BERT Question Answering on Squad 2.0

Sai Raghava Mukund Bhamidipati and **Gagan Gopinath** and **Tarun Anand**
Department of Computer Science
University Of Wisconsin-Madison
bhamidipati3@wisc.edu,gagan.gopinath@wisc.edu,tanand3@wisc.edu

1 Introduction

The ability to answer questions is an important task to establish further progress in machine comprehension of human language. Question answering is a multi disciplinary domain that involves information retrieval and natural language processing. This requires models capable of converting human text into an internal representation, that can generate answers to user questions. Question Answering has a long history dating from the 1970s, but the initial focus was on the development of domain specific models, that utilized a pre-existing database to answer user generated questions. However, research in the past decade has yielded improvements in the field, where general purpose question answering systems can easily be built on given text, using different model architectures.

Question answering can be broken down into the following three categories-

- IR-Based Factoid Question Answering - This involves answering a question by using context obtained from the internet, or from relevant documents.
- Knowledge-Based Question Answering - This involves using context derived from structured databases, which are queried to obtain the relevant, valid answer.
- Using multiple sources - These systems utilize multiple sources to obtain the relevant answer, and a prominent example is IBM's Watson.

We focus on IR-Based Factoid Question Answering by working on reading comprehension tasks, which is used as an important metric to measure natural language understanding. Once a source is obtained, it is used as context and the questions that follow are treated as reading comprehension tasks. While it might appear to be straightforward once the relevant context is obtained, the extremely subjective nature of language, coupled with its varying nuances, makes question answering a non trivial task.

Despite these difficulties, there have been significant improvements in contextual question answering. Models capable of achieving human level of accuracy on the Stanford Question Answering Dataset (SQuAD) (1) have been implemented. The SQuAD dataset is a large, high quality dataset that serves as an important benchmark in question answering tasks. Prior work that uses state of the art language representation models such as BERT (2) have been able to achieve F1 scores that outperform other attempts on the existing leader-boards. However, SQuAD consists of questions which are guaranteed to have an answer in the provided context, which decreases its robustness to distracting sentences. This results in models that end up selecting the span that appears to be most related to the question, rather than checking whether the answer is contained within the text.

To address these shortcomings the SQuAD 2.0 dataset (3) was constructed, compiling the existing answerable questions from SQuAD, along with unanswerable questions on the same paragraphs. The newly added questions are constructed such that they are relevant to the context, and contain a plausible but incorrect answer. Model architectures achieving human level accuracy on SQuAD perform a full 23.2 points lower on SQuAD 2.0 when compared to human performance, proving the complex nature of the dataset. In our work we re-implement the existing BERT implementation on SQuAD 2.0 to serve as a baseline. We then perform analyses on the type of questions present in the dataset and the performance of Bert with respect to different types of questions. We find certain shortcomings due to the nature of the dataset, and we propose methods to mitigate these issues.

2 Literature Survey

Prior work implementing different model architectures on SQuAD and SQuAD 2.0 exists. An important implementation is the original BERT paper (1) which utilizes SQuAD as a benchmark. In this extremely influential paper, the BERT model is introduced as a powerful language representation model, that drastically improves upon the performance of its contemporaries. BERT incorporates a novel pre-training and fine-tuning approach which reduces the time required for learning, as well as reduces the need for heavily-engineered, task specific architectures. Despite its non specific nature, it is capable of outperforming many task-specific architectures on various sentence-level and token-level tasks. BERT involves pre-training on unlabelled data over different tasks, and uses a pre-training corpus consisting of the BooksCorpus (4) and English text from wikipedia. Fine-training on SQuAD 2.0 is performed to measure its performance. BERT finetuning is relatively straightforward, since it is possible to model downstream tasks through the self-attention heads. This is done by representing the input question and passage as a single packed sequence, and using just a start and end vector. Questions that do not have an answer are treated as having an answer span with start and end at the [CLS] token. Sum of log-likelihoods of the correct positions is taken as the training objective. With fine-tuning for 2 epochs, BERT was capable of obtaining results comparable to those at the top of the leaderboards, and achieves a 5.1 F1 improvement over other models that do not use BERT as one of their components.

Another important model architecture is that of DistilBERT (5), which uses a smaller model, that is pre-trained with knowledge distillation. This results in lighter models, that require smaller computational power, but still obtain performances similar to larger models. Knowledge distillation is a compression technique where a student model which is compact is trained to obtain performance similar to a larger teacher model. DistilBERT utilizes the same general architecture as BERT, with a 2x reduction in number of layers. It also takes advantage of the common dimensionality between teacher and student networks, to initialize the student by taking one layer out of two. Finetuning DistilBERT on SQuAD results in performances withing 3 points of the original BERT model, despite a 40% reduction in model size.

