Explicitly Learning Implicit Classification

Sai Kandallu
kandallu@stanford.edu

Abstract

With the introduction of SQuAD 2.0 by Rajpurkar et al. [2018], question-answering has become an implicit classification task in which it is just as important to classify unanswerable questions as it is to answer answerable ones. In this paper we explore ways to alleviate implicit classification by treating question-answering as a partial classification problem. We use new data augmentation by Removal of Answer Sentences (RAS) to fix the negative class distribution, but only observe an increased Dev F1 of 76.1 in yes/no-style questions with a downside of increased training time. We test the feasibility of a fine-tuning BERT with just the classification portion of SQuAD 2.0 and observed reduced performance compared to a baseline question-answering model. We find that fine-tuning BERT is very sensitive to changes in training distributions and RAS degrades performance due to differences in context lengths.

0.1 Introduction

Language modeling has produced increasingly better quality models for a variety of tasks [Devlin et al., 2018], including question-answering tasks such as such as the Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al., 2018]. These models have been tested on a variety of tasks that require natural language understanding.

With the release of SQuAD v2.0, there is a need to know whether the question can be answered at all with the given context. Between 30-50% of the questions in SQuAD 2.0 do not have answers.

The rise of pre-trained contextual models have shown that models trained with general language modeling tasks can be fine-tuned. Since fine-tuned models have been shown to perform well on a variety of tasks [Howard and Ruder, 2018, Peters et al., 2018, Devlin et al., 2018], it is of particular interest to answer SQuAD 2.0 as well. We choose to explore ways to improve the classification task in Bidirectional Encoder Representations from Transformers (BERT) as a way to additionally improve the question-answering task.

0.2 Related Work

In the past, other transfer learning techniques have been tried with limited amounts of success such as Universal Language Model Fine-tuning (ULMFiT) [Howard and Ruder, 2018]. ULMFiT shows improvements in many text classification datasets, but requires careful application of training techniques, including discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing, during the fine-tuning stage.

More recently, other pre-trained contextual embeddings have greater success, transferring general language tasks into a variety of NLP tasks such as Embeddings from Language Models (ELMo) and BERT. Of these tasks is question-answering of which the SQuAD dataset is commonly used.

Since the release of the original SQuAD 1.1 dataset, models have been built that match and even surpass human-level performance. Particularly notable of these models is BERT, which achieves an F1 of 91.835, 0.614 points over human-level performance. When ensembled, BERT achieves even higher scores.

BERT hoped to solve a representational issue with current language models. Despite improvements in encoder-decoder networks, language models relied on a unidirectional representation of contextual embeddings. In the best case, architectures created an artificial, bidirectional encoding by performing a shallow concatenation of left-to-right and right-to-left encodings [Devlin et al., 2018]. BERT uses stacked Transformer units [Vaswani et al., 2017] to train these encodings efficiently and at a large scale.

0.3 Approach

0.3.1 Model

BERT is able to transfer pre-trained contextual embeddings into a variety of NLP tasks. Devlin et al. [2018] show performance improvements across tasks ranging from classification, tagging, and question answering. Because of this, we choose to use BERT as the base of our exploration.

In order to learn a transferable bidirectional encoding, BERT pre-trains a neural architecture composed of L Transformer block layers, H as the hidden size dimension, and A attention heads.

BERT uses two pre-training tasks: "Masked LM" (MLM) and "Next Sentence Prediction" (NSP). In MLM, the model learns to predict a word that has been masked at random. The prediction happens by taking just the hidden vector from the masked token and getting a softmax probability distribution over the vocabulary. The second pre-training task, NSP, learns sentence-level relationships and understands whether two sentences are related. In this task, the second sentence in a sequence of sentences has been replaced with a random sentence 50% of the time and the model has to learn whether or not the second sentence is actually the next sentence. This was shown to empirically improve performance on sentence-level tasks such as question answering and inference.

For question-answering tasks, we use the BERT base architecture described above along with a linear layer such that we have start and end logit probabilities for each token. If $S \in \mathbb{R}^H$ is the learned start vector and $E \in \mathbb{R}^H$ the learned end vector, the learned start and end positions s and e are:

$$s = \text{argmax}_i \left(S \cdot T_i\right)$$
$$e = \text{argmax}_i \left(E \cdot T_i\right)$$

We also train a separate classification model, which is composed of the original BERT base architecture described above with an additional dropout layer and a linear layer. We use dropout probability of 0.1 for classification. As per Devlin et al. [2018], we use `CLS` token at the beginning of each input example from which we extract classification labels.

Both of these models have special treatment for examples with large number of tokens. We only include `max_seq_length` tokens from the full example. If the example contains more tokens, then we consider it separately but create a subexample moving forward `doc_stride` tokens.

