Understanding Multitask Learning with Transfer Learning

Abhishek Sharma
sharma21@stanford.edu

Abstract

This study focuses on verifying the hypothesis behind multitask learning whether training related tasks jointly, improves the performance of the model when exposed to transfer learning. In particular, study focused on Question Answering (QA), Natural Language Inference (NLI) and Sentiment Analysis (SA) tasks and tried to understand the relationships between each of these by analyzing: i) the performance of the jointly trained model for the tasks in the same domain (for instance, SA and NLI both are classification tasks) on a pretrained model. ii) performance of the jointly trained model for the tasks in different domain (QA is span extraction problem which was trained with two different classification tasks) on a pretrained model. Contribution of the project is to help understand the relationship between SQUAD, MNLI and SSt-2. It was found that training tasks from the same domain, jointly, yields better results confirming the hypothesis behind multitask learning.

1 Introduction

Multitask Learning that aims at improving generalization by learning multiple tasks at the same time. It can be viewed as mimicking human learning activities as the world we live in requires us to know many things, we transfer knowledge from one task to another wherever these tasks are related to make it easy for us. DecaNLP [2], a challenge released by Salesforce research makes an attempt to solve 10 tasks simultaneously. Another area of research in NLP that has seen a surge recently is transfer learning. Bidirectional Encoder Representations from Transformers [1] is one such model whose representations can be used to train other models via fine tuning or through feature extraction.

This paper analyses the effect of different tasks on each other in the multitask setting over a pretrained model. The three tasks were chosen such that two of them are in the same domain and are classification tasks (Sentiment Analysis and Natural language inference), while the third task is from another domain (Question Answering) and is a span extraction problem. This allowed us to analyse how tasks behave with each other when trained jointly.

Tasks in the same domain were trained using the hard parameter sharing technique [3] [5], where some layers are shared by the model but the model also has task specific layers that emits the final output of the model. The tasks in the different domain were trained on different subsets of data for each task in a round robin fashion. Following sections cover the approach in depth that was used to train the tasks jointly and the comparison of the performance of those tasks with respect to the single task model which is nothing but fine tuned pretrained model for that problem.

2 Related Work

MQAN model [2] introduced as part of decaNLP is an architecture designed to solve 10 tasks. It treats every task as a question answering problem and extends the co-attention of [4]. The contribution of

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

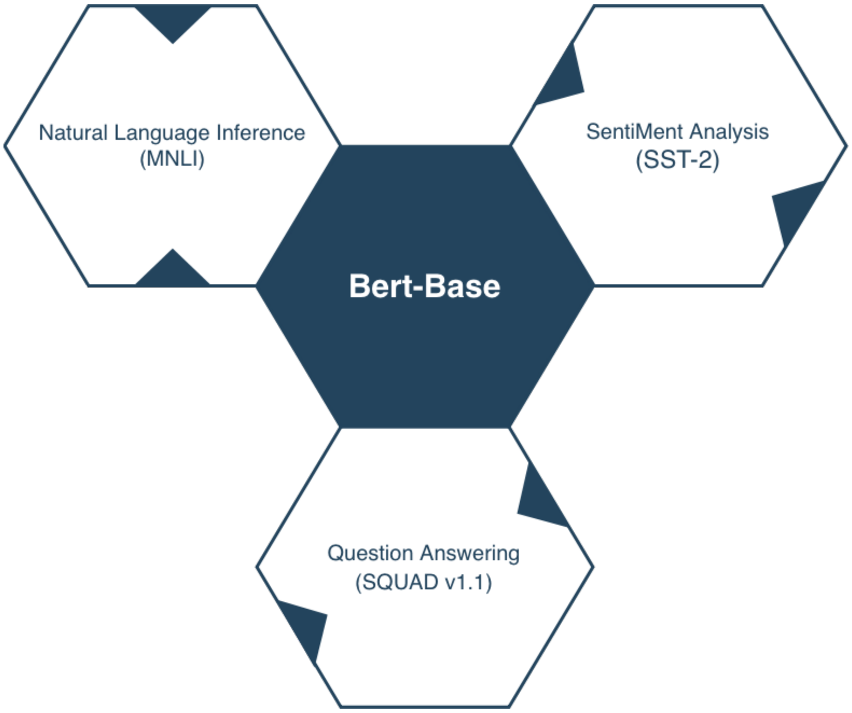

Figure 1: pretrained Bert base was fine tuned for different pairs of multitask models; (SA, NLI), (SA, QA), (NLI, QA)

MQAN model [2] is the usage of a multipointer-generator-decoder in the end that generates the word from the question, answer or a predefined target vocabulary.

