True/False Question Answering: Comparing simple transformers with state-of-the-art mega-models

UC Berkeley School of Information Master of Information in Data Science
DATASCI266 - Natural Language Processing with Deep Learning
Summer 2023, Section 3, Jennifer Zhu
Andrew Higgins

Abstract

Extractive closed-domain question answering tasks in natural language processing involve the reading comprehension of a passage of text so that a question may be answered about it. This paper explores a sub-category of such tasks: boolean, or yes/no, question answering. Models BERT and T5 and some of their pre-trained variants are compared with the state-of-the-art, ST-MoE from Google. Results show that pretraining BERT is able to marginally improve its performance, but the biggest success comes from a variant of T5 trained specifically for boolean question answering.

Introduction

Question Answering (QA) in natural language processing is a fundamental task in which a model aims to answer user questions correctly and coherently. The state-of-the-art for QA tasks has evolved quickly in recent years, due to innovations in transformer-based architectures. One type of QA task is yes/no question answering, in which a model is given a text passage and asked a binary question about it. This task is simple yet quite difficult to get right. This paper compares a few of the fundamental building block models that make up today's state-of-the-art QA models.

State-of-the-art language understanding models, as discussed in the Background section, perform well across many question answering tasks, and ST-MoE from Google is no exception. As it pertains to this paper, ST-MoE performs the best at yes/no QA, achieving 92.4% accuracy when benchmarked against the BoolQ dataset. This is the gold standard upper limit against which to compare the models in this paper.

Transformer models are the basis of the state-of-the-art, and this paper examines a few of them: T5, RoBERTa, and BERT. The goal is to better understand how each model performs, and where it underperforms, to understand why.

Background

Question answering tasks can be categorized by the domain of knowledge that a model can use to answer the question. There are two domain-specific types: open-domain and closed-domain. This paper's task of yes/no question-answering is closed-domain. Open-domain tasks require a model to answer general information questions by retrieving relevant data from an external knowledge source. Closed-domain QA tasks involve reading comprehension questions about a provided text input, without any external retrieval. Answering yes or no to a question about a block of provided text, with no external information retrieval, is therefore a closed-domain task.

QA language tasks can also be categorized by the nature of the answer produced. There are two main methods for answer production: extractive and abstractive. Yes/no QA is an extractive task. Abstractive tasks require the model to generate an answer sequentially, usually using an encoder-decoder transformer architecture. Extractive tasks require identifying the answer directly from the reference text or some set of provided multiple choice questions. The BoolQ dataset used in this paper is structured with [question, passage, answer] triplets, and the resulting task is to extract the correct answer from the provided passage.

The state-of-the-art language model for closed-domain, extractive, yes/no question answering is ST-MoE, Switch Transformers Mixture-of-Experts, developed at Google Brain (Zoph and Bello, et.al., 2022). Indeed, it is the state-of-the-art model for many language understanding tasks, even when down-scaled to the same compute requirements as the next-best. ST-MoE improves upon the Google team's own previous state-of-the-art work (Fedus and Zoph, et. al., 2021). Both models are based on the transformer-to-transformer architecture T5, which is a general-purpose closed-domain encoder-decoder language model that can be fine-tuned for specific language tasks (Raffel, Shazeer, Roberts, Lee, et. al., 2020). The Google researchers added "expert" layers to the T5 model, which consist of many transformers that become highly task-specialized through training. To account for this huge increase in parameters and the resulting computing demand in training, they implemented a "switch transformer" that learns to efficiently route batches of tokens to the appropriate expert. The full-sized ST-MoE is a 269 billion parameter model that achieves state-of-the-art performance across a staggering number and diversity of language tasks.

The limited scope of this project does not allow for expansion on Google's ST-MoE model. Instead, this paper will analyze one of the base components of the model, T5, and compare its performance to other similar transformer-to-transformer models, like BERT, for the narrow task of yes/no question answering.

The models will be benchmarked against BoolQ, a dataset of [question, passage, yes/no answer] triplets (Clark, et. al., 2019). The questions were gathered from aggregated queries of the Google search engine. Questions assessed to be of an answerable yes/no format were then assigned to human annotators to identify a corresponding passage and answer from a Wikipedia article. All questions are answerable.

Methods

The code provided in the Jupyter Notebook "BoolQ.ipynb" contains all operations performed in the course of this project. The analysis can be broken down into three main steps: (1) loading the BoolQ dataset and performing exploratory data analysis, (2) fine-tuning two different BERT models on the dataset and evaluating their performance, and (3) evaluating the one-shot performance of two different fine-tuned T5 models.

BoolQ Dataset Exploration

The BoolQ dataset was easily accessible through the TensorFlow Datasets (tfds) library. Exploratory data analysis showed that the dataset consists of 9,427 training examples, and 3270 validation examples. Each dataset consists of a set of key-tensor pairs: a boolean answer, a bytestring passage, a bytestring question, and a bytestring title (not used). Train and validation sets were loaded into separate python dictionaries for processing. About 62.3% of training examples belong to class “Yes”, which is used as the baseline accuracy metric to beat.

BERT Models Implemented

Two pretrained BERT models were fine-tuned and evaluated with the BoolQ data: BERT-base-uncased, and distilBERT-base-cased-distilled-SQuAD.

