Understanding Attention for Question Answering models

Akhsha Kamath
Stanford University

Problem

Question Answering involves a short paragraph (context) and a question (query) with the goal to output the location of the answer within the given text. Since PCE-based methods require a large amount of computational time and costs, we explore and compare the performance of non-PCE trained Contextual Embeddings (PCE) based approaches - Bi-directional Attention Flow (BIAF), Dynamic Coattention Network (DCN), and FusionNet and also consider ensemble-based approaches of FusionNet to get an idea of attention focused models' effectiveness on the SQuAD-2.0 dataset.

Background

Dynamic Coattention Networks

Xiong and Zhang [2] proposed a "coattention model" to give attention to both the question and the context at the same time.

- Built a co-dependent representation of them simultaneously which they use to predict the starting and ending point of the answer within the context.
- Introduced a dynamic point decoder, which iteratively moves through the context to determine which start and endpoints in the context satisfy the question best.

FusionNet [3]

- Non-PCE reading comprehension model.
- Simple model encoding with RNN features like pre-trained word vectors, term frequencies, part-of-speech tags, name entity relations, and whether a context word is in the question or not.
- Encoder is used to learn a global, vectorized representation of the query sentence, followed by a convolution over the context word embeddings to learn a representation of each word within its local context.

Experiments

Our experiments mainly just consisted of training our model on the Squad 2.0 dataset and seeing how it performed on unseen dev sets of data; we trained 3 different models and saw their performance on the F1, EM and AUC@K metrics provided by Tensetree.

Coattention

- Converted the character-level answer spans from SQuAD 2.0 to token indexing.
- We used the Adam optimizer with learning rate 0.01. The weights and biases and hidden states were initialized to zero whereas the encoding weights were randomly initialized. The start and end indices were initialized to the beginning of the context. We used LSTM hidden layers of size 200, BLSTM1, thus producing hidden states of dimension 400. We used a maxpool size of 50.
- We stopped training after 100000 steps since the model’s F1 score no longer increased on the dev set after that point.

FusionNet

Since the original paper is evaluated on Squad 1, we prepared a OOV (Out of Vocabulary) taken to the beginning of each context, when the question is unanswerable.

- Learning rate 0.0001, adamax optimizer. Exponential moving average decay rate 0.999, maximun gradient norm for gradient clipping 5.0, and a dropout probability 0.25.
- FusionNet Ensemble: We use 6 models from 6 runs with slightly different hyperparameters (16 models in total), giving us a dev F1 of 67.725 and EM score of 65.23.

Analysis

<table>
<thead>
<tr>
<th>Model evaluated on</th>
<th>Dev NDC</th>
<th>FL</th>
<th>EM</th>
<th>AUC@K</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIAF Baseline</td>
<td>06.81</td>
<td>58.61</td>
<td>74.56</td>
<td></td>
</tr>
<tr>
<td>Coattention</td>
<td>04.67</td>
<td>38.29</td>
<td>51.34</td>
<td></td>
</tr>
<tr>
<td>FusionNet single</td>
<td>06.21</td>
<td>64.49</td>
<td>60.56</td>
<td></td>
</tr>
<tr>
<td>Fusion Net ensemble</td>
<td>05.81</td>
<td>64.38</td>
<td>61.79</td>
<td>74.53</td>
</tr>
</tbody>
</table>

Both, the FusionNet and the FusionNet ensemble methods perform better than our baseline. However, the coattention model performs far worse than the baseline.

Model robustness upon Adversarial Attack

We change the dev set by substituting two words in a sentence and by replacing words by random words to explore how our model behaves. We see a nearly linear correlation between the number of values changed and the drop in performance. While there is a performance drop, the linear relation also shows some level of robustness to changes in the input sentences.

Conclusions

We tried to replicate the coattention network with a similar behavior as shown in the paper, but resulted in significantly decreased performance. The FusionNet single and ensemble modules boost the performance on the question-answering task and show some level of robustness to adversarial attacks. We consider the following as possibilities for future work:

- Lower training time: We wish to speed the implementation of the DCN so it takes less time to execute.
- Combination of different types of attention: Developing a model structure combining the different types of attention layers either intrinsically in the model or externally.
- Initialization techniques: We could also explore the effect of alternative initialization schemes, such as Xavier initialization for our LSTMs [3].
- Training attention to improve robustness to adversarial attacks: Perhaps the next version of Squad might include context with misspelled words or grammatically incorrect sentences to encourage models that promote robustness.

References


samphaksanbintu@