Fine-tuning BERT-inspired Deep Learning Models for Question-Answering

Yakoob Khan
Dartmouth College
yakoob.khan.21@dartmouth.edu

Abstract

In this paper, we explored how various BERT-inspired transformer models perform on a question-answering task. We utilized SQuAD dataset version 1.1 and the performance of models is measured using two metrics – the F1 and Exact Match (EM) scores. While SOTA models have amazingly surpassed human performance standards, training such models require immense computational power and time that is unavailable to the general public. Given these constraints, our project novelty lies in fine-tuning different high-performing pre-trained transformer models that anybody can do on a single free GPU under **2.5 hours**. We tested three deep learning models (BiDAF, BERT and RoBERTa) that achieved a maximum F1 score of **84.86%** and EM score of **76.09%**. Our project concludes that the **RoBERTa** model offers the best accuracy-to-training time ratio for the SQuAD question-answering task[1].

1 Introduction

Question-Answering (QA) is a downstream task in natural language processing that focuses on the developing language models and systems that can automatically answer questions posed by humans. Traditional rule-based approaches to solving the QA problem involve building parsing trees, knowledge-graphs, etc. Given the challenge of modelling language and answering complex queries, such classical approaches experienced limited success, especially in use cases where data is plentiful.

In recently years, deep learning-based approaches have been shown to be successful on many language tasks. The success of generalized pre-trained language models on large corpus of texts have revolutionized the field of NLP. One can utilize Transfer Learning, the idea of taking a generalized language model trained on a large corpus of text and apply fine-tuning to one's particular NLP task. The success of this approach has enabled the creation of robust QA systems that have paradoxically surpassed human performance, inviting philosophical discussions of machines beating humans.

2 Related Work

In the past, language models relied on Recurrent Neural Networks (RNN) to model language. Such models were very ineffective in modelling long-term dependencies due to the vanishing gradient problem. Long-Short Term Memory (LSTM) models provided limited improvements to overcome this problem.

The discovery of the transformer architecture significantly improved the modelling of long-term dependencies. In 2018, the invention of the **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) inspired a flurry of models

[1] Full source code is available here: https://github.com/Yakoob-Khan/squad

that improved language modelling in a wide range of benchmarks. The novelty of using the bi-directional attention mechanism marked an important advance in neural architectural models. Since then, numerous variants of BERT-inspired models have been developed and continue to improve state-of-the-art results in QA. For instance, the current SQuAD leaderboard models noticeably make use of ALBERT, or A Lite BERT, a more efficient variant of the BERT model. One also notices that ensembling, or combining, different models is a common technique used in top-scoring QA models to improve performance. In fact, 8 of the current top 10 models use the ensemble approach.

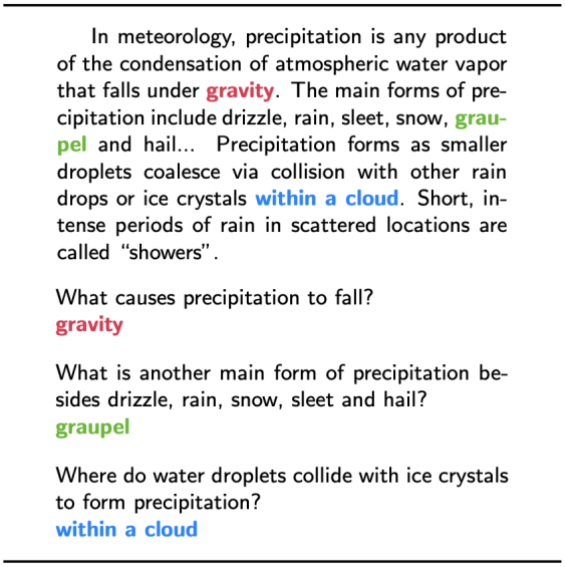
While improvements of BERT-based models are encouraging, recent SOTA models such as XL-net contains hundreds of millions of parameters that poses a serious computational problem. Training such large language models requires immense time and computational resources that is unavailable to the general public. Recognizing this inequity, the novelty of this project lies in exploring how high performing pre-trained language models can be utilized for solving downstream NLP tasks like the fundamental QA problem. Solving this problem is an important step towards developing Natural Language Understanding (NLU) that is the heart of mastering language.

3 Dataset

We chose the Stanford Question Answering (SQuAD) dataset for training our model. This dataset is commonly utilized in the QA domain as a benchmark and thus makes it a suitable choice. It contains around 100,000+ questions posed by crowdsourced workers on a collection of Wikipedia articles, and each question is answered by three different people to minimize human bias.