0.3.2 Data Augmentation

The suggested application of BERT is to fine-tune the pre-trained model with a modified output head depending on the given task for 3 to 4 epochs. For SQuAD 1.1, Devlin et al. [2018] show that using two logit outputs representing the start and end indices achieves state-of-the-art Test F1 of 93.2.

Unlike SQuAD 1.1, the SQuAD 2.0 training set has 33.4% negative examples. We observe that there is a larger negative class distribution in both the development and test datasets with 50.1% and 48.9% negative examples respectively [Rajpurkar et al., 2018]. Based on this observation, we expected to correct this class distribution with augmented data that expands the negative class.

Using RAS, we remove sentences containing the answer using ruled-based sentence splitting, regular expressions. For a given question and answer, we remove all sentences that contain any part of the answer span. RAS provides an approximate two-fold augmentation factor. For every positive class example, we can obtain a negative class example by removing the sentences containing the answer. We say approximate because a small portion of the original positive class examples, about 2.3%, contain no remaining context sentences after removing the full answer span. With RAS, we augment the negative class to 59.6%, which more closely matches the negative class distribution in development and test datasets.

Algorithm 1 Data augmentation using RAS

```
1: Questions $Q$
2: Initialize $R \Leftarrow \emptyset$
3: for $q \in Q$ do
4:     if $q$ has answer then
5:         Let $c, a \Leftarrow$ context($q$), answer($q$)
6:         Let $S_c, S_a \Leftarrow$ sentences($c$), sentences($a$)
7:         Create $q_r \Leftarrow$ no-answer-question with context $S_c - S_a$
8:         Update $R \Leftarrow R \cup q_r$
9:     end if
10: end for
11: Return $Q \cup R$
```

Context: Many of the world's largest cruise ships can regularly be seen in Southampton water, including record-breaking vessels from Royal Caribbean and Carnival Corporation & plc. The latter has headquarters in Southampton, with its brands including Princess Cruises, P&O Cruises and Cunard Line.
Question: Besides Carnival, what other major cruise line parks its record-breaking cruise ships in Southampton Water?
Answer: Royal Caribbean
Updated Context: The latter has headquarters in Southampton, with its brands including Princess Cruises, P&O Cruises and Cunard Line.

Figure 1: Example of context, question, and answer from the SQuAD 2.0 training set. The updated context is the additional example gained with RAS-augmentation (the question and answer remain the same).

Although there are other ways to augment this data by swapping or masking the original context of a question, we find in experiments that simple data augmentation does not improve performance. In future work, we can attempt to generate synthetic data using an adversarial model similar to the process described in SWAG [Zellers et al., 2018]. We also make recommendations for other approaches in the Analysis section.

0.4 Experiments

0.4.1 Dataset

We use SQuAD 2.0 for experimentation and implemented RAS for data augmentation. In SQuAD 2.0, the model is given a context paragraph from Wikipedia and a crowdsourced question, which may be answerable.

With RAS augmentation, we use this example as well as an additional example as seen in the example in Figure 1. The Question remains the same and the new example does not have an answer.

With RAS-augmentation, we were able to augment the original 130,319 questions with an additional 84,802 negative examples.

0.4.2 Experimental Setup Details

We started with a pre-trained base BERT model of $L = 12$, $H = 1024$, and $A = 16$ [Devlin et al., 2018] trained on uncased tokens using WordPiece tokenization [Sennrich et al., 2015] found in HuggingFace's PyTorch implementation of BERT.[1]

We train a baseline model with the standard training set as well a variant trained with the dataset augmented using RAS. RAS uses our implementation.

Because training time scales linearly with the augmented dataset, we train each model for a two epochs using a batch size of 6, `max_seq_length` of 384 after WordPiece tokenization, `doc_stride` of 128, and an initial Adam optimizer learning rate of 0.00005. Although we trained the baseline model

[1]https://github.com/huggingface/pytorch-pretrained-BERT

System	AvNA	EM (EM on Test Set)	F1 (F1 on Test Set)	Precision	Recall
BERT (Baseline, Epochs = 1)	**75.8**	**67.4** (-)	**70.8** (-)	**69.8**	**87.3**
BERT + RAS (Epochs = 1)	68.8	61.5 (-)	64.6 (-)	65.7	73.0
BERT (Baseline, Epochs = 2)	**78.5**	**71.0** (70.482)	**74.4** (73.877)	**74.4**	**84.2**
BERT + RAS (Epochs = 2)	72.8	66.5 (64.176)	69.1 (67.492)	70.9	73.5

Table 1: SQuAD 2.0 Dev results. AvNA refers to typical classification accuracy in which questions with answers are the positive class. Even considering the negative class as more important, the RAS variant did not show improvements in specificity or negative predictive value either (not reported here).