Using conventional machine learning techniques can improve the effectiveness of BERT on SQuAD 2.0 as shown by Hulbard et. al (7). Using a context aware convolutional network, initialized with filter generators has a positive effect on BERT's performance. Replacing traditional CNN filters with meta-filters which are applied over the input sequences results in the generation of a filter-feature map, which can recognize the important and relevant tokens from the given context. Using this approach results in BERT predicting the answerable and non answerable SQuAD 2.0 questions reasonably well. Another approach followed by Hulbard et. al involves using recurrent neural networks with LSTM units to maintain an internal state representation of the question's significance. However, this approach results in significant overfitting of the non answerable questions, with a poor performance on answerable questions.

During the course of our analyses, we discovered that the SQuAD dataset faced certain limitations due to the presence of a skewed dataset. One approach we looked at to correct this imbalance was to generate further questions to address the skew. We aimed to automatically generate questions from the SQuAD dataset based on the work of Zhou et. al (6). Zhou et. al introduce a novel framework called the Neural Question Generation (NQG) framework, that can generate natural language questions from a given text, without the need to specify rules for the same. It utilises a sequence-to-sequence model with an enriched encoder that uses lexical features to generate answer focused questions. The word vector is concatenated, along with the lexical feature embedding vectors and an answer position indicator, which serves as the input to a bidirectional Gate Recurrent Unit (BiGRU) encoder. It also deploys an attention-based GRU decoder that decodes sentences and answer information in order to generate new questions. They use the SQuAD dataset as a test set to determine the effectiveness and quality of the questions generated via a mixture of human ratings and ablation tests. When compared to the baseline score, NGQ outperforms it by a factor of 0.76, indicating that the generated questions are quite related to the given sentences and answer spans.

Parameters	Value
Learning Rate	2e-5
Weight Decay	0.01
Epochs	3
Max. Feature Length	384
Dropout	0.1
Activation	GeLU
No. of Layers	12
No. of heads	12

Table 1: BERT configuration

Parameters	Value
Learning Rate	2e-5
Weight Decay	0.01
Epochs	3
Max. Feature Length	384
Dropout	0.1
Activation	GeLU
No. of Layers	6
No. of heads	12

Table 2: DistilBERT configuration

3 Reimplementation

In our experiments we focus primarily on BERT and DistilBERT models, the basic backbone architecture of most NLP models. Since it is computationally inefficient to train these models from scratch, we built them on top of pre-trained models. These pre-trained models were obtained from the Hugging Face open source repository (8). We then followed the following pipeline for cleaning and preprocessing the data -

- Since the dataset contains just the start index for each answer, we tokenize the inputs and format it to include the end index as well
- In typical NLP tasks, long documents are dealt with by truncation. However, that does not appear to be ideal for question answering since removing part of the context could result in losing the answer we are looking for. We deal with large text examples by allowing them to provide several input features, each having a length shorter than the maximum length of the model. Overlap between features is allowed to account for cases where the answer lies at the point of splitting
- We merge the tokens in such a way that the [CLS] token is followed by the question, followed by [SEP], and finally the context. The final prediction will be the first and last token index with reference to the context, which in turn will be the span of our answer

Post the preprocessing and cleaning phase we finetune both the models under the configurations shown in Table 1 and Table 2.

We measure the final model's performance based on the two popular metrics for question answering i.e F1 score and Exact Match (EM). F1 score can be defined as the average overlap between the prediction and ground truth answer. Exact match as the name suggests gives a score of 1 if the predicted answer is exactly the same as ground truth answer, 0 otherwise.

We obtain an F1 score of 74.3% and EM score 72.7% on BERT and an F1 score of 66.1% and EM score 63.2% on DistilBERT. It is important to note that DistilBERT makes a tradeoff between architecture complexity and performance as evident in the scores obtained, but it drastically reduces the resources involved in training.

4 Analysis

While BERT and DistilBERT provided accuracies comparable with the original implementation, we performed further analyses that highlight the following issues-

4.1 Skewed Dataset

Q-Type	Train Data	Dev Data
what	77847	7500
who	13904	1081
how	13528	1281
which	8056	490
when	8009	736
where	5320	488
why	1898	191
others	1757	106

Table 3: Breakdown of the type of questions in the dataset

We classified the questions into different categories on the basis of the type of questions as shown in Table 3, and evaluated the F1 score for each category. Figures 1 and 2 illustrates the results of the category wise, model performance evaluation. A clear distinction can be observed between categories that have greater representation in the training data set, versus those with a lower representation. Section 5 expands upon this observation in greater detail.

