Question Answering on SQuAD 2.0
by Boxiao Pan, Gaël Colas and Shervine Amidi

Introduction

Problem
• Task: Extractive question answering
• Dataset: SQuAD 2.0
• Challenge: Unanswerable questions

Current state-of-the-art:
BERT-based model from Google AI
EM = 86.7 ; F1 = 89.1

non-PCE category

Data
SQuAD 2.0 split:
• train = 129,941 examples
• dev = 6078 examples
• test = 5915 examples

Approach

A more powerful Word Embedding

<table>
<thead>
<tr>
<th>Word features</th>
<th>BiDAF word embedding</th>
<th>Character-level word embedding</th>
<th>Tag features: POS ; NER</th>
<th>Additional features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embeddings</td>
<td>Pretrained GloVe embeddings</td>
<td>Trainable character embeddings</td>
<td>One-hot encoding</td>
<td>EM = context-question exact match</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TF*IDF = term-frequency</td>
</tr>
</tbody>
</table>

Segment-Based Aggregation (SBA)

Approach Overview
• Slide a window over context
• Predict for each window
• Aggregate over all windows

Hyper-param Setting
• Window length = 50
• stride = 20

Deep Dynamic Co-attention (DDCo)
Adopted from [3], this idea generates a co-attention score by looking at context and question simultaneously, thus being able to leverage useful mutual information. A dynamic decoder is used to iteratively predict the start and end index at each time step.

Baseline

BiDAF – BiDirectional Attention Flow

A new loss term

Goal: penalize more predictions far from the true label

Formula

\[ L_{\text{dist}} = \lambda \cdot \sum_{i} |m(B_i) \cap p_{\text{pred}}| \]

3 types of penalization tested

Distance loss

- Square root (SR)
- Linear (L)
- Quadratic (Q)

Results (dev)

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF</td>
<td>55.1</td>
<td>58.2</td>
</tr>
<tr>
<td>BiDAF + SR</td>
<td>55.7</td>
<td>59.8</td>
</tr>
<tr>
<td>BiDAF + L</td>
<td>56.0</td>
<td>59.7</td>
</tr>
<tr>
<td>BiDAF + Q</td>
<td>56.6</td>
<td>60.5</td>
</tr>
<tr>
<td>BiDAF + DDC</td>
<td>57.3</td>
<td>61.7</td>
</tr>
<tr>
<td>BiDAF + SBA</td>
<td>58.4</td>
<td>62.1</td>
</tr>
<tr>
<td>BiDAF + Char (C-BiDAF)</td>
<td>60.6</td>
<td>64.0</td>
</tr>
<tr>
<td>C-BiDAF + Tags (Tag-BiDAF)</td>
<td>61.4</td>
<td>64.9</td>
</tr>
<tr>
<td>Tag-BiDAF + DDCo</td>
<td>60.7</td>
<td>62.5</td>
</tr>
<tr>
<td>Tag-BiDAF + SBA</td>
<td>62.0</td>
<td>65.4</td>
</tr>
</tbody>
</table>

Conclusion

• More powerful word embeddings significantly improved performance
• Segment-based approach can include more information
• A penalization term based on the distance of the distribution from the true label yields promising results

References