The Role of Data in Bert Question-Answering

Xian Wu

Abstract

This project explores how the performance of the Bert Question-Answering system responds when fine-tuned on a variety of data sources. We will attempt to fine-tune Bert pretrained embeddings on a variety of different tasks in addition to Question-Answering on SQuAD, and then evaluate on SQuAD, thereby assessing how exposure to different types of additional data can change performance on a target task. If time permits, we will also attempt some data augmentation on the SQuAD training set in order to assess how availability and quantity of training data affects performance.

1 Introduction

In this work, we aim to understand the influence of data and how training for multiple tasks affects performance on a specific task of focus, which is Question and Answering in the SQuAD dataset. Our specific focus is not on creating new architectures or models, but rather we want to understand how performance on the SQuAD dataset changes when the underlying basic building blocks, embedded representations of words, have been exposed to different datasets and have been trained to perform other tasks as well before being trained to answer questions.

Our perspective is not new, many others have emphasized the importance of data in building artificial intelligence, as well as the importance of models that can simultaneously and effectively perform many related tasks. We take a small step along this direction by focusing on contextual representations introduced [DCLT18], which are the building blocks for recent models that have achieved state of the art results in many NLP tasks, and understanding how fine-tuning these contextual representations for a variety of related NLP tasks affect performance on the SQuAD task.

One recommended practice for creating a representation-based NLP solution is to take pre-trained embeddings and fine-tune them on the specific task that one is interested in. Leveraging pre-trained embeddings is arguably a better idea than creating one's own representations from scratch because in the former, one is able to benefit from a warm-start and initialize their training with embeddings that have been already been trained on massive corpuses of data and with massive amounts of computation. These weights often already represent a good model of the language, and so it suffices then to build architecture on top of these pre-trained weights to fine-tune these weights to the specific task that one is interested in, as well as create a model that will give the kinds of outputs specific to the target task.

Our study builds upon this recommended practice. We study the effect of first fine-tuning the pre-trained weights on one task, and then fine-tuning the resulting model for SQuAD, versus fine-tuning the pre-trained weights directly for the SQuAD task. We analyze the effects of exposing and training the contextual representation weights to different datasets curated for different tasks and then learning to answer questions on SQuAD. Our study touches upon broader ideas in machine learning, including transfer learning and multi-task learning.

We emphasize that the goal of this study is not to build complex models to outperform as much as possible, the state of the art on the SQuAD dataset, but rather to explore how this training technique enhances of adversely affects single-task fine-tuning. These conclusions can then be adapted to more complex models to achieve even better results.

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

2 Related Work

Data is an important ingredient in the training and learning process of a good artificial intelligence model. This point of view is the driving force behind research that aims to collect and assemble quality datasets, for example, Imagenet [DDS+09] and SQuAD [RZLL16]. Well curated datasets are essential in helping scientists produce and test machine learning models. Well curated datasets are evaluated on the basis of quantity, diversity, lack of bias, and other factors [Har15] that all contribute to how representative the dataset is of the world in regards to the problem it aims to tackle. Our work is related to the idea of exposing a single to different types of datasets, designed to train for different tasks and how this additional training improves or weakens performance on the target tasks. The closest areas that attempt to address this angle include Transfer Learning and Multi-task learning.

Transfer learning is inspired by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions [PY10]. Our work studies how learning to perform one task can more efficiently enable the model to learn how to do well on SQuAD. [MMY+16] performs some experiments for transfer learning specifically for NLP tasks. They find that learning transfers very well for semantically equivalent tasks but performance degrades for tasks that are not semantically equivalent.

[MKXS18] explores the idea of creating a model capable of doing well universally among many NLP tasks, with the tasks being categorized around the types of outputs they aim to generate and whether the output vocabulary comes from the context or the question or from the general vocabulary. For example, the output of a sentiment analysis is taken from the question, is the sentiment positive or negative, etc, whereas the output of SQuAD, for example, is a subsequence of text from the given context. [MKXS18] builds a model that takes as input a question (which essentially asks the model to perform a variety of NLP tasks) and a context and this model internally decides how to form the answer to the question. They experiment with different types of training processes, with anti-curriculum training, the process of training first the hardest tasks and then fine-tuning for the easy tasks, performing the best. For each task, they compare their multi-task model against a similar model that was exclusively trained for the one task. Their results for SQuAD is actually quite discouraging, their multi-task models are consistently inferior to the single-task SQuAD models. For other tasks, however, they are able to achieve gains in performance via multi-task training.

Puzzled by inconsistencies around multi-task learning in NLP applications and architecture designs, [BS17] identifies data characteristics and patterns in single-task learning that predict task synergies in deep neural networks. The goal is to understand why and when multi-task learning will work well in NLP domains. Their conclusion is that multi-task learning is more likely to help in target tasks that quickly plateau with non-plateauing auxiliary tasks, where the plateaus reference the learning curves in the optimization. Labels are also important to consider, particularly the label entropy between the main and the auxiliary task.

