BIDAF PRO MAX FOR QUESTION ANSWERING

Zhengguan Dai garydai@stanford.edu
Qingyue Wei qywei@stanford.edu
Yitao Qiu yitaoqiu@stanford.edu
Stanford University

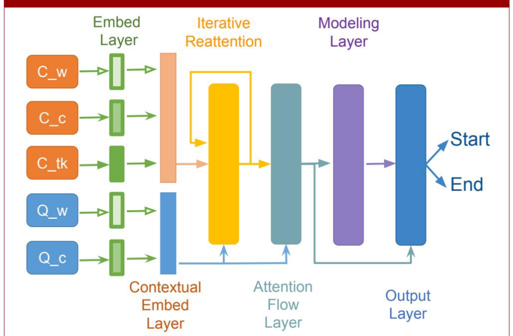
Problem/Background

As one of the ultimate goals of natural language processing (NLP), machine comprehension can be assessed by answering one or multiple questions with a chunk of text, such as a news article or a short biography. Most benchmark datasets contain questions whose answers are single entities or single tokens, while the Stanford Question Answering Dataset contains questions whose answers can be any sequence of tokens from the passage.

Datasets

- SQUAD 2.0 developed by Rajpurkar et al.[1]
  - 50,000 unanswerable questions written adversarially
  - Requires the model to determine whether the question is answerable, and answer the question when it is possible
- spaCy "en_core_web_sm" WordNet
  - The named entity and part of speech recognition are powered by spaCy's small English model based on WordNet 3.0
- English Word Frequency dataset
  - Published on Kaggle 1/3 million most common English Words

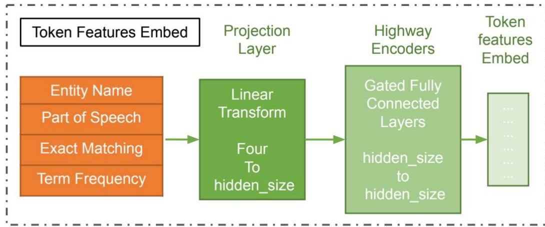
Model Overview



Model Details

We propose BiDAF Pro Max for machine reading comprehension tasks based on BiDAF [2]. Besides the basic model,

- we employ new token features with another token feature embed layer, and
- adopt two iterative reattention blocks before attention flow layer.

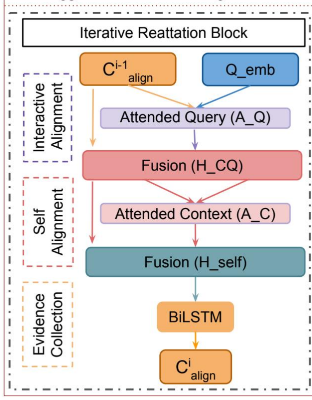


ENT: Named Entities Recognizer powered by spaCy, of which gives the type of noun phrase in the context, such as "*U.K.*" as *Geopolitical entity.*

POS: Part Of Speech Tagger also powered by spaCy. It parses context and tag each word as a sentence component.

EM: Exact Match context and question words, where if the word in context matches any words in the question with any capitalization version.

TF: Term Frequency of the word in general English. The dataset was published on Kaggle and based on Google Web Trillion Word Corpus.



In each of **reattention blocks [6]**, previous attentions are temporally memorized for current attention refinement which could avoid the problems of attention redundancy and attention deficiency.

$$A_Q = softmax(C_{align}^{i-1} Q_{emb}^T) Q_{emb}$$
$$H_{CQ} = \text{Fusion}(C_{align}^{i-1}, A_Q)$$
$$A_C = softmax(H_{CQ} H_{CQ}^T) H_{CQ}$$
$$H_{self} = \text{Fusion}(H_{CQ}, A_C)$$
$$C_{align}^{i} = \text{BiLSTM}(H_{self})$$

$\text{Fusion}(x, y)$:

$$g = \text{ReLU}(W_r[x; y; x \circ y; x - y])$$
$$h = \sigma(W_g[x; y; x \circ y; x - y])$$
$$\text{return } h \circ g + (1 - h) \circ x$$

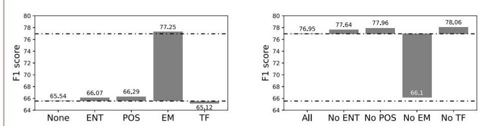
Results/Conclusions

Experiments	F1	EM
BiDAF (baseline)	60.90	57.74
BiDAF + Cond. Prob.	61.73	58.36
QANet	65.03	62.09
QANet + Add. Output Layer	66.13	63.13
BiDAF(C)	65.54	62.02
BiDAF(C) + Token Features	76.95	73.25
BiDAF(C) + Token Features + Iter. Attn.	**80.32**	**76.59**

* BiDAF(C) represent BiDAF + Character Embedding.

The experiments above show that

- Token features significantly improves the prediction. Compared with BiDAF(C), token features raise F1 score and EM score by >10
- The iterative attention further improves the model



- The absolute game changer is EM token feature. Single feature of EM can boost the F1 score from 65.54 to 77.25
- Although the ablation study shows No TF token feature configuration out performs that with four, with four token features and iterative attentions, the model works better than without TF

Reference

[1] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Association for Computational Linguistics (ACL), 2018.
[2] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603, 2016.
[3] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051, 2017.
[4] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. arXiv preprint arXiv:1804.09541, 2018.
[5] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. arXiv preprint arXiv:1608.07905, 2016.
[6] Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Zhen Huang, Furu Wei, and Ming Zhou. Reinforced mnemonic reader for machine reading comprehension. arXiv preprint arXiv:1705.02798, 2017.