Ensemble-Learning for Sustainable NLP

Stanford CS224N Custom Project

Elena Berman
Department of Computer Science
Stanford University
eaberman@stanford.edu

Surya Narayanan Hari
Department of Management Science
& Engineering
Stanford University
surya21@stanford.edu

Abstract

Large NLP models result in higher accuracy scores but are more resource intensive on tasks such as QA, incurring a significant environmental cost [1].The question we are addressing is the "Why a task such as QA requires such resource intensive deployment". In order to minimize inference costs while maximizing accuracy at inference time, we develop classifier models that can inform whether we could invoke a small model instead of a big one to answer questions. We found that a small QA model has the potential to answer questions with accuracy comparable to a BERT-based model in approximately 4 in every 5 questions. We find that a rule based ensemble can improve the F1 by over 25% and save 58% of the resources used. We also find that a neural ensemble that is used to predict whether a small model can answer a question correctly covers 50% of the gap between a small model and a big model. Our research recommends the development of NLP classifiers to find more energy efficient deep learning implementations.

1 Key Information to include

- Mentor: Matthew Lamm
- We have no external collaborators and are not sharing this project.
- We choose grading Option 2.

2 Introduction

Increasingly large and complex architectures have dominated the field of NLP. Such architectures as BERT have immense computational and environmental costs. [1]. Our goal is to reduce these costs.

BERT models are effective but computationally expensive, leading to a great deal of work to optimize their computational performance. Most existing benchmarks for performance measure efficiency by time to process one minibatch of data [2]. Existing optimizations to these large algorithms include accumulating updates to the gradient. Other measures, check the number of FLOPS required for an algorithm to converge by optimizing hyperparameter search. [3].

Our goal is to create a model which is comparable in accuracy to the performance of existing models but be more resource aware. Previous BiDAF-based models have been able to achieve exact match (EM) and F1 scores of 68.53 and 71.46 on SQuAD 2.0 datasets [4]. A BERT model we used, has an F1 of 60. Our resource efficient Our models were able to obtain a 75 F1 whilst conserving 41.3 % of the resources used in deploying a BERT only model [1]

[1]We used a BiDAF model whenever a BiDAF model produced an F1 score of 1, which happened 58% of the time. We deployed a BERT model for the remaining 42% of the questions and computed resources saved by hours that hour machine was turned on during deployment (10 minutes for BiDAF and an hour for BERT), 1 - (10 + .42 * 60)/60 = .413 . Unfortunately, we have to deploy a BiDAF model 100% of the time to find where it

3 Related Work

BERT was developed two years ago as the state-of-the-art [5]. Since then, more details about the environmental consequences of BERT-based models have come to light [1]. Most recently, DistilBERT was developed to be a "lighter" version of BERT, which reduced the size of BERT by 40% [6]. However, while DistilBERT is lighter than BERT, it is still of a comparable cost when compared to smaller models such as BiDAF [7] (DistilBERT still has 30x params that a BiDAF model has). Recently, Green AI [3] has made a foray into the world of NLP, from vision and seeks to make the distinction between resource heavy 'Red AI' and resource light 'Green' AI.

4 Approach

Our goal is to identify situations in which using a BERT-based model can be avoided. We take on a question-answering task from SQuAD v2.0 and explore questions which can be accurately answered by a small model. To serve this end, We employ ensemble learning, hypothesizing that some questions can be answered by a small model, while others can only be answered by a larger, BERT-based model.

In order to test this, we trained a BiDAF model as our small model and a BERT-based model as our large model. We created rule-based and neural architectures that predict accuracy and confidence of the smaller model to answer a given question.

Comparing Models. We trained a Bidirectional Attention Flow (BiDAF)-based model as our "small" model. At the time it was released, BiDAF based models [7] topped the leaderboard on SQuAD 1.1 It has 50x fewer parameters than a BERT-base model. To establish a base for a large model, we fine-tuned a few varieties of BERT-base released by Huggingface

Rule-based approach Our initial analysis consisted of data science from the output of running a BiDAF model. We were especially interested in how often a BiDAF model output results that were comparable with that of a BERT model. We found that on 58% of our test set, a BiDAF model achieved a perfect F1 score.

As a rule based approach, we could automatically triage all the scores that a BiDAF model achieves a perfect F1 score of to a BiDAF model and only triage the remainder to a larger BERT model, reducing the number of invokes by 30%. This presented us with a baseline.