Both models required preprocessing of the dataset before fit. Question-answer pairs were fed to BERT’s tokenizer, which adds the necessary [CLS] token at the start, [SEP] token between the two strings, and another [SEP] token at the end. The tokenizer was also used to pad or truncate the inputs to a constant 128 token length. The preprocessor then returned the input tokens, attention mask, and input token id types, along with the boolean answers for use in the models.

The models were then constructed by freezing all BERT layers except for the final output layer and any pooling layers. The final BERT layer was then passed through a 0.3 dropout layer and into the binary classification output layer, where the answer was classified as yes/no with a sigmoid nonlinearity. Prior to training, class weights were calculated to prevent training bias due to class imbalance. The models were fine tuned through standard gradient descent via binary cross entropy loss and optimized with Adam.

T5 Models Implemented

Two pretrained T5 models were evaluated on one-shot performance of the binary QA task: T5-base, and T5-small-finetuned-BoolQ by user “mrm8488” on HuggingFace.

Since T5 is an abstractive generative model, it is not immediately suited for multiple choice or binary QA. The prompt for each model had to be constructed in the following format:

“question: {BoolQ[question]} (A) Yes (B) No context: {BoolQ[passage]}”

The input text was then tokenized and padded or truncated to 128 tokens in length. At inference, the max output length was constrained to 2 tokens, with all outputs beginning with a “<pad>” token, and then ideally immediately followed by the word “yes” or “no”. The T5 models produced other answers than just “yes” or “no”, especially T5-base, so post-processing was required before output candidates could be compared against ground truth answers for evaluation. Decisions about how to interpret the outputs for analysis are discussed in the results section.

Results and Discussion

AUTHOR”S NOTE: My Google Colab runtime ran out of compute tokens on the assignment’s due date, so supporting evidence, plots, etc. for this analysis fall short. I know I should have planned further ahead, but it is still so immensely frustrating to have my models slow to a crawl in the final stretch, after having spent so many hours building, troubleshooting, and training them. I strongly urge UC Berkeley to provide students with a Colab Pro subscription if we are expected to use the tool for these projects. At the least, *allow* the linking of a payment method to berkeley.edu google accounts so we can pay for it ourselves.

The accuracies of the four models in this report all fall between the baseline (predict “yes”), and the state-of-the-art (ST_MoE), as expected. I was unable to report a reliable accuracy metric for the T5-base model with my approach, and possible solutions are discussed below.

	Model	**Accuracy (↑)**
	Baseline (predict “yes”)	62.3%
BERT	base-uncased	64.3%
	BERT-base-cased-distilled-SQuAD	65.3%
T5	T5-base	~
	T5-small-finetuned-BoolQ	85%
	State-of-the-art (ST-MoE)	**92.4%**

BERT

Overall, the BERT models as fine-tuned here did not perform particularly well on the task, scoring just barely above the baseline on the best validation accuracy scores. The BERT-SQuAD fine tuned model had the highest accuracy in its first epoch, and saw decreasing performance.The following plots of test and validation accuracy by epoch show that the models are susceptible to overfitting. This could be addressed in future work by performing hyperparameter tuning, and/or by increasing the dataset size.

T5

T5-base produced “no” answers, but did not output any “yes” answers. Here is an example of a generated answer that is neither a “yes” nor a “no”

Ground Truth	No
T5-base output	“descent”
Question	can i get canadian citizenship if my grandfather was canadian?
Passage	There are four ways an individual can acquire Canadian citizenship: by birth on Canadian soil; by descent (being born to a Canadian parent); by grant (naturalization); and by adoption. Among them, only citizenship by birth is granted automatically with limited exceptions, while citizenship by descent or adoption is acquired automatically if the specified conditions

	have been met. Citizenship by grant, on the other hand, must be approved by the Minister of Immigration, Refugees and Citizenship.'

The model correctly identified a very relevant word to the answer of the question; it identified the mode of citizenship most closely related to the question, by ‘descent’. It failed, however, to understand the nuance that distinguishes the “parent” descent requirement, from the “grandfather” context in the question.

An accuracy metric was attempted for this model, by assigning any “no” answer to “no”, and any other output to “yes”. This proved to be a very flawed metric, resulting in an accuracy of ~47%, much lower than the baseline method. Whether the metric was flawed or not, the model was not outputting the correct answer format, so fine-tuning was needed. This is where the T5-small-finetuned-BoolQ model shines.

T5 is designed to be finetuned for these tasks, as evidenced by the resounding success of the second T5 model analyzed. The T5-small-finetuned-BoolQ model boasts an accuracy of 85%, which is just 7% behind the absolute state-of-the-art. For such a small model, it holds up surprisingly well to the best.

Conclusion

The work in this paper compared two basic transformer-based language models BERT and T5 against ST-MoE, the state-of-the-art in question answering, in the task of boolean (yes/no) question answering. BERT only showed a marginal improvement over the dummy baseline model of predicting the most frequent class. The performance of a pre-trained variant improved slightly over the base model. T5 failed the task of one-shot yes/no question answering without fine tuning, but excelled when fine-tuned specifically for the task.

References

- Zoph and Bello, et.al. *ST-MoE: Designing Stable And Transferable Sparse Expert Models*. 2022. https://arxiv.org/pdf/2202.08906v2.pdf
- Fedus and Zoph, et. al. *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. 2021. https://arxiv.org/pdf/2101.03961v3.pdf
- Raffel, Shazeer, Roberts, Lee, et. al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. https://arxiv.org/pdf/1910.10683.pdf
- Clark, et. al. *BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions.* 2019. https://arxiv.org/pdf/1905.10044v1.pdf