Pseudo Labeling as a Method for Unsupervised Domain Transfer in Extractive Question Answering

Erik Leffler

lefflererik@gmail.com

November 17, 2020

Abstract

In this work we explore the effect of pseudo labeling as a means of unlabled domain transfer for Extractive Question Answering. More specifically, we use an ALBERT model, pretrained on SQuADv2, to evaluate pseudo labeling as a means of domain transfer to the biomedical domain. As a dataset for the biomedical domain we use a version of the BioASQ dataset that has been adapted for the MRQA shared task of 2019. We found that fine-tuning with 1200 pseudo-labled BioASQ examples interleaved with an equal amount of labled SQuADv1 examples yielded a performance increase of 4.0 and 3.05 percentage points in Exact Match (EM) and F1 score respectively relative to fine-tuning with only SQuADv1 examples.

I. Introduction

i. Transfer Learning and Domain Transfer

Recently, the field of Natural Language Processing has seen great success in the development of models and fine-tuning procedures that allow for transfer learning. Transfer learning refers to the process of obtaining a model that has been pre-trained on some general task, and then adapting it via further more specific training, to a different target task. Generally, transfer learning can be broken down into two subcategories, inductive and transductive transfer learning. Inductive transfer learning, also commonly called fine-tuning, refers to the case where the pre-training task differs from the target task. This usually require augmenting one or more top layers of the model before starting fine-tuning. Transductive transfer learning, which is what this paper is centered around, refers to when the data that was used for fine-tuning stems from a different distribution than the target data. In natural language processing, when the pre-training data is in the same language as the target data, but different in some way, we usually say that we are performing domain transfer. Consider for example a model that has been pre-trained for some specific task on text from Wikipedia. A practitioner might perform domain transfer before using the model to generate predictions from social media text. Here we call Wikipedia the source domain and social media the target domain.

ii. The Focus of This Paper

Transfer learning has significantly reduced the barrier for applying deep learning to new problems and domains by allowing efficient models to be developed with relatively small datasets. What is of further interest, and the focus of this paper, is how we can efficiently utilize unlabeled data as a means to perform domain transfer.

We picked Extractive Question Answering as a task to focus on for this paper. This decision was made somewhat arbitrarily, mostly because extractive QA is an intricate and interesting task. Our pre-trained model is an ALBERT model [4] that has been fine-tuned for extractive QA on SQuADv2 [8]. As a target domain we have picked research papers from the biomedical domain. Specifically, we use a version of the BioASQ dataset that has been adapted to the MRQA 2019 task [2]. This dataset is used for two reasons. First of, it has labels. This enables straightforward evaluation. Secondly, the MRQA adapted dataset is in the same format as SQuADv1 [9] , which is almost identical to the format of the dataset that our model has been fine-tuned for. We strip the labels from the BioASQ dataset for training and evaluate pseudo labeling as a procedure for unsupervised domain transfer.

iii. Background

iii.1 Transfer Learning and Natural Language Processing

For half a decade, transfer learning in the field of natural language processing was limited to the use of pre-trained word or subword embeddings [6] [7]. Whilst these provide a performance increase, they only make up the base layer of any model. The first successful transfer learning framework for NLP that offered a fully pre-trained model was presented in 2018 and given the name ULMFiT [3]. In the paper, the authors presents a novel approach based on using learning rate schedules that vary both for different epochs and layers, that allow users to fine-tune a pre-trained LSTM based model to different tasks.

It was right after the ULMFiT paper that BERT was introduced [1]. BERT is a transformer model [10] that was designed to easily enable transfer learning in NLP. In the paper, the authors demonstrate that the model can be augmented and fine-tuned to achieve, what was at the time, state of the art results in 11 different NLP tasks. In short, BERT is a transformer encoder that has been pre-trained on a joint task of masked language modelling and next sentence prediction. Masked language modelling is the task of predicting masked out words in a sentence. Next sentence prediction is the task of classifying whether two sentence follow each other in a piece of text. This pre-training task proved general enough that slight augmentations and fine-tuning of the model yielded state of the art results. Since this, there have been endless variations of BERT presented in the literature. We have chosen a to use a pre-trained ALBERT model [4] for this paper. This decision was made since it is relatively small in size, and performs well on extractive QA.

iii.2 Extractive Question Answering

Extractive Question Answering is a specific format of question answering where a model is given a context and a question and tasked with finding a span of the context that answers the question. All the datasets used in this paper are of this format.

