BoBA: Battle of BERTs with Data Augmentation
Ishira Fernando  Rich Desai
Department of Computer Science, Stanford University

Problem
Building QA models that generalize well to unseen data distributions that are distinct from the models’ training distribution is a difficult problem. Humans are innate at this task, which is known as domain generalization, but even state-of-the-art models on QA benchmarks such as SQuAD are known under perfunctory on-off-domain IOODs. Data domain generalization is essential to QA systems as they are expected to reason well to different applications, which may involve structural and contextually different language use.

We present BoBA: Battle of BERTs with Data Augmentation. BoBA combines Data Augmentation and Mixture of Experts (MoE) to improve domain generalization, by outperforming a DistilBERT baseline by 5.7 F1 points and 0.5 EM points.

Background
One of the most widely utilized NL models is DistilBERT, a knowledge distilled version of BERT. DistilBERT remains comparable to BERT in its performance, despite being over 40% smaller than BERT. Due to its performance and ease of deployment, DistilBERT is used as an atomic component in many QA systems today.

Augmenting training data through random transformations is well known to help with domain generalization and robustness. Due to the role-based nature of language data, augmentation can be very complicated. In EDQA, the authors propose a range of simple techniques by which to perform data augmentation on language-based classification models. We adopt this approach in our project to implement data augmentation to improve the domain generalization of BoBA.

Approach
Mixture of Experts are a class of ensemble models consisting of several individual expert models, each trained on one domain, which are “gated” by a learned gating function. The outputs of each expert is linearly combined according to the output of the gating function, which learns which expert to weigh more by conditioning on the input.

Data Augmentation
Adapting EDQA’s approach we implemented two different types of data augmentation: synonym replacement (SR) and random swapping (RS). SR replaces words that are not stop words with a random synonym with some probability $p_{SR}$. RS swaps words in the context according to a probability $p_{RS}$. Some examples are shown below.

Experiments

Dataset and Metrics
- 3 in-domain datasets: SQuAD, NewsQA, Natural Questions
- 3 out-of-domain datasets: DocRED, RACD and Relation Extraction
- EM (Exact Match) and F1 metrics to evaluate performance

Other Mixtures of Experts Approaches

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistilBERT</td>
<td>64.5</td>
<td>64.7</td>
<td>3</td>
<td>0.0101</td>
</tr>
<tr>
<td>DistilBERT SR</td>
<td>62.1</td>
<td>62.3</td>
<td>3</td>
<td>0.0101</td>
</tr>
<tr>
<td>DistilBERT RS</td>
<td>62.1</td>
<td>62.3</td>
<td>3</td>
<td>0.0101</td>
</tr>
<tr>
<td>DistilBERT SR+RS</td>
<td>62.1</td>
<td>62.3</td>
<td>3</td>
<td>0.0101</td>
</tr>
</tbody>
</table>

Table 1: Baseline results for the DistilBERT models trained on the in-domain training set.

Training BoBA
1. Train model $M_0$ on the 3 in-domain train sets with data augmentation.
2. Let expert $E_i$ of $M$ after fine-tuning on the i-th in-ood train set with data augmentation.
3. Train MoE model $R = f(E_1, E_2, E_3)$ on the 3 in-domain train sets without augmentation, where $f$ is the gating function.
4. Fine-tune and validate $B$ on the three out-of-domain train sets with data augmentation.

Table 2: Hyperparameters for training each expert; $p_{SR}$ and $p_{RS}$ are the random synonym percentage and synonym replacement percentage, respectively. We use the AdamW optimizer and Cross Entropy Loss.

Conclusions

We developed BoBA, a MoE that uses random swapping and synonym replacement augmentation along with fine-tuned unshared experts and DistilBERT gating functions, a success of improving the domain generalization of DistilBERT on QA tasks. We gain a 5.7-point increase in F1 score and 6.6-point increase in EM score. This shows that our model can be trained as an F1 of 59.03 and an EM of 40.69, indicating strong generalization to the new domains.

Over the course of our experiments we found that data augmentation requires careful fine-tuning on a case-by-case basis due to the significant distributional differences between domains. Furthermore, hyper-parameters’ largely learning rate and batch size appeared to have a large impact on the generalization of the model. While frozen experts appeared to not generalize as well as unshared experts, exploring models that had different numbers of frozen transformer blocks may prove insightful going forward. Finally, we were unable to explore other augmentation practices such as back translation or random insertion/deletion, which may further boost the performance of our model. We leave these avenues to explore in the future.