Robust QA using Mixture of Experts

Stanford CS224N Default Project

Raj Prateek Kosaraju
Department of Computer Science
Stanford University
rprateek@stanford.edu

Abstract

QA systems have become prevalent in everyday life, but the challenge of training effective QA systems to answer questions from out-of-training-domain still exists today. In this work, we explore a Mixture-of-Experts based approach to train a Robust QA system to for out-of-domain QA. We show that a Mixture of Experts based system can indeed offer competitive performance increase (+2.7 validation EM score and +3.71 validation F1 score) over a baseline model on RobustQA tasks where we have little to no knowledge and training over the evaluated datasets. Based on the results, we find that simply having a MoE system recovers most of the gains we were able to achieve, even without having a trained MLP system or hyperparam-tuning.

1 Key Information to include

- Mentor: N/A
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Over the last few years, we have seen tremendous progress on fundamental natural language understanding problems. At the same time, there is increasing evidence that models learn superficial correlations that fail to generalize beyond the training distribution [1, 2, 3, 4]. From a practical perspective, robustness to out-of-distribution data is critical for building accurate NLP systems in the real world since train and test data often come from distinct user interactions. In this paper, we introduce a question answering system that can adapt to unseen domains with only a few training samples from the domain. Using an Mixture of Experts technique, we find that building a QA system consisting of several DistilBert models combined with an MLP model are able to achieve +2.7 in validation EM score and +3.71 in validation F1 score compared to simply using a single DistilBERT model when evaluated on the three out of domain datasets.

3 Related Work

DistilBERT [5] is a smaller, distilled version of the original BERT model. The authors of DistilBert leverage knowledge distillation during the pre-training phase and show that it is possible to reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. The smaller, faster and lighter model is cheaper to pre-train and has been demonstrated to be especially useful for on-device computations.

Mixture of Experts [6] is a technique where multiple expert learners are used to divide a problem into regions with each experts tasked to excel at one of the regions. The authors present a new supervised

learning procedure for systems composed of many separate networks, each of which learns to handle a subset of the complete set of training cases. The new procedure can be viewed either as a modular version of a multilayer supervised network, or as an associative version of competitive learning. In the original work [1], the authors demonstrate that the learning procedure divides up a vowel discrimination task into appropriate sub-tasks, each of which can be solved by a very simple expert network.

sectionApproach In this section, I describe the detailed approach being taken in this project to generalize a QA system that can generalize to unseen domains.

3.1 Overall approach

3.2 Baseline

The baseline model fine-tunes the standard pre-trained DistilBERT available through the huggingface library. We use the loss function of cross-entropy loss for start and end locations. We average across the batch and use AdamW optimize to minimize the loss.

3.3 Mixture of Experts

Mixture of Experts [6] is a technique where multiple expert learners are used to divide a problem into regions with each experts tasked to excel at one of the regions. Specifically, in my approach, we have a separate DistilBERT model (experts) over each of the datasets available at training-time as well as an additiona model trained over all the in-domain datasets. Additionally, we also have a small MLP model that can act as a gating function. The MLP model is tasked with deciding which expert's results to use for any given Question.

We train k+1 (k=number of in-domain datasets) different DistilBERT experts, each on a specifci in-domain dataset and one with all the datasets combined. These models have a span classification head that compute span start logits and span end logits. We first attempt to evaluate the impact of giving equal weight to each of the k+1 experts, and then train then train a simple MLP, pre-trained on the k datasets and fine-tuned on the out-of-domain few examples, that will weigh the span start and span end logits from each model. Using the ML, we take a weighted average to produce the final span start and span end logits that would produce the result.

3.4 Hyper-parameter tuning

Hyper-parameter is a well-known approach throughout Machine Learning used to tune the hyper-parameters such as learning rate, weight decay, and batch size to obtain the ideal combination that gives us the best performance on validation set.

Using the dev/validation set, the finetuning setup will attempt to tune the learning rate and batch size used for the models' training using a simple grid search where we try each combination of values. For learning rate, we use values 2e-5, 3 e-5, and 4e-5. For batch size, we use values between 8, 12 and 16. Additionally, all of the DistilBERT models used in this approach are the standard pre-trained DistilBERT models available through the huggingface library. The MLP model is pre-trained using the in-domain examples.

4 Experiments

The concrete experiment setup is detailed below.

4.1 Data

The data used to train the k=3 DistilBERT models is the SQuAD [7], NewsQA [8], and Natural Questions [9] datasets respectively, each with 50,000 training examples. The (k+1) i.e. 4th DistilBERT model is trained over all of the datasets together. The MLP model is pre-trained over the SQuAD, NewsQA, Natural Questions, and then fine-tuned over 127 examples each from DuoRC [10], RACE [11], and RelationalExtraction [12].

In the dev/validation set, we have 10.5k examples for SQuAD, 4.2k examples for NewsQA, 12.8k examples for Natural Questions, and 126 examples each for DuoRC, RACE, and RelationExtraction.

In the test set, we have 1248 , 419, and 2693 examples each from DuoRC, RACE, and RelationExtraction respectively. Hence, these are considered as the out-of-domain datasets.

4.2 Evaluation method

We compute the EM score and F1 score over the validation/dev set and test set to compare the performance of our system against a baseline system.