More work in the space of multi-task learning and transfer learning include [LLS+15], which argues that the efficacy of these methods require large datasets for the target task, much larger than for the auxiliary task, while [BMH17] suggests that multi-task learning is particularly effective when we only have access to small amounts of target data, as studied through an application in healthcare. [AP16] suggest that success depends on the distribution of the auxiliary task labels.

3 Approach

We will start with the standard 'bert-base-uncased' pre-trained weights from the huggingface repository. We will train a baseline model from the example squad script that the repository provides from the pre-trained weights. This will serve as our baseline model.

We will also experiment with fine-tuning the 'bert-base-uncased' pre-trained weights on other example tasks that the repository provides, save that model, and then fine-tune that model on squad and compare the EM/FM scores. We will do this on a variety of tasks that the example folder provides, including a classification task (using the Microsoft Research Paraphrase Corpus), a multiple choice task from SWAG, and a language-model fine-tuning script.

Lastly, we perform some naive data augmentation on the SQuAD training set that we are given, namely we train on this dataset twice and evaluate any differences in performance.

The goal is to see if fine-tuning on a variety of other related tasks and exposure to different types of NLP-related data can improve performance on the target task, SQuAD.

4 Experiments

4.1 Metrics

We use the standard EM/F1 metrics that the CS224N leaderboard uses. EM stands for exact match, and is a binary measure of whether the system output matches the ground truth answer exactly. The F1 metric is the harmonic mean of precision and recall, and is a softer metric that assigns scores between 0 and 1. When a question has no answer, both the F1 and EM score are 1 if the model predicts no-answer, and 0 otherwise. The scores are averaged across the entire dataset to give the final EM/F1 scores.

4.2 Baseline

Our baseline model is the direct integration of the huggingface repository run_squad.py example training script with the SQuAD training set as provided by CS224N. There is one small integration issue which is that the default settings in the training script are a little bit more resource intensive than what the virtual machines (NV6) that we are using can provide. So we experimented with some settings and have trained several baselines and will use the best baseline as our actual baseline.

The first setting is using the default settings but with gradient accumulation of 10. This model achieved very low EM/F1 scores (both in the 20's) and actually were not able to outperform the expected results of the baseline CS224N SQuAD model.

Our second experiment used gradient accumulation of 5, and this model achieved EM/F1 scores of 67.391/70.917, respectively. All other settings are based on the default in the provided script.

Our last setting follows the example run command on the huggingface repository's readme (so the settings are slightly different than the default in the script), except with a batch size of 6. This model performed a few points worse than the second model. Therefore, we chose the second model's parameters to be the training parameters we use for SQuAD training for the rest of the experiments.

4.3 Training a Classifier on the GLUE dataset

The huggingface repository offers a fine-tuning example script that works with the Microsoft Research Paraphrase Corpus. This is a dataset for a binary classification task. It is a text file with 5,800 pairs of sentences which have been extracted from news sources in the web, along with human annotations indicating whether each pair captures a paraphrase or a semantic equivalence relationship. We take present a positive example from their training file:

```
Amrozi accused his brother , whom he called " the witness " , of deliberately
    distorting his evidence . Referring to him as only " the witness " , Amrozi
    accused his brother of deliberately distorting his evidence .
```

as well as a negative example:

```
That compared with $ 35.18 million , or 24 cents per share , in the year-ago period
    . Earnings were affected by a non-recurring $ 8 million tax benefit in the year-
    ago period .
```

This is actually quite an easy task to train for, the training took very little time, and the results of our fine-tuning on the Bert pre-trained model achieved 83% accuracy. Then we trained the resulting model again on the SQuAD dataset using all default parameters and gradient accumulation equal to 5. This model performed slightly worse than just training on SQuAD, with EM/F1 scores of 66.305 (-1.086) / 69.589 (-1.328), respectively, which are a few points below the baseline.

It is interesting that the results turned out this way, because this task seems very similar to the pre-training task referred to in the original Bert paper, which is that of Next Sentence Prediction. [DCLT18] shows that pre-training towards Next Sentence Prediction is beneficial to QA.

4.4 Training on SWAG

SWAG stands for Situations With Adversarial Generations [ZBSC18]. It consists of 113k multiple choice questions about grounded situations. Each question is a video caption from LSMDC or ActivityNet Captions, with four answer choices about what might happen next in the scene. The correct answer is the (real) video caption for the next event in the video; the three incorrect answers are adversarially generated and human verified, so as to fool machines but not humans.

This task is actually very similar to the Next Sentence Prediction task as described in [DCLT18]. The differences are that SWAG is designed to help solve the problem of adversarial inputs and so the options for the next sentence are often perturbations of the correct answer or a possible correct answer, whereas in the Next Sentence Prediction Task, oftentimes the next sentence could be very different from the first sentence.

This task was harder to train, and took many hours. We fine-tuned on this task on the Bert pre-trained model and achieved 80% accuracy on the prediction task. Then starting from the new model that was trained on the SWAG dataset, we fine-tuned on the SQUAD dataset. Our new model achieved EM/F1 scores of 65.548 (-1.843) / 69.221 (-1.696), respectively, which are a few points below the baseline.

