JAM For HotpotQA

Mars “Saucy” Huang, Ashton “Salty” Tang, Juan “Spicy” Zambrano
{mschuang, ashting, jzm} @stanford.edu

Abstract

HotpotQA is a natural language question-answering dataset that tests a system’s ability to perform multi-hop reasoning over multiple paragraphs to produce an answer. A successful model must not only answer yes/no or a span within the text but also identify the supporting fact sentences that led to its conclusion. The model published with the dataset was trained to simultaneously predict the supporting facts, as well as the answer to the question (yes/no, text span); this architecture is challenging for the model to learn well, since one loss is used to update the weights for two different tasks. Thus, we developed a pipeline model for this task, consisting of a supporting facts classifier and a QA model. Our model achieved a 17 point increase in joint F1 compared to the baseline model.

Models

Table 1: Performance of pipelined versions of JAM on dev set compared to baseline

<table>
<thead>
<tr>
<th>Metric</th>
<th>Training Data Source</th>
<th>Answer F1</th>
<th>Supporting Facts F1</th>
<th>Joint F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAM</td>
<td>SQuAD + HotpotQA</td>
<td>54.68</td>
<td>48.64</td>
<td>47.95</td>
</tr>
<tr>
<td>JAM</td>
<td>HotpotQA</td>
<td>53.64</td>
<td>67.53</td>
<td>47.95</td>
</tr>
<tr>
<td>JAM</td>
<td>SQuAD + HotpotQA</td>
<td>51.32</td>
<td>64.85</td>
<td>47.95</td>
</tr>
<tr>
<td>JAM</td>
<td>HotpotQA</td>
<td>50.32</td>
<td>63.54</td>
<td>47.95</td>
</tr>
<tr>
<td>Baseline [1]</td>
<td>HotpotQA</td>
<td>44.44</td>
<td>58.28</td>
<td>21.95</td>
</tr>
</tbody>
</table>

As shown in Table 1, adding on the the yes/no/span neuron helped our model achieve a 3.36 EM and 3.79 F1 improvement. Pre-training with the SQuAD dataset also benefited our model with an extra 1 point improvement for both EM and F1.

Discussion and Future Work

Our model’s marked improvement over the baseline may be explained by three main benefits of a pipelined model: 1) The two models can specialize on separate tasks with separate loss functions; 2) Increased interpretability of results - we can visualize the intermediate supporting facts before feeding it to the QA model; 3) As the supporting facts classifier filters ten paragraphs to a few sentences, we can use a more complex model like BERT for the QA Module while still remaining computationally tractable. Experimenting with optimizers, hyperparameters, and the introduction of BERT all significantly boosted our results. Furthermore, we demonstrated that operating two large models in parallel at test time, although requiring careful partitioning of GPU resources, was computationally feasible. A future direction is to modify our system to operate with the full-wiki setting in HotpotQA, which is a perfect use case of our noise-filtering pipeline model. We would also like to brainstorm other avenues and datasets where JAM could be useful - any natural language processing task that requires a “filtering” step and then a more focused “answer generation” step could benefit from our general approach.

References


HotpotQA Dataset and Evaluation

Dataset

The dataset consists of 112,779 questions, the vast majority of which involve reasoning from supporting facts embedded within two paragraphs of text. Each paragraph corresponds to the first paragraph of 5 million+ Wikipedia articles. Questions require multi-hop reasoning about a bridge entity or comparing a common property between two entities in each paragraph. There are two modes of model evaluation: distractor and full wiki setting. We focused on the distractor setting, where two paragraphs that contain supporting facts for the answer are shuffled among 8 other paragraphs selected from the corpus as the most similar as determined by bigram TF-IDF.

Figure 2: JAM model architecture

Supporting facts module

The supporting facts classifier looks at each sentence for all input paragraphs and with a binary classifier predicts whether or not each is a supporting fact to answer the question. The model roughly consists of character and word embeddings, a Bi-Attention layer that fuses context and question information, a Self-Attention layer for the question-aware context to attend to itself, and an output layer of logits the same length as the number of sentences in all paragraphs, with each neuron predicting whether the sentence at that position is a supporting fact.

QA module

The question answering model takes in the concatenated supporting facts and a question as an input, and outputs an answer. The core of our QA model is BERT (Bidirectional Encoder Representations), pretrained using the SQuAD v1.1 dataset and fine tuned using the HotpotQA training data. Additionally, our model has an extra yes/no/span output channel to accommodate the wide variety of questions types HotpotQA has to offer. The span output layer is used to indicate if our answer should be a span from the input text, or should be a yes or no answer.