A transformer can be fitted for extractive QA by attaching and fine-tuning a head that outputs two one-hot vectors. The first vector represent probabilities that each token in the context is the start of the span, the second vector represents probabilities that each token in the context is the end of the span.

iii.3 Pseudo Labeling

Pseudo labeling is a method that enables the use of unlabeled data for a traditionally supervised training objective [5]. The method is quite simple, it requires a model and unlabeled data and basically consists of labeling unlabeled data with the models predictions. Mathematically it can be written as

$$y_i^{pseudo} = \begin{cases} 1 \text{ if } i = \text{argmax}(f(x)), \\ 0 \text{ otherwise }, \end{cases} \tag{1}$$

where y^{pseudo} is a one-hot representation of the pseudo labels, f is the model and x is a data sample.

Pseudo labeling on it's own will usually result in overfitting [5]. One way to overcome this is to use a dataset that consists of samples with regular labels alongside the pseudo labeled samples. As this paper aims to investigate unsupervised learning, we use source domain data for the labeled examples and target domain data for the pseudo labeled data. Further flexibility can be introduced to training by letting the final loss be a weighted sum between the labeled and pseudo labeled loss. We can write this as

$$L = \alpha \cdot L(x^{unlabeled}, y^{pseudo}) + L(x^{labeled}, y), \tag{2}$$

where α is a new hyperparameter that specifies the weighting of pseudo labeled data samples.

II. Experiment

We use an ALBERT base model that has been fine-tuned on the SQuADv2 dataset as our starting point. For our target domain we have picked the biomedical domain. We use a version of the BioASQ dataset that has been adapted to the SQuAD format for the MRQA shared task of 2019. For training, we strip the labels from the BioASQ dataset. We attach pseudo labels to the BioASQ data that is updated before every forward pass. The BioASQ data is interleaved with equal parts labeled data from the SQuADv1 dataset. Thus there is a one to one relationship between the amount of pseudo labeled and regularly labeled data at each training batch and the loss weighting is controlled with the α hyperparameter. The training split of the BioASQ dataset consists of 1200 samples. These were repeated for each epoch. The SQuAD dataset is big enough that we could keep using different samples for each epoch, so we did. We use a linear learning rate warm up for the first third of epochs and a constant learning rate is used thereafter. The hyperparameters were not thoroughly tested due to limited resources and time. K-Folds evaluation was used to pick a reasonable learning rate and epoch count, this was not done for each value of α though. We trained the model for 20 epochs with a target learning rate of $3e-06$, a batch size of $8+8=16$, for values of α in $\{0, 0.08, 0.16, 0.32\}$.

i. Evaluation

To evaluate the model we use Exact Match (EM) and an averaged F1 over all samples calculated over included tokens. Both scores are calculated after text normalization. For some questions, there will be multiple possible answers. In these cases, the answer that grants the highest score will be used.

The EM score is simply the percentage of times that the model answer equalled a label answer.

The F1 score used here is adapted for machine comprehension and thus differs slightly from the F1 score which is often used in other areas of machine learning. As usual, we let the F1 score be the harmonic mean of recall and precision. However, in this case, these metrics are calculated as

$$\text{recall} = \frac{|L \cap M|}{|\text{L}|},$$

$$\text{precision} = \frac{|L \cap M|}{|\text{M}|},$$

where L, M are the sets of tokens that exist in the label and model answer respectively. Moreover, the F1 score is calculated for each data sample and then averaged over the whole dataset. The text normalization routine consist of the following steps,

- Convert text to lowercase,
- Simplify whitespace,
- Remove the following articles: *a, an, the*,
- Remove punctuation.

Note that since the model is restricted to extract an answer from the context text, the removal of articles will in most cases imply that we do not care whether the given answer is "answer" or "the answer".