The need for generating the word arises due to inclusion of Machine translation and summarization tasks. Since the tasks chosen for this study are classification and span extraction problems, multipointer-generator-decoder was not needed as per the project proposal which was proposed to be used inititally. Instead a pretrained model was fine tuned to solve the problems jointly.

3 Approach

The core idea is that a pretrained model was fine tuned for different pairs of multitask models; (SA, NLI), (SA, QA), (NLI, QA). This was done keeping in mind that pretrained Bert base model [1] when fine tuned for a particular task would have the encoder representation and to achieve multitask learning, this representation needed to be learn jointly. Figure 1 demonstrates this idea.

There were two approaches that were used to train same domain tasks and tasks in the different domain.

3.1 Same Domain Tasks

Sentiment Analysis and Natural language inference are classification problems and the strategy used here was hard parameter sharing technique [3], where the model shares some layers but has different layers at the end for each specific task. For the given tasks, SA and NLI, model shared the pretrained BERT base model [1] and two different linear layers were used in the end for each task for the classification, which allowed us to train SA and NLI simultaneously. The label for the task is passed as input to the model based on which model decides the appropriate softmax to use.

During training, label also helps calculating the appropriate loss which was backpropagated throughout the network. This setting allowed the model to learn the common representation.

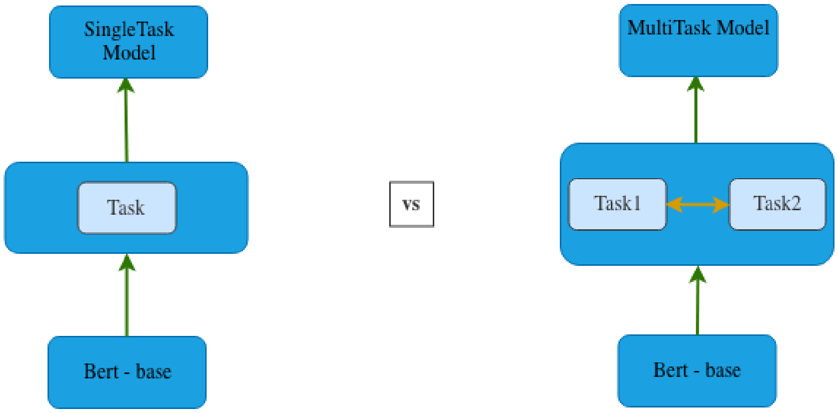

Figure 2: Single task models were compared with the multitask models

3.2 Cross Domain Tasks

Question answering was the third task and is a span extraction problem, here the technique used to train the multitask model was to train the model on QA for a portion of the dataset (50 %) and then train it for the classification task and then switch it back to training for QA. While making the switch from QA to classification, the pooled layers were stripped off and classification layer was added, this did not disturb the encoder representation learnt during fine tuning for QA. Similarly, when switching back to QA the classification layer was stripped off and the pooled layer was added for the QA that allowed to get the start and end indexes of the span. These switches allowed the model to learn common representation for the classification and the question answering tasks. During evaluation respective layer was added to the multi task model. This technique was one of the approaches used by [2] MQAN to tackle decaNLP.

The baselines that were used for the comparison were the single task fine tuned BERT[1] that was compared with the multitask model. Figure 2 demonstrates the comparison approach between the single task baseline and the multitask models.

Although Bert base was fine tuned for all the models and most of the code was provided by [9] for fine tuning BERT base [1], additional changes were made to modify the training of pretrained BERT in the multi task setting and for the analysis part.