		what	when	which	where	how	yes/no	why	other	who
	Total	3725	431	241	263	576	36	91	27	688
Baseline	EM	**67.6**	**72.2**	**72.6**	**60.5**	**63.7**	52.8	**56.0**	44.4	**70.5**
(Epoch 1)	F1	**70.9**	**73.8**	**76.2**	**65.7**	**67.7**	62.9	**65.3**	**60.4**	**72.7**
	AvNA	**76.1**	**78.0**	**82.2**	**71.5**	**72.0**	69.4	**76.9**	**81.5**	**75.9**
RAS	EM	60.7	68.4	63.5	58.2	60.2	**72.2**	45.1	**48.1**	64.8
(Epoch 1)	F1	63.5	70.0	68.6	62.5	64.8	**76.1**	58.4	54.7	67.1
	AvNA	67.4	74.0	74.3	69.6	69.3	**80.6**	67.0	63.0	70.2
Baseline	EM	**71.8**	**78.4**	**70.5**	**66.5**	**66.7**	55.5	**49.5**	44.4	**72.7**
(Epoch 2)	F1	**74.9**	**80.1**	**75.7**	**70.4**	**71.8**	64.5	**59.8**	51.8	**75.0**
	AvNA	**79.0**	**83.1**	**79.3**	**75.7**	**76.0**	72.2	**69.2**	**66.7**	**78.1**
RAS	EM	66.3	71.9	60.1	60.8	65.5	**63.9**	52.7	**51.8**	71.2
(Epoch 2)	F1	68.9	72.7	67.0	64.3	68.5	**67.0**	60.3	**57.4**	72.7
	AvNA	72.6	76.1	72.7	69.6	71.7	**72.2**	64.8	63.0	75.9

Table 2: SQuAD 2.0 Dev results, by question type. Although we see an improvement in yes/no-style questions, this a small class of questions. Question type was determined using rule-based matching of key words within the question.

without data augmentation for the recommended 3 epochs [Devlin et al., 2018], in our comparisons, we use the results from two epochs.

0.4.3 Results

We think that because the underlying architecture of BERT, Transformers, rely on convolutional operations, removing sentences from the context does not positively affect learning since there is translational invariance of learning from the existing sentences of the context. We suspect that improvement in the yes/no question type are due to higher variance from yes/no sparsity in the dataset.

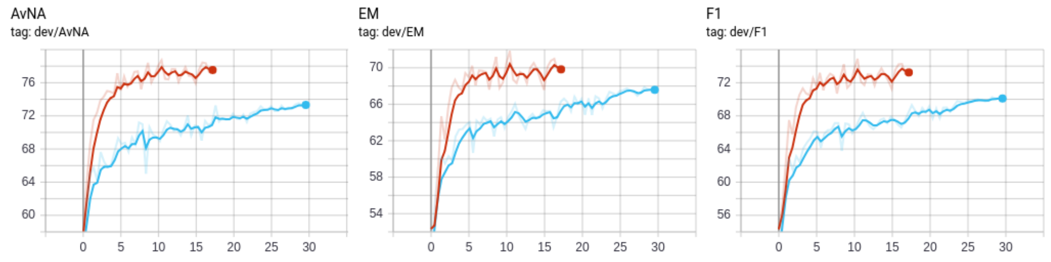

Figure 2: Improvements in AvNA, EM, and F1 over time. The red line is the baseline model. The blue line is the RAS model. Without early stopping, 3 epochs to train the baseline model took 17 hours and for the RAS model took 29.5 hours.

	Actual Positive	Actual Negative
Predicted Positive	2,139 (35.19%)	880 (14.48%)
Predicted Negative	771 (12.69%)	2,288 (37.64%)

Table 3: RAS Confusion Matrix

	Actual Positive	Actual Negative
Predicted Positive	2,450 (40.31%)	844 (13.89%)
Predicted Negative	460 (7.57%)	2,324 (38.24%)

Table 4: Baseline Confusion Matrix

Although we explored a pure classification model as well, even accuracy performance did not compare well with the baseline, even without RAS. Because of this degraded performance in just a classification task, we did not attempt to use the hidden embeddings from the final Transformer model.

Otherwise, if we let $T_i \in \mathbb{R}^H$ denote the hidden vector from the question answering BERT for the ith input token and let $C \in \mathbb{R}^H$ denote the hidden vector of the `CLS` token from the classification BERT, then in a combined model, two additional vectors $S \in \mathbb{R}^{2H}$ and $E \in \mathbb{R}^{2H}$ are learned such that the probability of the ith token being the start of the answer span is

$$P_i = \frac{\exp^{S \cdot [C;T_i]}}{\sum_j \exp^{S \cdot [C;T_i]}}$$

0.5 Analysis

0.5.1 Negative Predictive Rate

Since we wanted to fix the class distribution, our first points of analysis were the true positive and true negative predictions compared to a model trained on an unaugmented dataset. We had expected the true negative prediction to be higher since the augmentation increases the size of the negative class.