III. Results

The evaluation metrics of models trained with different values of α is presented in table 1. For comparison, we also fine-tuned the same model as was trained with pseudo-labels, with real

Table 1: *Results*

α	EM	F1
0.00	32.36	47.28
0.16	34.54	48.53
0.32	**36.36**	**50.33**
0.64	19.63	31.41
Without domain transfer	23.64	33.68
With labels	**64.36**	**73.54**

labels. This serves as a ceiling for what performance is attainable. We also evaluated the model before performing any kind of domain transfer.

IV. Discussion

Table 1 exhaustively lists all results, and we shall refer to these bellow with out explicitly including a reference to the table.

i. No Domain Transfer vs $\alpha = 0$

As can be seen in equation 2, letting α be equal to 0 is exactly equivalent to performing further pre-training on the SQuADv1 dataset. Since the model was already fine-tuned on the SQuADv2 dataset, it comes as a surprise that we see 3.27 and 9.62 percentage points of increase in EM and F1 respectively when fine-tuning the model with α set to 0. One possible explanation for this is that the SQuADv2 contains questions that has no answer, whereas SQuADv1 and the MRQA adapted BioASQ does not. The further fine-tuning with SQuADv1 would discourage the model from outputting that the context doesn't contain an answer to the question. Quickly looking through the test data shows that almost half of the predictions made before domain transfer are that there is no answer in the context. For the $\alpha = 0$ case, no question is predicted as unanswerable. If more time was available, we would rerun the experiment with a model that has been fine-tuned on SQuADv1. This would be nice for completeness. We do believe however, that the $\alpha = 0$ case serves as a perfectly adequate benchmark for the remaining analysis.

ii. Performance Evaluation

When comparing the $\alpha = 0.32$ results to the $\alpha = 0.00$ results, we see that pseudo labeling gave rise to a 4.0 and 3.05 percentage point increase in EM and F1 respectively. Whilst certainly significant in situations where labels are not attainable, the results pale in comparison the 32.0 and 23.21 percentage point increase obtained from regular labeled domain transfer. For situations where labels do not exist, but may be obtainable, a practitioner should seriously consider the cost of data-labeling as an alternative to pseudo labeling.

iii. Likelihood of Unlabeled Extractive QA Data

In this work we used unlabeled extractive QA data. The likelihood that one is faced with a extractive QA task, for where there exists data that consists of contexts and associated questions, but no labels, seems slim. In how many situations are we presented with contexts and corresponding questions naturally? It would seem that for a lot of cases, the context and associated questions would have to be created or paired by a human in some form, in which case it would seem natural to have the human supply an answer. For this reason, we suspect that pseudo labeling might be an unlikely fit with extractive QA exactly, but might be useful for other NLP tasks.

iv. Other Methods for Unsupervised Domain Transfer

Another method that could be used for unsupervised domain transfer is multi-task learning. There is one method that comes quite naturally to mind when thinking about transformer models and NLP. Namely, a two-headed transformer model. Such a model would consist of shared base layers, one head that is built for whatever unsupervised task the model was pre-trained on, and one head that is built for the target task. The unsupervised head would be trained with target domain data, whilst the task specific head is trained with some out of domain task specific data. Training these two heads simultaneously would allow us to instil target domain knowledge into the base layers in an unsupervised way, at the same time as we teach the target task head to perform the task. What would be interesting to see is if teaching the base layers the structure of target domain data would help the task specific head make better predictions.

References

[1] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[2] Adam Fisch et al. "MRQA 2019 shared task: Evaluating generalization in reading comprehension". In: *arXiv preprint arXiv:1910.09753* (2019).

[3] Jeremy Howard and Sebastian Ruder. "Universal language model fine-tuning for text classification". In: *arXiv preprint arXiv:1801.06146* (2018).

[4] Zhenzhong Lan et al. "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942* (2019).

[5] Dong-Hyun Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2. 2013.

[6] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

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

[8] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know what you don't know: Unanswerable questions for SQuAD". In: *arXiv preprint arXiv:1806.03822* (2018).

[9] Pranav Rajpurkar et al. "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250* (2016).

[10] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.