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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Question</th>
<th>Context</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD</td>
<td>Crowdsourced</td>
<td>Wikipedia</td>
<td>50,000</td>
<td>10,507</td>
<td>-</td>
</tr>
<tr>
<td>NewsQA</td>
<td>Crowdsourced</td>
<td>News articles</td>
<td>50,000</td>
<td>4,212</td>
<td>-</td>
</tr>
<tr>
<td>Natural Questions</td>
<td>Search logs</td>
<td>Wikipedia</td>
<td>50,000</td>
<td>12,836</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Question</th>
<th>Context</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>DuRoC</td>
<td>Crowdsourced</td>
<td>Movie reviews</td>
<td>127</td>
<td>126</td>
<td>1248</td>
</tr>
<tr>
<td>RACE</td>
<td>Teachers</td>
<td>Examinations</td>
<td>127</td>
<td>128</td>
<td>419</td>
</tr>
<tr>
<td>RelationExtraction</td>
<td>Synthetic</td>
<td>Wikipedia</td>
<td>127</td>
<td>128</td>
<td>2693</td>
</tr>
</tbody>
</table>

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.

Data Augmentation

- **i**. Randomly replace a word with its siblings in the same sentence.
- **ii**. Replace with Synonyms: Replace some words with their synonyms.

In-context Learning

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

Approaches

Mixture of Experts

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

where $f_i$ is the output of input $x$ evaluated on expert $i$'s DistilBERT QA model, $f_{\text{generalist}}$ is the output from the generalist model, and $g_i$ 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 mlqaug. The augmentation will only be performed in context text with the answer phrases fully preserved.

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

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.

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,

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

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

Hyperparameter Tuning

- **i**. 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 models from producing over detailed predictions. This trick significantly boosted both EM and F1 scores.
- **ii**. 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.

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 performance resulting 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.