However, both true positive and true negatives were higher in the baseline model. The baseline predicted +311 (5.12% of the total samples) more of the true positives correctly and +38 (0.59% of the total samples) more of the true negatives correctly. We expected in absolute terms for at least the true negatives to be higher since we provided the RAS model more negative samples.

The negative predictive rate (NPR) for the RAS-augmented model was 75.80%, which was -8.38% than the 83.48% NPR of the baseline model.

0.5.2 "I Don't Know" Mentality

We suspected next that the model, having seen more negative examples, may be more likely to predict that an example does not have an answer. The RAS-augmented model predicted negative 50.33% of the time, which was more than the 45.80% for the baseline.

Because this suspicion was true, we wanted to understand the largest class of incorrect examples, the false negative group.

0.5.3 Shorter Context Association

Because of the nature of RAS-augmentation, the context distribution of negatives is shorter by at least one sentence from the original training set. The initial hypothesis is that because of this, the RAS-augmented model may be associating shorter context lengths with the negative class.

Even though the shorter contexts are seen in the training set, this pattern carries over in the predicted examples of the model itself. As seen in Figure 3, the distribution of predicted negative examples is shifted left and has 4.51 fewer WordPiece tokens in the context than the actual negative example distribution. We observe a similar pattern in the positive example distribution in which 4.48 more

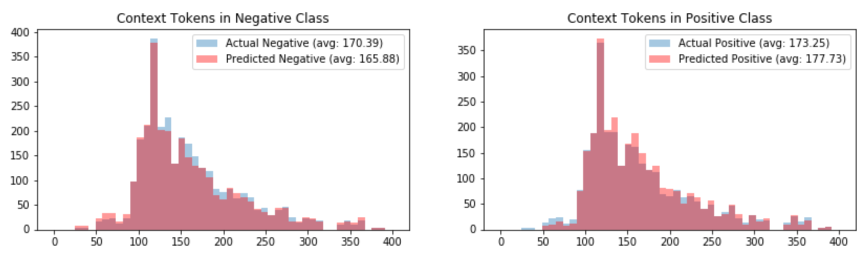

Figure 3: Distribution of context lengths within positive and negative classes.

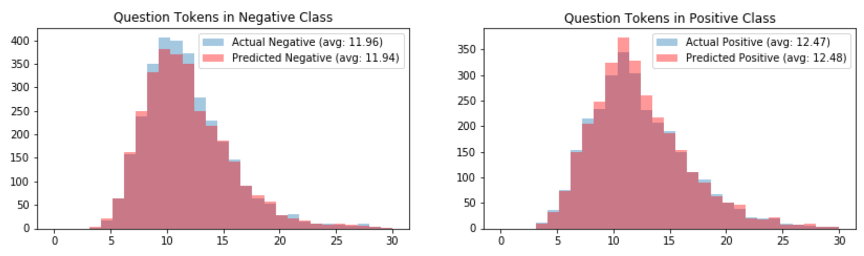

Figure 4: Distribution of question lengths within positive and negative class

WordPiece tokens are present in the predicted positive contexts compared to the actual positive context lengths.

As an additional check, we look at tokenized question lengths in Figure 4 and observe no significant difference.

This suggests that the RAS model associates shorter contexts with not being able to answer the question and that rather than simply removing a sentence, we should replace the sentence with another one of similar length. One simple way of doing this is finding a sentence of similar length from another context within the same topic. More advanced methods could explore using a generative model that provides a continuation sentence similar to adversarial generative methods used by Zellers et al. [2018].

0.6 Conclusion

With recent advances in pre-trained contextual embeddings, it has been possible to use the same base model with minimal fine-tuning for new NLP tasks. Although the neural architecture is important, the treatment of the dataset and pre-training is also important as shown by BERT. Using relatively simple augmentation schemes such as RAS may increase the number of training examples without human intervention, but we show we need to be careful since the augmented dataset alters the distribution of examples and does not generalize well.

In future work, we can explore whether augmentation can be used with the original SQuAD 1.1 dataset and attempt zero-shot learning on the SQuAD 2.0 dataset. Additionally because the context distribution length changes, context- or question-generating models can be used to keep distributions consistent while augmenting the original dataset.

References

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

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad, 2018.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. URL `https://arxiv.org/pdf/1706.03762.pdf`.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference, 2018.