There are two versions to this dataset. Every question in version 1.1 of SQuAD contains a valid answer while version 2.0 includes an additional 50,000 questions whose answers do not exist. Version 2.0 is clearly a better dataset as unanswerable questions make the QA task more realistic. However, when we tried to train our model using version 2.0, we exceeded Google Colab's GPU memory limits and was unable to train the model. As such, we resorted to using the smaller version 1.1 dataset.

The following is an example of a <question, context, answer> tuple from the dataset:


In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Fig 1: Sample training example. Image credit: Rajpurkar et. al.

The data is provided in the form of JavaScript Object Notation (JSON) files. Around 80% of the data is split into the **train** dataset and the remaining 20% is the **dev** dataset, used in validating and evaluating the performance of the model.

Before running the model, data pre-processing is required to tokenize the English sentences into word embeddings. We used 300-dimensional pre-trained GloVe embeddings to represents the words in the baseline model. For the improvement models, we embed the words using the respective tokenizer provided by the Hugging Face transformers library.

4 Evaluation Metrics

The following description of accuracy metrics is taken from Stanford CS 224 default project description document.

- **Exact Match (**EM) is a binary measure (i.e true/false) of whether the system output matches the ground truth answer exactly. For example, if the system answered a question with "Einstein" but the ground truth was "Albert Einstein", then the EM score would be 0 for this example. Clearly, this is a fairly strict metric.
- **F1 score** is a more lenient metric and is the harmonic mean of precision and recall. In the

"Einstein" example, the system would have 100% precision (its answer is a subset of the ground truth answer) and 50% recall (it only included one out of the two words in the ground truth output), so the F1 score is 66.67%.

5 Methodology

We used Jupyter Notebooks to write all our code for training and fine-tuning our deep learning models. In order to measure improvement, it's crucial that we first establish a baseline performance that we aim to beat in this project. While simple classifiers such as SVM and logistic regression are typical baseline models, it did not seem fair to use such classifiers as baselines as our project aimed to compare the relative performance of BERT-inspired models. We settled on using the Bi-Directional Attention Flow (BiDAF) model as its novel use of bi-directional attention inspired the invention of BERT itself.

After establishing a baseline, we investigated training many different BERT-like models. We ran into significant limitations here as many models are far too big to run on a single free GPU. In the end, we were able to successfully fine-tune the BERT and RoBERTa models by compressing the models using knowledge distillation technique.

In all of our experiments, we kept the hyper-parameters at their default values to compare the relative effect of neural architectural modifications on performance. Additionally, we used the Mean Squared Error (MSE) as the loss function for all of our experiments.

6 Model Architectures

Before discussing our results, we felt it is useful to provide a brief description of the models we used in this project.

6.1 Baseline Model

To establish the baseline, we used the Bi-directional Attention Flow model created by Seoe et. al in 2017 and code provided by Stanford's CS 224N course.

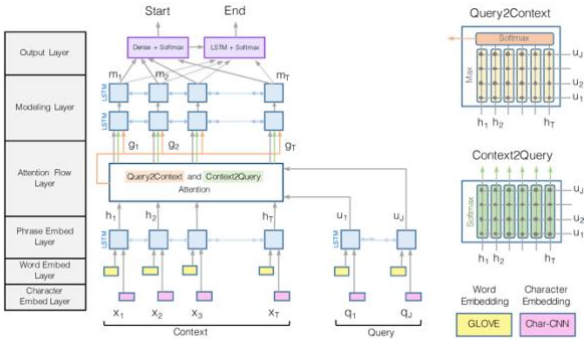


Fig 2: BiDAF model. Image credit: Seo et. al.

The BiDAF model consists of a series of layers and is composed of three main parts. First, there is a series of embedding layers whose function is to embed the query and context into word vector representations. Second, it contains attention and modelling layers that makes use bi-directional attention – from Query2Context and from Context2Query – to transform the representation into something called "query-aware context representation". Finally, this representation is passed through an output layer that is a SoftMax to finally produce an answer to a given query.

6.2 BERT

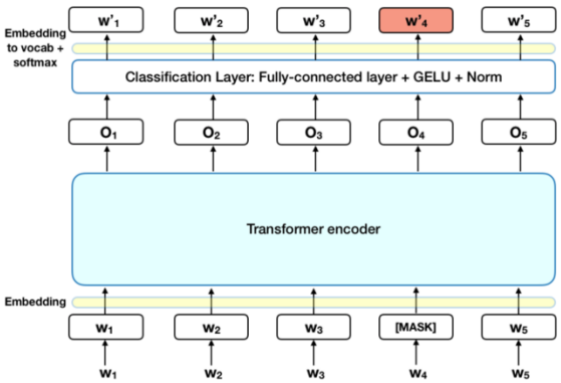


Fig 3: BERT model. Image credit: Rani Horev.

