Robust QA Few-shot Learning

Stanford CS224N Default Project

Sammy Mohammed
Department of Computer Science
Stanford University
sammym@stanford.edu

Ha Tran
Department of Computer Science
Stanford University
hahntrn@stanford.edu

Andrea Collins
Department of Computer Science
Stanford University
acoll3@stanford.edu

Abstract

Because many real-world NLP tasks rely on user data that is not necessarily guaranteed to be in-distribution, it is critical to build robust question answering systems that can generalize to out-of-domain data after seeing only a few examples from that domain. We aim to build a question answering system using few-shot learning on top of DistilBERT that is robust to domain shifts. We were inspired by one of the two approaches described in Gao et al. (2020) to sample and append out-of-domain demonstrations to each training example when finetuning the model. We find that our basic approach of simply appending randomly sampled out-of-domain demonstrations to in-domain contexts does not improve model performance, and plan to keep exploring modifications and hyperparameter tuning to make improvements.

1 Approach

We used the baseline described in the RobustQA final project handout: a pretrained DistilBERT transformer finetuned on solely in-domain data.

We implemented from scratch one of the approaches described in Gao et al. (2020). Specifically, when finetuning the model on in-domain data, for each training example, we first found the top 50% most similar out-of-domain examples from the out of domain training set, using cosine similarity between the example contexts as a measure of similarity. However, our implementation of cosine similarity is a departure from the approach in Gao et al.. In that work, the authors used a pre-trained Sentence-BERT model to obtain embeddings for input sentences. We first implemented cosine similarity with a bag-of-words vector representation for each input example, and later experimented with using sentence transformers trained on DistilBERT. After obtaining the top 50% most similar out-of-domain examples, we randomly sampled one out-of-domain example and appended the context from this example to the context of the current in-domain training example.

2 Experiments

For all experiments, we used the evaluation metrics of EM (exact match) and F1 score to evaluate our experiments and our improvements on the baseline model.

- **baseline-03 (baseline model):** F1: 48.432, EM: 33.246

- **finetune-44 (append out-of-domain demonstration):**
  As our first attempt at incorporating demonstrations into our training, we wanted to try appending a related out-of-domain context from each class to the context of our training data, which is the in-domain train set. Our intuition was that by getting more exposure to the out-of-domain contexts, the model would learn to deal with those contexts during evaluation.
  - **Result:** Eval F1: 47.10, EM: 31.68.
    We hypothesize that this approach yields lower scores than the baseline because we increased the context size by 4 times by appending 3 demonstrations pulled from the out-of-domain contexts, which is also a much smaller pool. Furthermore, we neglected to add a separator token in between the context and demonstrations. This might have hurt the model's performance to have to parse through more context for the answer, making it harder to find the correct answer.
- **finetune-lr (append out-of-domain demonstration, train with smaller learning rate):**
  The Tensorboard plots of our loss, EM, and F1 scores for our first experiment with appending demonstrations showed that the EM and F1 scores seemed to increase to a point, then decrease. This observation led us to experiment with the learning rate, as we hypothesized that perhaps a larger learning rate led to the model overshooting the minimum loss during training. We therefore ran an experiment where we loaded in the model we trained in finetune-44 (trained with appending of demonstrations) and reran training with a learning rate of $3e^-6$ rather than the default learning rate of $3e^-5$. We trained again on the in-domain training datasets (SQuAD, NewsQA, and Natural Questions) and appended contexts from the out-of-domain traning datasets (DuoRC, Race, and RelationExtraction) using the same method as described above.
  - **Result:** Eval F1: 48.36, EM: 32.98
    We found that decreasing the learning rate improved our model's approach from our first experiment. We hypothesize that the smaller learning rate may have led to a more globally optimal set of learned parameters. Based on this experiment, we plan to further explore hyperparameter tuning.
- **Incorporating Mask Tokens:** One potential area that could have caused problems in our model was that our model was overfitting to the contexts we were appending. Bidirectional models like BERT require data masking, since some of the model layers have the potential to leak information about upper layers. Thus, not masking the sentences we pulled from the out of domain training set could result in our model not truly learning these parameters. Thus, we masked two of the words in each of the sentences we appended to a given context in training. The goal of this was to enable our model to learn more about the the out of domain training data, rather than ignore the data completely.
  - **Result:** Eval F1: 48.62, EM: 32.20
    This led to a modest increase in F1 score, but a reduction in the EM score. It seems that masking specific words caused a reduction in exact match score, though a lower reduction than our original out-of-domain context appending approach. This indicates that the change in masking did end up improving our model's ability to predict words, a subtask of the question answering task, but also improve over our context appending approach. We hypothesize the performance reduction occurred due to the large increase in the size of the contexts, and the masking only partially affected this. Further work needs to be done to determine how beneficial masking is.

3 Future work

Going forward, we plan to focus on a few ideas for improvements.

- **Separator Tokens and Sentence Separation:**
  We hypothesize that perhaps appending two long contexts together may lead to worse attention scores, since it is harder to encode information accurately for larger pieces of text; this also has implications for attention. We plan to experiment with adding separator tokens in between sentences within the contexts and demonstrations in our training data,

and sampling from and appending the most similar sentences in the out-of-domain training set rather than sampling entire contexts.

- **Hyperparameter tuning:** We plan to further tune the batch size, size of train set split, learning rate, number of epochs (when decreasing the learning rate we might need to run for longer for the loss to converge), as well as the number of words masked.
- **Prompt generation** Following *Making Pre-trained models Few-shot Learners*, we want to integrate automatic prompt generation in order to improve the quality of context augmentation. This could led to improved F1 and EM scores.

4 Appendix

	F1	**EM**
Baseline	48.43	33.25
Append out-of-domain demonstration + masking	48.62	32.20
Append out-of-domain demonstration + smaller learning rate	48.36	32.98
Append out-of-domain demonstration	47.10	31.68

References

[1] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. 2020.

[1]