Exploring Mixture of Experts, In-context Learning and Data Augmentation for Building Robust QA

Chen Chen Hanzhao Lin Shan Lu

inquisit@stanford.edu hanzhao@stanford.edu shanlu33@stanford.edu

Introduction

Question and answering (QA) systems are commonly used to test the degree of learning and understanding exhibited by language models. In particular, the transfer learning scenario for QA tasks, where a language model is trained on a set of resource-rich datasets, and fine-tuned on resource-poor datasets, is challenging to achieve good results.

In this work, we implemented and benchmarked multiple techniques, namely mixture of experts, data augmentation, in-context learning and hyperparameter tuning, towards building a QA system with better robustness.

Datasets

Dataset	Question	Context	Train	Dev	Test
In-domain Datasets					
SQuAD	Crowdsourced	Wikipedia	50,000	10,507	-
NewsQA	Crowdsourced	News articles	50,000	4,212	-
Natural Questions	Search logs	Wikipedia	50,000	12,836	-
Out-of-domain Datasets					
DuoRC	Crowdsourced	Movie reviews	127	126	1248
RACE	Teachers	Examinations	127	128	419
RelationExtraction	Synthetic	Wikipedia	127	128	2693

Table 1. Data Sources and Splits

System Architecture

Our system is built with 3 building blocks, data augmentation, in-context learning, and a DistilBERT-based mixture-of-experts gating network. This setup allows us to conduct experiments on flexible composition of individual components.

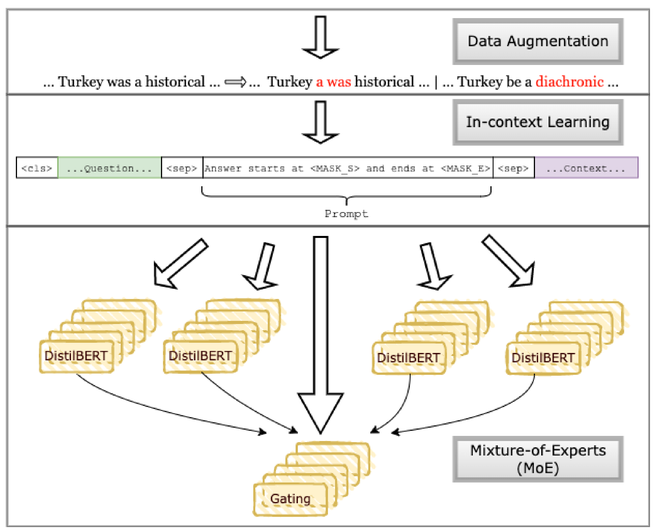

Figure 1. High level overview of system architecture

Approaches

Mixture of Experts

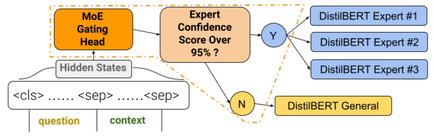

Figure 2. Mixture of Experts Classifier

In the mixture-of-experts network, multiple DistilBERT QA model instances are trained corresponding to every individual out-of-domain dataset. Additionally, a top-level gating network is trained for classifying the input source and forwarding the input to the potential domain experts.

In our implementation, the final output of our mixture-of-experts network is based on exactly one domain expert model or the generalist model. The expert model will determine the final output when high classification confidence earned, with the generalist model as fallback. Formally,

$$y = \begin{cases} f_i & \text{if } g_i \geq 0.95 \\ f_{generalist} & \text{otherwise} \end{cases}$$

where f_i is the output of input x evaluated on expert i's DistilBERT QA model, $f_{generalist}$ is the output from the generalist model. g_i is the classification confidence score produced by the classifier regarding expert i and input x.

Data Augmentation

We hook up our system with a few selected data augmentation techniques provided in `nlpaug`. The augmentation will only be performed in context text with the answer phrases fully preserved.