While BiDAF uses a bi-directional attention mechanism, BERT improves the model training process by two technical innovations. First, before the word sequences are processed by BERT, 15% of the words in each sequence is substituted with a [MASK] token. This self-supervised technique is known as Masked Language Modelling, designed to train the model to predict masked words from its surrounding context. Second, Next Sentence Prediction is used where the model receives pairs of sentences and is tasked at predicting whether the second sentence follows the first one. These two techniques resulted in the creation of a generalized

language model that performs well in a range of downstream NLP tasks such as QA.

6.3 Distil-BERT

Created by Hugging Face, this is a "smaller, faster, cheaper and lighter" BERT model that reduces the size of the original model by 40% while retaining 97% of its language understanding capabilities. We used this implementation of BERT for our improvement experiment.

6.4 Distil-RoBERTa

Created by Facebook AI, RoBERTa is a retraining of BERT with improved training procedures and uses 160GB text corpus, outperforming the original BERT model on many task benchmarks. Like distil-BERT, we relied on Hugging Face's implementation of RoBERTa that utilizes the distillation process.

7 Results

We summarize the results of all our experiments below:

Model	F1 (%)	EM (%)	Training (hrs)
BiDAF	78.98	69.73	2.5
BERT	**84.86**	75.94	5
RoBERTa	84.74	**76.09**	2
Human	89.452	86.831	
SOTA	93.011	90.724	

Fig 4: Table summarizing the performance of different models.

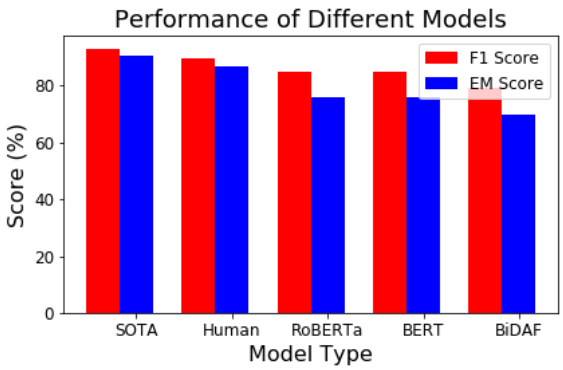


Fig 5: Table visualizing the performance of different models.

8 Analysis

A number of observations can be made from the results table. One notices that the EM score is always lower than the F1 score. This makes sense as the exact match metric is very strict and penalizes any missing or extraneous words in the answer to a query. Our BiDAF model establishes a strong baseline performance that challenges other models. The BERT model increases the F1 score by 5.88% and EM score by 6.21%, a significant improvement. Similarly, the RoBERTa model increases the F1 score by 5.76% and EM score by 6.36%.

Thus, the BERT model produces the highest F1 score while the RoBERTa model produces the highest EM score. However, the training time of RoBERTa is half that of BERT, suggesting that the former model offers the best performance-to-training-time ratio. We have also provided the human and SOTA performance for context. It's notable that SOTA has already outperformed humans in the QA task.

9 Future Work

While we experimented with knowledge distilled BERT models for the SQuAD task, there are a lot of ways for extending this work. Our results reveal that different models are suitable for maximizing either F1 or EM score, suggesting that an ensembling approach of stacking different models and averaging predictions should improve the result. This approach has been successfully utilized as 8 of the top 10 models on the SQuAD leaderboard use this technique. One could also perhaps experiment with data augmentation techniques to increase the size of the training set and improve performance. Finally, if one has access to more computing resources, one could train larger models such as ALBERT and XL-Net.

10 Conclusion

Overall, our project discovered that the Distil-RoBERTa model works best for the SQuAD QA task given limited computational resources. We concur with our hypothesis that pre-trained distilled models have democratized the field and allows anyone to apply deep learning NLP solutions to their particular use case or application.

Acknowledgements

I would like to thank Professor Rolando A. Coto-Solano for introducing me to the field of computational linguistics in Spring 2020. I'm also grateful for Stanford CS 224N course for guidance on fine-tuning their implemented baseline model and Hugging Face for open sourcing their implementations of BERT models.

References

Rajpurkar, et al. 2016. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. Conference on Empirical Methods in Natural Language Processing.

Seo, et al. 2017. *Bidirectional Attention Flow for Machine Comprehension.* International Conference on Learning Representations.

Sanh, et al. 2019. *DistilBERT: A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter.* 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing, NeurIPS.

Lir, et al. 2019. *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* Facebook AI.

Manning et. al, 2020, *CS224N Default Project: Question-Answering on SQuAD 2.0.* Stanford Univeristy.

.