Data Analysis: Analyzing features in questions. To improve upon our performance, we analyzed differences in the questions that a BiDAF model got right and those that it got wrong. We examined the distribution of features of the data such as what question word was used, how long the sentence was and the length of the overlap between the question and the passage as a way of discerning the distribution over the data. We found no discernable results for the BiDAF model but could discern a distribution over what a BERT model got correct more often. Our results are reported in Section 6, Analysis.

Data Analysis: Studying the confidence of a BiDAF model. One way of informing the decision to triage a question to a BiDAF model would be to determine whether the model would have a high confidence on its prediction. We performed data analysis on the test set predictions to understand how confident our model is and how often it were better than BERT.

Hence, we used the equation

$$\frac{|\text{ Number Confident and Better than BERT }|}{|\text{Total Number}|}$$

We found this value as a probability and used it for various thresholds of confidence. If the model were highly confident, it would make more sense for us to trust its predictions. Our results are presented in Table 1. If we analyzed one of the rows of this table, it would become immediately apparent that we could find a direct use for examples whose predictions from a BiDAF model achieve a best prediction confidence of 90% and higher, since they are better than BERT 13.7% of the time. The question remains as to whether among all questions that receive a best prediction rating of 90%

produces an F1 score of 1. In an alternate approach, we use a neural net to predict where the BiDAF model will have a score of 1 to avoid having to deploy the BiDAF model needlessly

Confidence at Least	Fraction	Fraction better than BERT
0.9	25.2 %	13.7%
0.8	35.5 %	20.8%
0.7	44.7 %	27.0%
0.6	54.4 %	33.7%

Table 1: Confidence of prediction and fraction of times better than BERT

of higher, which of them will be better than BERT. Currently, questions boasting this confidence have a $\frac{13.7\%}{25.5\%} \approx 50\%$ chance of being better than BERT.

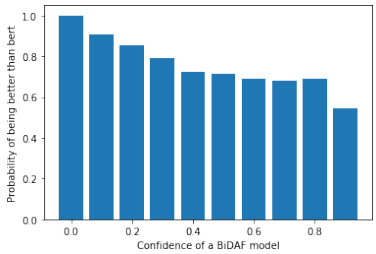

Figure 1: Probability of a small model being better than BERT

An interesting challenge posed to us was that the more confident our BiDAF model was, the less likely it was better than BERT. Our results are summarized in Figure 1.

Neural Architecture.

To conserve energy as much as possible, we trained a small neural network consisting of an LSTM with 8 hidden units following an embedding layer and followed by a linear layer. The model architecture is shown in Figure 2. We trained model variations with this underlying architecture to perform several tasks, described below. We use mean-squared error and accuracy metrics to determine the success of our models.

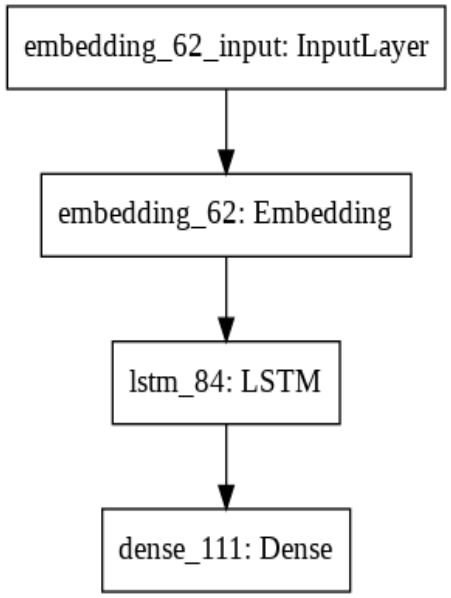

Figure 2: Small classifier model

Class name	F1 at least as good as BERT	F1 worse than BERT
More than 90% confident	Class 1	Class 2
Less than 90% confident,	Class 3	Class 4

Table 2: Division of Dataset into Four Parts for Multi-Class prediction

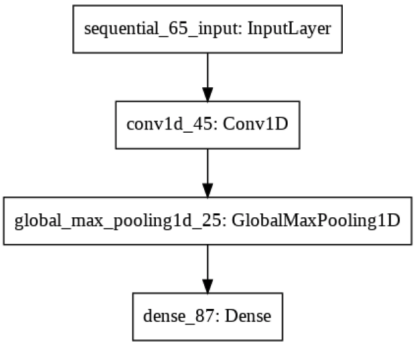

Figure 3: Neural architecture for model V

Model I: Predicting whether we should use the Small model. If the F1 score for a given question is equal to or higher than a BERT model, then we should use the Small model. We trained on this task.

