FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension

Hsin-Yuan Huang1,2, Chenguang Zhu2, Yelong Shen2, Weizhu Chen2
1National Taiwan University, 2Microsoft AI+Research
momohuang@gmail.com, {chezhu, yeshen, wzchen}@microsoft.com

Introduction

• Teaching machines to read, process and comprehend text then answer questions is one of the key problems in artificial intelligence.
• Given the context, C = (c1, ..., cn), and the question, Q = (q1, ..., qm), we want to find the answer Ans. In the SQuAD task [1], the answer Ans is always (q1, ..., qm).

Conceptual Architecture for Existing Models

- Fully-Aware Fusion Network
  - Upgrade Normal Attention to Fully-Aware Attention:
    - $S(h_i, A^n) \rightarrow S(h_i, W^n, A^n)$: attention score calculation
    - Since $W^n$ is much larger than $A^n$, a proper choice of $S$ is crucial for good performance.
  - We will explore this in the experiments.

- Fully-Aware Self-Boosted Fusion
  - HoW (History of Word) Concept:
    - $S$ is always $w_1, \ldots, w_n$.

- Fully-Aware Multi-level Fusion
  - Fully-Aware Self-Boosted Fusion will fuse $C$ to itself to incorporate long-distance information through the proposed fully-aware attention.
  - For the SQuAD task, the answer finding layer is based on pointer network similar to previous models.

- In neural architectures, we define history of the i-th word, $H_W_i$, to be the concatenation of all the representations generated for this word.

- Attention Function
  - Performance comparison between FusionNet and existing models on SQuAD dev set (left) and comparison of model architecture (right).

References


Experiments

• In this section, we focus on the SQuAD dataset [1] and the adversarial SQuAD dataset [3]. We use the standard exact match (EM) and F1 score for evaluation.
• Ablation studies on SQuAD dev set are shown in the following tables, including comparison of attention score $C_i(x)$ (left) and comparison of model architecture [right].

• By enhancing normal attention with fully-aware attention, we have also improved a state-of-the-art model for natural language inference. An open-source implementation can be found at https://github.com/momohuang/FusionNet-NLI.