicting a question is unanswerable), but only 57.9% of the questions that are unanswerable in the dev set are correctly predicted as being unanswerable by BERT-base. BERT+AV1 outperforms BERT-base on the metric of recall by almost 10%. This indicates that answer verification did improve the models ability to correctly determine if a question was unanswerable by decreasing false negatives. This also brought more balance to the values for precision and recall. However, this to comes at a cost of the model being more likely in general to predict a question as unanswerable resulting in lower a precision score.

5 Analysis

By reviewing the predicted answers from each model it is possible to make better assessments of how the addition of answer verification impacted performance. In reviewing these predictions the first obvious difference is that many more questions were predicted as unanswerable by the models that featured answer verification. In many cases this prediction was correct, but the models also tended to overpredict questions as being unanswerable. Another important difference is that the answers predicted by the models with answer verification tended to contain extra words that were not part of the true answer. This often included phrases separated by a comma from the true answer that were related to or modified the true answer, but not actually included by the human provided true answers. This difference further contributes to the lower performance on the evaluation criteria. It is likely that the model structure favored longer answer strings than shorter due to the answer verification step. The second stage takes as input the predicted answer and would likely perform better at the task of answer verification if more context were included. By predicting longer answers the second stage was receiving greater context enabling it to be more accurate about the questions answerability. This indicates that one potential modification to the model architecture would be to add additional context to the second stage by taking a window's worth of tokens on either side of the answer.

6 Conclusion

Overall, the addition of answer verification to the network in the manner proposed was not successful. The method showed small success in improving the precision of predicting whether or not a question was answerable, but this came at a large cost to the overall evaluation metrics of EM and F1 scores. This also resulted in a large drop to the recall score on predicting answerability indicating a large increase in false negatives for a small gain in true positives. It is likely that separating the two stages of the model so that they do not share any weights would have been an improvement. However, this change would come at a greater increase to the training time as well as doubling the size of the model.

Future work on this technique should pursue models with different structures for the answer verification stage final layers. One example mentioned previously would be to increase the amount of context provided to stage 2 by adding more tokens from before and after the predicted answer. Other options include using only the context sentence that the predicted answer was extracted from along with the question and answer as input to the second stage. It is possible that this will improve performance due to enabling the second stage to make a better determination if the predicted answer is valid. Additionally, pursuing a split model setup where the two stages do not share weights would likely produce improved results. This second stage could also take the structure of BERT or perhaps another model focused on the task of sequence classification. In this case the sequence would be classified as a true answer to a question or an invalid answer. This would provide two benefits. First, the two stages could be trained independently of each other. Second, the weights for these stages would not be pulled in separate directions during training and they would be able to better specialize in their respective tasks. This should result in more stable training and improved performance.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Ming Zhou. Read + verify: Machine reading comprehension with unanswerable questions. *CoRR*, abs/1808.05759, 2018.

[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] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.