Model II: Predicting the confidence of a BiDAF model. A hypothesis we have is that if we can predict the confidence of a BiDAF model, then we can deploy it with some confidence. Our results are discussed in the results section.

Model III: Predicting whether a small model will have greater than 90% accuracy and whether it will outperform BERT. We split our training data into four classes, as described in table 2. We ran our Neural model to perform a multi-class classification task. Our results are summarized in the results section

Model IV: Predicting the F1 score of a small model. A fourth neural model we ran was to check if we could predict the F1 score of a small model. We used two models for this task, one that predicts an F1 as 0/1 and the other that predicts the F1 as a real-valued number.

Model V: Predicting the F1 score of a small model. We improved our model to include a Conv1D to capture spatial information. Our architecture is shown in Figure 3.

5 Experiments

5.1 Data

We trained all our models on SQuAD 1.1, fine-tuned our big models on SQuAD 2.0, and evaluated the latter on the SQuAD 2.0 dev set. The models performed the SQuAD v2.0 QA-task of answer-prediction by generating a span in the context paragraph to identify the correct answer if an answer exists, and reporting that there is no answer if not.

5.2 Evaluation method

We used mean-squared error (MSE) in our neural models. We measured accuracy based on our MSE. Additionally, we tracked the train and test error over the course of the epochs, to observe whether our model was overfitting.

Model	F1	Energy Used (Minutes to deploy one Eval Set)
BERT	60.9	60
BiDAF	57.9	10
Rule Based Ensemble	**75.5**	**35.2**
Deep Learning Ensemble	59.8	39.0

Table 3: Summary of results

5.3 Experimental details

We trained our neural models for 100 epochs on results obtained by our models on approximately 5950 SQuAD v2.0 questions. We used a 80/20 train/test split. The exact inputs and outputs are described in our Model definitions above.

5.4 Results

Directly comparing small and large models. We ran our BiDAF model and a BERT-based model on a set of $5,952$ SQuAD v2 questions, using the same questions for both models. We wrote code that analyzed model f1 scores and designated whether a question should be answered by the BiDAF model (if the f1 score was greater than or equal to BERT-based model) or the BERT-based model (otherwise). Results are shown in Figure 4. Of the $5,952$ questions, 4920 could be answered by BiDAF, and 1031 by the BERT-based model. This is **82%.** F1 scores were equal across models (and thus assigned to BiDAF) in 3831 cases.

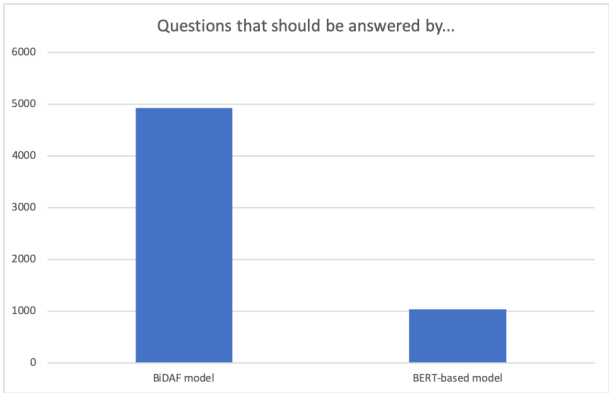

Figure 4: Model that Could Answer Questions in Test set

Neural Model I: Small model to predict whether we should use a small model to classify a particular question As mentioned in our approach section, we fit a small classifier and overfit our training set (as shown in Figure 9). Our prediction accuracy was still relatively low, and the model always predicted the overrepresented class label, despite adding class weights.

Neural model II: Predicting the confidence of a BiDAF model

Our training curves are reported in Figure 10. Again, we overfit the training set but have relatively low accuracy.

As an addendum result, we also tried to narrow the scope of predicting accuracy, to see if our model would be any more precise in predicting the confidence of a small model if presented with the binary task of determining whether the small model would have a confidence of 0.9 or greater. Again, our model seems to have ascertained no discernable way of determining whether the small model would have such a confidence above a certain threshold. Our training curve from this experiment are shown in Figure 11.

