IID SQuAD Track: Expanding BiDAF and QANet Architectures for SQuAD

Eva Prakash[1], Josh Singh[2]

Department of Computer Science, Stanford University

[1]eprakash@stanford.edu, [2]jsingh5@stanford.edu

Stanford Computer Science

Goal

- Our task is to produce question answering (QA) systems that use the BiDAF and QANet architectures as starting points to perform well on the SQuAD dataset and improve upon the provided BiDAF baseline.

Background

- Machine reading comprehension and automated QA have become critical NLP fields.
- RNNs cannot provide the parallel computation and long-term memory needed for QA.
- A prominent NLP goal is to replace QA RNNs with full convolution or attention architectures like BiDAF and QANet.

Stanford University

BiDAF Model

- This baseline uses word-level embeddings.
- The bidirectional attention flow (BiDAF) layer uses similarities between all pairs of context and query words to find both Context-To-Question Attention and Question-To-Context Attention.

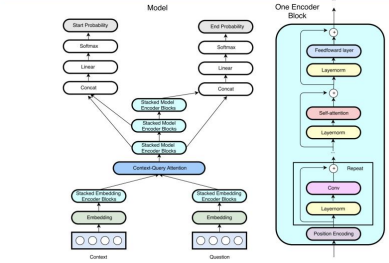
Augmenting BiDAF: Character-Level Embeddings

- We added character-level embeddings for the model to condition on the morphology of words.
- Using 2D convolutional and max-pooling layers, the character-level embeddings of each context and query are projected into the space of the word-level embeddings.

Augmenting BiDAF: Coattention Layer

- Coattention involves two-way attention between context and query words and attends a second time over attention outputs.
- We added coattention to aid in learning pairwise attentions.

QANet Architecture



- **Input Embedding Layer**
  - Concatenates word-level and character-level embeddings for the context and question.
- **Embedding and Model Encoder Blocks**
  - Stacks convolution, multi-head self-attention, and feed-forward layers with residual connections and layer normalization.
- **Context-Query Attention Layer**
  - Calculates attention using similarities between context and query words.
- **Output Layer**
  - Predicts the start and end position of the answer to the question in the given context.

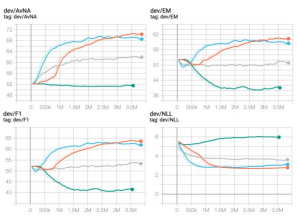
Augmenting QANet: Segment-Level Recurrence

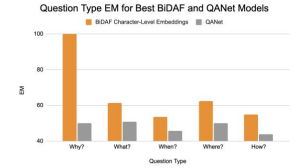
- We added the Transformer-XL self-attention segment-level recurrence mechanism to QANet to reuse the hidden states of previous context segments as memory for the current context segment and remember long-term dependencies.

Results

- We used the following evaluation metrics to evaluate our models on the SQuAD dataset:
  - **EM**: exact match accuracy
  - **F1**: harmonic mean of recall and precision
  - **AvNA**: accuracy of "Answer" vs "No answer" predictions
  - **NLL**: negative log likelihood
  - **Question Type EM**: EM of specific question types (i.e. "When?," "What?," "How?", "Why?", "Where?")
- Our dev statistics for our various experimental models are as follows:

MODEL	F1	EM	AvNA	NLL
Baseline	56.86	60.39	67.65	3.20
BiDAF Character-Level Embeddings	60.83	64.18	70.85	2.74
BiDAF Character-Level + Word-Level Embeddings	60.19	63.39	69.92	2.84
BiDAF Character-Level Embeddings + Word-Level Embeddings + Coattention	52.19	52.19	52.14	5.18
QANet	51.89	54.28	62.33	3.55
QANet + Transformer-XL (Initial)	52.20	52.19	52.76	5.16





Analysis and Conclusions

- Character-level embeddings are beneficial in modeling more specific sub-words as compared to full words.
- The pairwise attentions of coattention can find patterns that are possibly too complex.
- QANet Bottlenecks
  - Small batch sizes and number of encoder blocks can sink performance.
  - A small segment size for QANet + Transformer-XL may look at and remember too less context.