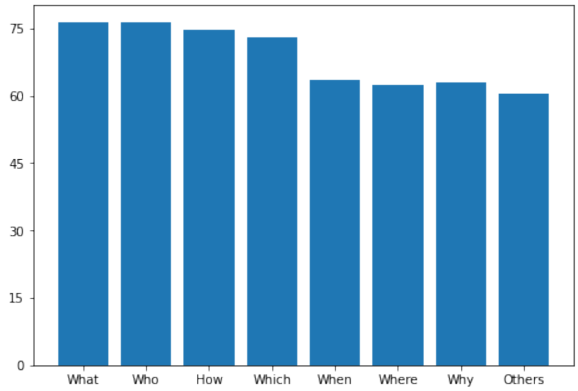

Figure 1: BERT F1 scores for different categories

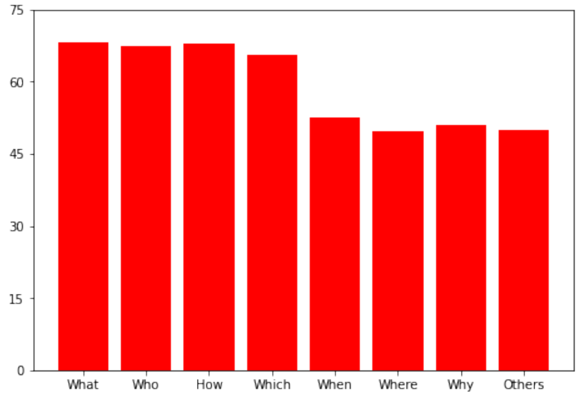

Figure 2: DistilBERT F1 scores for different categories

4.2 Numeric vs Non Numeric Answers

Q-Type	Train Data	Dev Data
Numeric	19008	3648
Non-Numeric	111311	8225

Table 4: Breakdown of numeric vs non numeric answers in the dataset

Table 4 classifies the dataset on the basis of numeric or non numeric answers to the question. Figures 3 and 4 illustrates the performance of the two models on the different types of expected answers. Despite numeric answers forming a mere 14.58% of the training dataset, the performance on numeric answers is much higher than non-numeric. We believe that since its easier to look for numerical answers in a large corpus of text, such questions are easier to answer, resulting in a better performance.

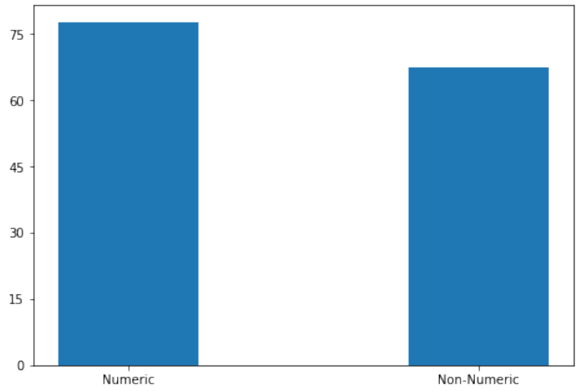

Figure 3: BERT F1 scores for numeric vs non numeric

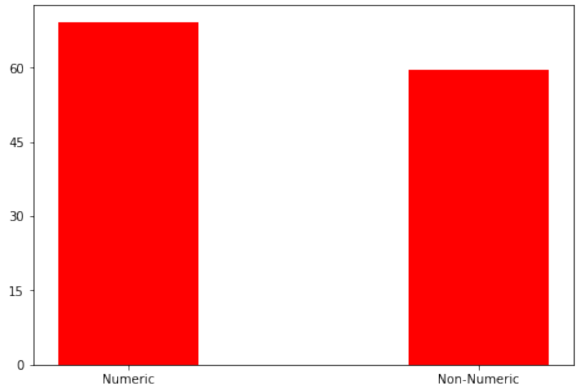

Figure 4: DistilBERT F1 scores for numeric vs non numeric

4.3 Answerable vs Non-Answerable questions

Q-Type	Train Data	Dev Data
Answerable	86821	5928
Non-Answerable	43498	5945

Table 5: Breakdown of answerable vs non-answerable questions in the dataset

An important addition to SQuAD 2.0 is the presence of unanswerable questions, and the breakdown of answerable vs non-answerable questions is present in Table 5. The performance of the two models with respect to these questions is depicted in Figures 5 and 6. We notice that both models underperform on unanswerable questions. This likely occurs due to the models attempting to predict an answer for most questions, without learning any information about the impossible questions. Loss of contextual information due to filtering could result in underfitting of non answerable evaluation data.