- **EM score**: is a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly.
- **F1 score**: The harmonic mean of precision and recall. It is naturally less strict than EM score.

The EM and F1 scores are averaged across the whole evaluation data to get the final score on validation set and test set scores.

4.3 Experiment Details

The experiments can be categorized as 4 sub-experiments: training the baseline, training the MoE system with equal weights for each expert model (without an MLP model), training the MoE system with MLP model, and finally tuning the hyper-parameters

In the interest of time, we train all models for 2 epochs.

In order to perform hyper-parameter tuning, we perform a grid search for only 1 epoch for each of the learning rates and batch sizes specified earlier for a single DistilBERT model in order to find the best learning rate and batch size combination. We then use this learning rate and batch size value to re-train the whole QA system.

The code repository [13] will be publicly available for easier access to the training setup.

4.4 Results

The results for the first two experiments are detailed below.

QA System type	Train time	Val EM score	Val F1 score	Test F1 score	Test EM score
Baseline system	2h30m	31.85	46.94	Untested	Untested
MoE system w/ equal weights	10h30m	33.25	49.60	58.9	40.8
MoE system w/ trained MLP	11h45m	34.35	50.65	Untested	Untested
+ param tuning	12h15m	34.55	50.95	59.076	40.940

Based on the validation-set EM and F1 score, going from the Baseline single model DistilBERT system to MoE system w/ trained MLP provides an interesting +2.7 EM score and +3.71 F1 score. Adding parameter tuning enhances the results a little as the parameter tuning suggests that the a learning rate of 2e-5 is more optimal that the pre-existing value. It's interesting to note that most of the gain in the validation metrics comes from simply going to the MoE system, even without a trained MLP.

Based on the test-set EM and F1 score, while we don't have the test-set metrics for all of the experiments (due to limitations on how many times we can submit results), the results look consistent with changes in the validation-set metrics. Going from a MoE system w/ equal weights to MoE system w/ trained MLP and parameter tuning, we see a small increase in test F1 score and test EM score, consistend with validation-set metric increases.

It is important to note that since the MoE system trains 4 different DistilBERT models, the train-time increases by at least 4x to 10h30m which is rather large. However, since this is a one time increase during training time that does not impact evaluation (each model is evaluated in parallel), this may be acceptable in situations where better performance at test-time is important at the expense of train-time time and resources.

The trend of the results is consistent with expectations and the results themselves are also in-line with expectations. The MoE system, while expensive from a training-time point of view, is an extremely simple setup and yet it is still able to improve results over the baseline model.

5 Analysis

We perform a brief analysis of validation set results from the best performing system (MoE w/ trained MLP and param tuning) in this section.

- **Difficulty in maintaining context**: The final QA system performs poorly on questions that need the system to maintain context as it parses through the sentence. As a simple example (as the real data examples are much longer), if the text started with "While Raj was studying, he found that the X happened due Y and Z. He also found it did not happen due to A", and the question was "How did Raj know X did not not due to A?", the system may respond with "he found X happened due to Y and Z", instead of maintaining and returning the initial context that he was studying.
- **Answering with larger-than-needed chunks of sentences**: The system often answers with large chunks of text which would hurt the EM score but not the F1 score. For example, if the text was "Machester united is a club founded in the year X in England" and the question was "Where was Machester united founded", the system often responds with the likes of "in the year X in England".
- **Good at questions that simple who/when questions**: Based on analysis of various examples, the system seems to perform quite well at questions that ask for a person's name or a year that an event happened.

The patterns noticed above apply to both the baseline model and the MoE system with and without the trained MLP model, although the issues noted above are slightly less prevalent with the MoE system.

6 Conclusion

QA systems have become prevalent in everyday life, but the challenge of training effective QA systems to answer questions from out-of-training-domain still exists today. In this work, we show that a Mixture of Experts based system can indeed offer competitive performance increase over a baseline model on RobustQA tasks where we have little to no knowledge and training over the evaluated datasets. Based on the results, we find that simply having a MoE system offers most of the gains we were able to achieve, even without a trained MLP system or hyperparam tuning.

7 Limitations and Future work

While this performance increases shown in this work are promising, the primary limitation of this work is that this approach takes 4x the amount of time to train the system compared to the baseline model. This may be acceptable in certain situations, but directions of future research should include how this 4x increase can be further optimized. For example, mixed precision training is an interesting avenue to explore. While it may decrease the performance of an individual model along with reduction in train-time, the MoE system involves multiple models which can compensate for each other and mixed precision training may be fruitful here.

References

[1] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. arXiv preprint arXiv:1707.07328, 2017.

[2] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In Association for Computational Linguistics (ACL), pages 107–112, 2018.

[3] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Association for Computational Linguistics (ACL), 2019.

[4] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4902–4912, Online, July 2020. Association for Computational Linguistics.

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

[6] R. Jacobs, Michael I. Jordan, S. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. Neural Computation, 3:79–87, 1991.

[7] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. CoRR, abs/1606.05250, 2016.

[8] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. ACL 2017, page 191, 2017.

[9] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. In Association for Computational Linguistics (ACL), 2019.

[10] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In ACL, 2018.

[11] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale reading comprehension dataset from examinations. In EMNLP, 2017.

[12] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. arXiv preprint arXiv:1706.04115, 2017.

[13] https://github.com/rajprateek/nlp_squad