Neural model III: Predicting the confidence and usability of a small model. We overfit the training set with a model to predict whether a small model will be more than 90% confident of

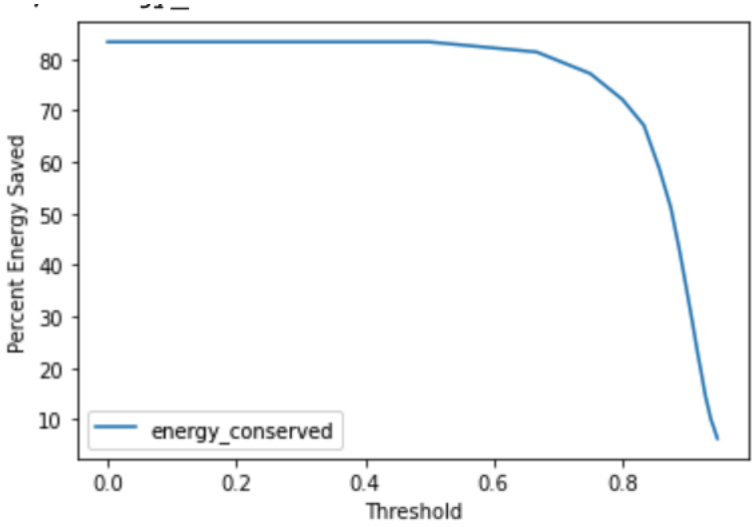

Figure 5: Energy conserved when questions predicted F1 score higher than threshold are deployed to the small model at inference time

answering a question and whether the small model will do at least as well as BERT. Despite overfitting as shown in Figure 12, our model was only as good as flipping a coin.

Neural Model IV: Predicting the F1 Score of a small BiDAF model. On a test where we are predicting the F1 score of a BiDAF model as a 0/1 classification task, our model achieves its best results by predicting a 1 for every answer. We fix this by adding class weights and find that the neural architecture still encounters a high bias problem. Our test accuracy was 0.42 when we try to predict the F1 as a real number and .5 when we try to predict the F1 as 1/0.

Neural Model V: We improved the architecture by adding a Conv1D. We got significantly better results and were able to predict over 89% of the F1 scores of the small model when the small model got F1 scores of 0.9 or over. We also found that the F1 of the neural model on the test set was one point better than the small model when predictions above a certain threshold were used. The threshold we set was 0.95, meaning that whenever the model predicted that the small model would receive an F1 score of 0.95 or above on this Question, the question was triaged to the small model.

Our results from the neural model where we investigated how much energy could be saved against some threshold of F1 above which we accepted all results that a small model produced are shown in Figure 5

6 Analysis

Question Features of Success and Failures in Small Model. We engineered question-based features to analyze which features that may contribute to a question's accuracy or inaccuracy. Then, we analyzed these features across models. We defined the following relevant features: 1) question words (as defined by first word in question); 2) question length; and 3) number of words present in both question and context passage; and, 4) longest common subsequence of words between question and context passage.

The distribution of question words across correct and incorrect questions is shown in Figure 6. Correct and incorrect are defined as questions having F1s equal to 1 and 0 respectively, for simplicity. There were more questions incorrect in the "OTHER" category, indicating that BiDAF may have a more difficult time with non-straightforward questions, but more questions should be studied to be conclusive. (A similar question-spread with BERT-based model is available in our Appendix, Figure 13.)

In order to test whether these features could help us differentiate between questions that our small model answered correctly and incorrectly, we implemented a Random Forest Classifier, because it is

a classifier that generally performs well on tasks by introducing randomness to a more traditional decision tree model. We trained the classifier with 100 estimators, taking the question concatenated with features we engineered as input, and predicting "correct"/"incorrect" as output. Our accuracy was only 0.512 on our test set, leading us to believe that the features we engineered were not as useful as we had hoped. When we analyzed the weights learned by the model, we found that the question word was not particularly important compared to the longest-common subsequence (lcs) and number of overlapping words (num_overlap), as shown in Table 4.

Selected Feature name	what	how	was	OTHER	q_len	num_overlap	lcs
Weight learned	0.018	0.014	0.013	0.006	0.0170	0.454	0.408

Table 4: Weights Learned in Feature-based Classifier

Qualitative Analysis. To understand why this could potentially be the case, we looked at questions. It seemed like our model tended to get questions that would be highly likely to have overlapping subsequences with contexts correct, such as "What encoding decision needs to be made in order to determine an inaccurate definition of the formal language?" or "Nobody has generalized the meaning of the word imperialism down to general-purpose what?" This is confirmed since it got incorrect questions such as "What's the party's take on Muslim history?" and "When did violence end in war?" However, although this generally seems to be the case, we cannot make the full connection to this because we also found some exceptions to this rule; for example, our model got incorrect the question: "Secular Arab nationalism was blamed for both the success of Arab troops as well as what type of stagnation?" which appear on the surface to be closely connected to the context passage.