4.5 LM Fine-Tuning

We did another experiment where we first fine-tuned on a language model. There is no task, we simply train on a corpus of text to predict the next word in the corpus. We use the dataset that huggingface recommends on their GitHub website that is a text file from wikipedia articles, split into 500k sentences with spaCy. This corpus is actually in German, so we tested the effects of language model training in one language (German) affects question answering in a different language (English), using the same weights.

We first fine-tune on the LM dataset and we fine-tune the obtained model on SQuAD. Our new model achieved EM / F1 scores of 64.725 (-2.665)/68.262 (-2.655), which is worse than our benchmark model. We note that this experiment resulted in a model that performed worse than all the other multi-task fine-tuned models (down from the baseline by over 2 points, as opposed to other models that were down by only 1 point). We conjecture that this might be due to the fact that our language model corpus is in a different language (German), but we are not sure.

4.6 Fine-Tuning twice on SQuAD

We performed another experiment where we took our baseline model (fine-tuned on SQuAD) and we fine-tuned it again on the SQuAD dataset, as a naive way of augmenting our own dataset. The resulting EM/F1 scores were 64.775 (-2.616) / 68.662 (-2.254), which is a little bit down from just fine-tuning once on SQuAD. We conjecture that this is due to the model overfitting too much on the data, so perhaps this method of data augmentation does not work well.

4.7 Summary of Experiments

We summarize our experiments with the following table:

EM / F1 Results on Dev Set		
Dataset / Task	EM Score	F1 Score
SQuAD only	67.391	70.917
MRPC + SQuAD	66.305	69.589
SWAG + SQuAD	65.548	69.221
LM + SQuAD	64.725	68.262
SQuAD + SQuAD	64.775	68.662

4.8 Results on the Test Set

We submitted our best model (according to the dev set), which is a model just fine-tuned on SQuAD and achieved an EM/F1 score of 72.849 / 76.209 on the test set. We submitted under the name “final-blue".

5 Analysis

First we analyze the number of empty / no answer responses produced by each model. The differences between them are quite high, with the baseline (best model) returning 2159 no answer responses, MRPC + SQuAD returned 2126 no answers, SWAG + SQuAD returned 2050 empty strings, LM + SQuAD returned 2044 blanks, and the SQuAD + SQuAD produced 2021 blanks. So actually it seems that the better models had more no matches than the worse models, so maybe the worse performing models were overly optimistic in trying to match answers to the text.

This first observation is highlighted in the following example. The question is:

```
Who gave their name to Normandy in the 1000's and 1100's
```

The correct answer is that this question is unanswerable from the context. However, the baseline model and the MRPC + SQuAD predicted *Normans*, the SWAG + SQuAD model predicted *The Normans* and the last two models predicted no answer.

There are also some questions for which all the models across the board were overly optimistic in producing answers for. One example is:

```
Who did King Charles III swear fealty to?
```

The responses for the five models were *West Francia* for baseline, SWAG + SQuAD, LM + SQuAD, and SQuAD + SQuAD, and *Charles III of West Francia* for MRPC + SQuAD. In response to this "who" question, most of the models gave an answer that was a region, and only the MRPC + SQuAD answered with an actual person, however, all the models were wrong, this question was not answerable from context.

We also notice that oftentimes the weaker performing models were more verbose in their answers, which could have potentially harmed their score. One example is:

```
Who ruled the duchy of Normandy
```

The responses for the five models were *Richard I* for baseline, SWAG + SQuAD, MRPC + SQuAD, and *Richard I of Normandy* for LM + SQuAD, and SQuAD + SQuAD. The human responses were all *Richard I* so the LM + SQuAD and the SQuAD + SQuAD models were more verbose and therefore achieve lower scores for EM even though they are perfectly fine answers.

Overall, it seems that the differences in the final results are coming from instances where the question is unanswerable. We think that further fine-tuning to specifically target and handle these cases better would yield improvements in the model.

6 Closing Remarks

In this project we explored questions around whether a single-task model benefits from fine-tuning on other datasets and other tasks as well. This question is inspired by the human being's capacity of making connections between different areas and different topics and using experiences from related tasks to enhance their performance on their target task. We explore this idea by fine-tuning pre-trained contextual embeddings on another task with a different dataset before then fine-tuning on SQuAD, which is our target task of focus. We compare the performance of these models against the performance achieved by a model that is fine-tuned only on SQuAD. Our results show that single fine-tuning still produces the best model. Further work includes understanding how to build and train models that are capable of emulating humans in this way of learning.

References

[AP16] Héctor Martínez Alonso and Barbara Plank. When is multitask learning effective? semantic sequence prediction under varying data conditions. *arXiv preprint arXiv:1612.02251*, 2016.

[BMH17] Adrian Benton, Margaret Mitchell, and Dirk Hovy. Multi-task learning for mental health using social media text. *arXiv preprint arXiv:1712.03538*, 2017.

[BS17] Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303*, 2017.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Har15] Eszter Hargittai. Is bigger always better? potential biases of big data derived from social network sites. *The ANNALS of the American Academy of Political and Social Science*, 659(1):63–76, 2015.

[LLS+15] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[MKXS18] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[MMY+16] Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*, 2016.

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[RZLL16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[ZBSC18] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.