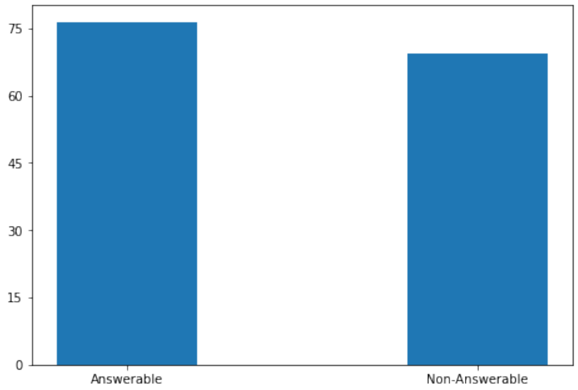

Figure 5: BERT F1 scores for answerable vs nonanswerable

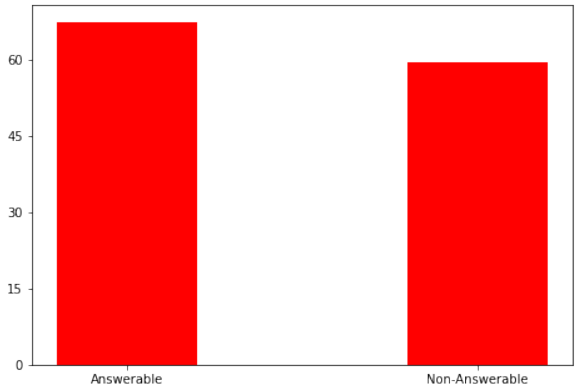

Figure 6: DistilBERT F1 scores for answerable vs nonanswerable

5 Proposed Improvements

From Section 4, it is evident that the models under perform on questions belonging to the 'Why', 'Where', 'When', and 'Misc' ('Do','Does','Is', etc.) categories, when compared to questions from the 'What', 'Who', and 'How' categories. We theorize that this occurs due to the skewed nature of the dataset, which results in an imbalance in the number of questions of each category. We hypothesize that the higher proportion of questions from the latter categories enables the models to learn more effectively, and perform better on similar test samples. However, in a real world scenario, we believe that there would be an equal representation of different categories of questions. Therefore, it is imperative that these models perform well on all types of questions asked of them.

We propose the following ideas to address the aforementioned shortcomings -

- Augmenting the pre existing dataset: This would involve enhancing the existing dataset by augmenting it with samples from other question answering datasets in order to balance the different question categories. Select contexts and questions from similar datasets such as, Natural Questions (NQ) (9), Question Answering in Context (QuAC) (10), Conversational Question Answering (Coca)(11) would achieve this desired result. A uniform dataset would not only improve consistency across categories of questions but also improve robustness as a whole

- Generate new under represented questions: In continuation to the above point, robust natural language modelling techniques can also be used to generate brand new never before seen questions. This could include, for instance, using neural networks such as those mentioned in (6) to generate new questions in each under represented category.

- Using statistical techniques: Using an array of statistical techniques and data augmentations such as Over Sampling, Under Sampling, Stratified Sampling etc. to improve question distributions in each batch/minibatch. This would use the already existing questions efficiently while also increasing the weightage of minority question types

We hope that exploring the above mentioned proposals helps level the playing field in terms of representation of different types of questions which in turn would lead to better model performance overall.

6 Plan of Activities

Table 6 provides an outline of our planned future directions as well as the breakup of tasks amongst the team members.

Task	Deadline	Assigned to
Augmenting existing Dataset	4/25	Tarun
Generate new questions	4/25	Gagan
Statistical techniques	4/25	Mukund
Project Presentation	4/28	All 3 members
Final Report	5/2	All 3 members

Table 6: Schedule to complete the final report

References

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

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

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

[4] Y. Zhu et al., "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 19-27, doi: 10.1109/ICCV.2015.11.

[5] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).

[6] Zhou, Qingyu, et al. "Neural question generation from text: A preliminary study." National CCF Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2017.

[7] Gupta, Suhas. "Exploring Neural Net Augmentation to BERT for Question Answering on SQUAD 2.0." arXiv preprint arXiv:1908.01767 (2019).

[8] Hugging Face - Home of Machine Learning, https://huggingface.co/, accessed on 4/8/2022

[9] Kwiatkowski, Tom, et al. "Natural questions: a benchmark for question answering research." Transactions of the Association for Computational Linguistics 7 (2019): 453-466.

[10] Choi, Eunsol, et al. "QuAC: Question answering in context." arXiv preprint arXiv:1808.07036 (2018).

[11] Reddy, Siva, Danqi Chen, and Christopher D. Manning. "Coqa: A conversational question answering challenge." Transactions of the Association for Computational Linguistics 7 (2019): 249-266.