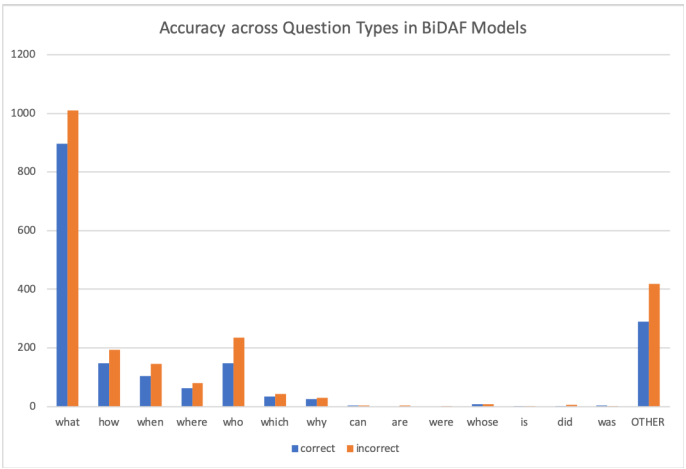

Figure 6: Distribution of Question Words in Correct and Incorrect Questions

Analyzing Output of Neural Model Predictions. In Neural Model IV, we analyzed the distributions of model predictions based on the F1 scores. For example, for all the questions that have an actual F1 score of 1, what will our neural model predict the F1 score to be? This question is answered in Figure 7. We noticed that the vast majority of predicted F1-scores were 0 or 1. While we were initially surprised to notice that this was the case for questions with actual F1 scores of 1, meaning the model would sometimes predict 0, we also noticed that the number of intermediate values between 0 and 1 in our dataset were fewer, so it matched our dataset. This trend is matched when we plotted the numerical difference between observed and predicted F1 scores (Figure 8). An important avenue of future work would be to use more data in order to expose our models to more examples of questions with varied F1 scores.

7 Conclusion

We found that a small model has the potential to be used in approximately 80% of questions in the SQuAD v2.0 question-answering task as compared to a big BERT-based model. However, we found that determining which questions could be answered by a small model was a more difficult task. We

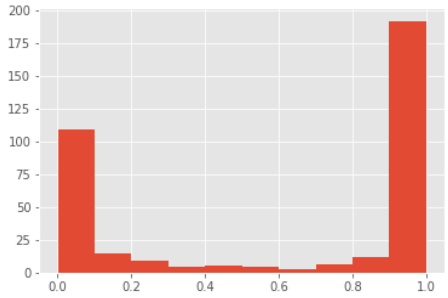

Figure 7: Neural Model-Predicted F1 Scores Of Questions With Actual BiDAF F1 Score of 1

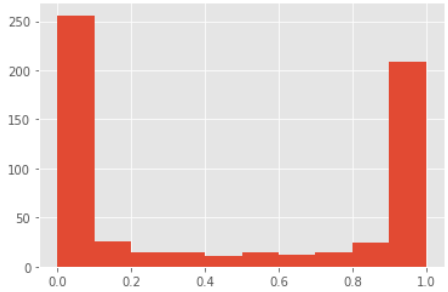

Figure 8: Difference in Neural Model-Predicted F1 Scores And Actual BiDAF F1 Scores

analyzed the types of questions the models got correct and incorrect using a rule-based classifier model and feature-engineering. This analysis did not produce clear results in a binary classification algorithm, but suggested the importance of some features over others.

Additionally, we implemented neural classifiers to determine various aspects of whether we should use a small model to answer questions. We found that our neural model always overfit. We limited our question lengths to about 16 words out of interest of energy savings, but faced the trade-off of finding sufficient features to bolster our search. This highlights the limitations of NLP.

Ultimately, we found that rule-based and neural ensembles show promise solutions in QA accuracy and potential to save energy, since we found that a rule based ensemble can save energy by not running BERT on about 58% of questions, and a neural ensemble can achieve a high F1 score, and save more than 30% of the energy at inference time with a very high bar for F1.

In the future, we believe that testing models and training neural classifiers on more data could be promising and address many of the problems we encountered. Once a classifier is trained, it has the potential to prevent immense computational and environmental costs.

References

[1] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. *CoRR*, abs/1906.02243, 2019.

[2] Cody Coleman, Daniel Kang, Deepak Narayanan, Luigi Nardi, Tian Zhao, Jian Zhang, Peter Bailis, Kunle Olukotun, Chris Re, and Matei Zaharia. Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark, 2018.

[3] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai, 2019.