- **Back Translation**: Translate the context text from English to an intermediate language (Russian or German) and then back to English using Facebook WMT models.
- **Random Swap**: Randomly swap a word with its siblings in the same sentence.
- **Replacing with Synonyms**: Replace some words with their synonyms.

In-context Learning

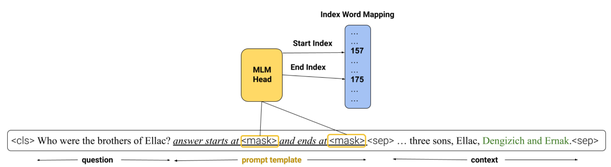

Figure 3. Prompt-based In-context Learning with MLM Head

Instead of classification head in the default model, our approach uses an **MLM** head as proposed in LM-BFF. Moreover, in order for certain generated words to convey unambiguous meanings of labels, a **Label Word Mapping** between MLM generated words and the true label is required.

The choice of framing QA as a language modeling task allows us to use the standard encoder-decoder objective that maximizes the log likelihood of the text in the ground truth target from the output of the model, specifically at the masked indices. Formally,

$$L(\theta) = -log\mathbb{P}(y_{m_s}|x;\theta) - log\mathbb{P}(y_{m_e}|x;\theta)$$

Approaches (cont'd)

Hyperparameter Tuning

- **Reduced answer length**: After inspecting model outputs and analyzing the given datasets, we reduced the max length of model predictions from 30 (default) to 9 to prevent model from producing overdetailed predictions. This trick significantly boosted both EM and F1 scores.
- **Number of frozen DistilBERT layers when fine-tuning**: We freeze embedding layers and first 4 layers of transformers block in DistilBERT when fine-tuning on out-of-domain datasets, since they contain lower-level language features which shouldn't be broken when fine-tuning.

Experimental Results

The model performance is measured via two metrics: Exact Match (EM) score and F1 score, where EM score represents the percentage of predictions matching corresponding ground truth answer, and F1 score is the harmonic mean of precision and recall.

Model name	EM/F1 (Dev)	EM/F1 (Test)
Baseline	34.55/50.28	-
Back Translation (BT)	32.46/47.45	-
Random Swap (RS)	35.08/50.11	-
Synonyms Replacement (SR)	34.03/49.21	-
Reduced Output Length (ROL)	34.82/50.92	-
Mixture of Experts	37.17/52.44	42.94/59.82
Mixture of Experts (ROL)	38.73/54.19	43.85/61.90
Mixture of Experts (RS, ROL)	38.22/54.27	-
Mixture of Experts (SR, ROL)	**39.27**/53.75	-
Mixture of Experts (with Generalist, ROL)	39.01/**55.13**	**43.88/61.93**
In-context Learning	17.28/39.94	-
In-context Learning (SR)	18.48/38.79	-

Table 2. Experimental results on validation and test sets

Model name	Precision (Dev)	Precision (Test)
Mixture-of-Experts Classifier	98.43%	99.45%

Table 3. Performance of the mixture-of-expert data domain classifier on dev and test sets

Based on our building blocks, we conducted experiments on various combinations of implemented techniques to observe their performance improvement. Surprisingly, the mixture of experts model with hyperparameter-tuning reached the highest scores in the validation set with 40.31 EM score and 54.75 F1 score. And the final submission of this approach was **ranked 4th place out of 56 submissions** in the test set, with 43.88 EM score and 61.93 F1 score.

Conclusion

After trying out many combinations of techniques, we concluded that the simple idea of mixture-of-experts worked very well on the target domain. This could be the result of drastically different distribution and quality seen in target datasets. We also tuned a few hyperparameters to generally improve model performance based on statistical analysis. Unfortunately, the metrics of in-context learning are subpar, probably due to the diminishing MLM inference ability resulted from model size reduction or lack of demonstration. Our future work includes identifying the root cause of the poor performance of in-context learning approach, exploring effectiveness of demonstration, and introducing more building blocks to observe their performance and composability.