4 Experiments

Experiments were conducted to train multitask models on Question answering, Sentiment Analysis and Natural Language Inference.

In Question Answering model is given a question and a context and it returns the span (start and end) of the answer in the context. The Dataset that was used for this task was: SQUAD v1.1 [6]. The evaluation metric used was F1 score.

Sentiment Analysis aims to find the polarity of the sentence. Dataset used was SST-2 [7] , binary classification. The evaluation metric used was EM score.

Natural LanguageInference uses a premise and a hypothesis and predicts whether premise contradicts, entails or is neutral wrt hypothesis. Dataset used for this was :Multi-Genre Natural Language-Inference [8]. The evaluation metric used was EM score.

The multi task models that were trained are: (SST-2, SQUAD), (MNLI, SST-2), (MNLI, SQUAD). Bert [1] was also fine tuned for the single tasks: SSt-2, MNLI, QA.

Table 1: Sentiment Analysis Evaluation: SST-2 dataset

Model	EM
SST-2	90.825
SST-2 + MNLI	**93.577**
SST-2 + SQUAD	**92.087**

Table 2: NLI evaluation on MNLI

Model	EM
MNLI	83.529
MNLI + SST-2	**84.264**
MNLI + SQUAD	83.311

4.1 Results

Table 1 summarizes the results that were obtained by evaluating the different models on the SST-2 dataset.

Table 2 summarizes the results that were obtained by evaluating the different models on the SST-2 dataset.

Table 3 summarizes the results that were obtained by evaluating the different models on the SST-2 dataset.

5 Analysis

The multitask model that was trained for SST-2 and MNLI did extremely well and the score improved to 93.577 compared with the baseline model's performance that was 90.825. This model also performed well on the MNLI dataset and the score improved to 84.264 wrt baseline which was 83.529. It can be safe to say that the tasks in the same domain help each other learn better or aid in each others learning.

On the cross domain front, (SQUAD, SST-2) model improved the score when compared to SST-2 and SQUAD baseline models as the score improved to 92.087 for SST-2 and 88.135 for SQUAD. (SQUAD, MNLI) didn't do well on the MNLI dataset and the score dropped a little to 83.311 from 83.529. Although, the reason behind this could be many but we can say that SST-2 and SQUAD helped each other learn better representations.

6 Conclusion

Multitask models for tasks in different domain showed some improvement (two out of three) whereas all multitask models for the same domain tasks showed great improvement that proving our hypothesis that training similar tasks jointly does aids to learning.

Collaborators

I would like to thank my mentor Michael Hermann Hahn for his constant guidance and support throughout the project. Also, special thanks to the all the TA's for all the motivation and help provided.

Table 3: QA evaluation on SQUAD v1.1

Model	EM
SQUAD	87.398
SQUAD + SST-2	**88.135**

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805.

[2] McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). *The natural language decathlon: Multitask learning as question answering.* arXiv preprint arXiv:1806.08730.

[3] Caruana, R. 1998.*Multitask Learning. In Learning to Learn.* Springer. 95–133

[4] C. Xiong, V. Zhong, & R. Socher. it Dynamic coattention networks for question answering. ICLR, 2017.

[5] Ruder, S. (2017). *An overview of multi-task learning in deep neural networks.* arXiv preprint arXiv:1706.05098.

[6] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *Squad: 100,000+ questionsfor machinecomprehension of text.* arXivpreprintarXiv:1606.05250.

[7] Socher, R., Perelygin, A.,Wu, J.,Chuang, J., Manning,C. D., Ng, A., & Potts, C. (2013). *Recursive deep models for semantic compositionality over a sentiment.*

[8] Nangia, N.,Williams, A., Lazaridou, A., & Bowman, S.R. (2017). *Therepeval 2017 shared task: Multi-genre natural language inference with sentence representations.* arXivpreprint-arXiv:1707.08172.

[9] *Pretrained PyTorch models for Google's BERT, OpenAI GPT & GPT-2, Google/CMU Transformer-XL.* `https://github.com/huggingface/pytorch-pretrained-BERT`