[4] Haoshen Hong, Kuangcong Liu, and Xiao Wang. Accelerated and accurate question answering. 2019.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. 2019.

[7] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

A Appendix.

Our Appendix makes available figures not in our main paper.

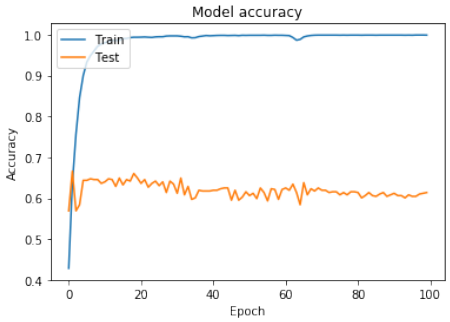

Figure 9: Train and Test (Validation) Acccuracy for small classifier model to predict whether a given question should use a BiDAF model

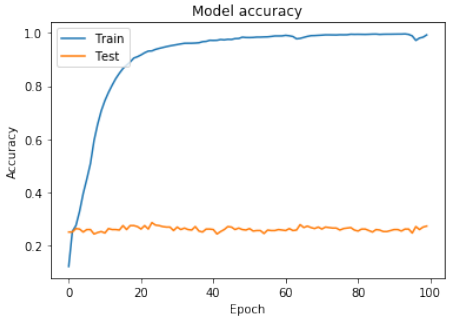

Figure 10: Training curves of predicting the confidence of a small BiDAF model

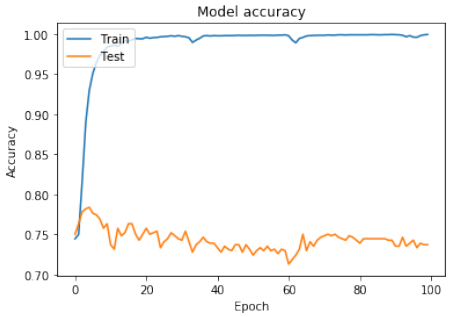

Figure 11: Training curves to predict whether a question will receive more than 90% confidence

A.1 Back of envelope calculations for how much energy it takes at evaluation time to deploy BERT

The GPU (M-60) has a max power consumption of 300W. Assuming an average case consumption of 150W, evaluation takes $150W \times 2$ hours$/12000$ examples $= 0.025Wh/example =$

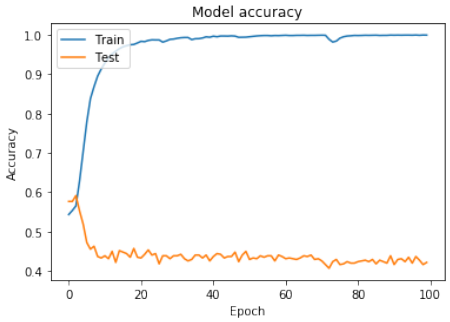

Figure 12: Training curves from running a model to predict the confidence and usability of a multi-class model

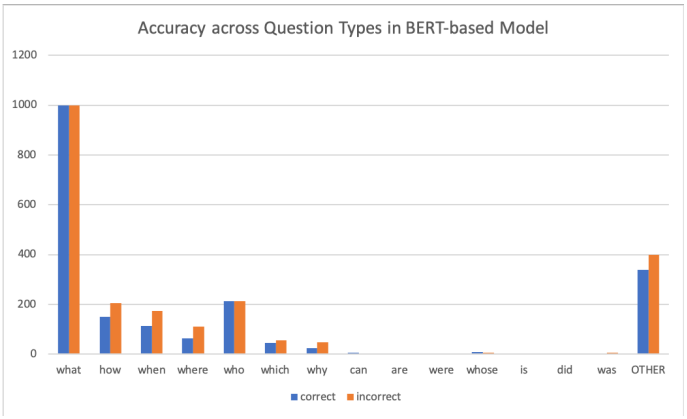

Figure 13: Distribution of Question Words in Correct and Incorrect Questions in BERT-based Model

$2.5 \times 10^{-5} KWh$. Assuming Google gets 4 billion search hits in a day, (`https://www.internetlivestats.com/google-search-statistics/#rate`) then to evaluate a BERT model on every search query would be $2.5 \times 10^{-5} kWh \times 4$ billion $= 1 \times 10^{5} kWh = 100 MWh$. If our ensemble model saves 58% of that energy, then only 42% of the 4 billion queries will be evaluated on a BERT-based model. This means we would save approximately 58 MWh of energy (assuming energy used to run queries on BiDAF model are negligible in comparison).