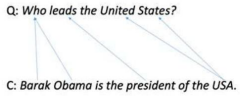
Question Answering with Self-Matching Attention

Ankit Patel, Xavier Arguello
Department of Computer Science, Stanford University
TA Mentor: Christopher Wolff

Introduction

The goal of an end-to-end QA system is to extract information from a given source (a context) such as a passage, document, image, etc. based on the user's request (a query). The effectiveness of such a system lies in its ability to provide concise and accurate answers.



Key Findings

Through our experiments, and without relying on pre-trained language models, we improved the baseline BiDAF model to perform well on Stanford Question Answering Dataset (SQuAD version 2). We showed how Self-Matching Attention (SMA), first used in R-Net, can mitigate information loss in the context-query attention mechanism, and provide considerable improvement to the baseline. In addition, we showed how to integrate convolution output of character embedding with word embeddings. We analyzed the contributions of these techniques in the final model to achieve our final results.

Related Work

The birth of the SQuAD dataset has ushered in vast research from the ML community towards building better Language Models. Question Answering in particular has gained momentum since that time. BiDAF is a great example of a successful implementation of a recurrent model with context-query attention mechanism at its core. Even with the advent of Transformer architecture, a non-recurrent approach, the primary way of encoding information about the passage and question has not changed. These Attention-based approaches continue to thrive and evolve.

For example, QANet achieved much higher score on the SQuAD leaderboard by combining ideas from BiDAF and Transformers. Models like QANet also show improvements in architecture can boost training performance.

The landscape seems to be changing yet again with the introduction of large pre-trained models such as BERT. These models reduce the task of implementing state-of-art models to a plug-n-play approach. These large LMs have stemmed from the ground-breaking research, first explored at the grassroots-level. So it pays to study them well!

Experiments

Data

We used the SQuAD 2.0 dataset with custom dev and test sets. (The official test set is unknown and reserved for final evaluation.)

- train: 129,941 examples
- dev: 6078 examples
- test: 5915 examples

The dataset contains records of (context,question,answer) triples of both answerable and unanswerable questions. The training set has one answer per question whereas the dev set has three answers for every question. In addition, 300 dimensional GloVe word embeddings and 64 dimensional character embeddings are provided.

Evaluation Method

We used the SQuAD official Exact Match (EM) and F1 metrics for quantitative evaluation of our model. EM score measures whether the predicted answer span exactly matches the ground truth. F1 score is the harmonic mean of precision and recall. Precision (p) is calculated as the number of correct words divided by length of predicted answer. Recall (r) is calculated as number of correct words divided by length of ground truth.

$$F1 = 2\,\frac{r\,p}{r+p}$$

To track the classification accuracy of no-answer predictions, we used the recommended Answer vs. No Answer (AvNA) metric. It simply states the percentage of correct predictions.

Experimental Details

We used the default configuration to train the baseline model:

- character embedding size: 64
- characters in a word: 16 maximum

After applying character-level embeddings, we trained the model with the following hyper-parameters:

- hidden state size: 100
- learning rate: 0.5
- char dropout rate: 0.05
- batch size: 32
- word dropout rate: 0.2
- L2 weight decay: 0.0005

The training time was ~15 minutes for each epoch on Tesla V100. All models were trained for a maximum of 30 epochs.

Method

Improvements to Embeddings

The main idea is to combine word and character-level embedding for every word in the embedding layer. Word-based models cannot deal with unknown (or misspelled) words i.e. words not in vocabulary. Character embedding alleviates this issue and helps with morphology. The challenging part is in doing it efficiently since each word is made up of multiple characters. For each word we thus generated a fixed-length vector of predetermined size by first convolving over character embeddings and then applying max-pooling in the dimension of word length. This vector is then concatenated with word embeddings before passing through a projection and highway network *f* as shown below.

$$e(c_i) = f([\,GloVe(c_i),\ CharEmb(c_i)\,])$$
$$e(q_i) = f([\,GloVe(q_i),\ CharEmb(q_i)\,])$$

e(ci) and e(qi) are then passed through two bidirectional LSTMs separately to produce contextual embeddings for context and query.

Improvements to Attention

One limitation of the attention mechanism, in the BiDAF model, is that the query-aware condensed representation of the context has limited information of the context (as seen below). Running inference on this condensed representation to get an answer is less than ideal. Important cues from the context stay hidden from the answer candidate.

$$\alpha_{i,j} = softmax_j(S_{i,j}) \qquad a_i = \sum_{j=1}^{M} \alpha_{i,j} q_j \qquad \beta_i = softmax_i(max_{j=1}^{M}(S_{i,j})) \qquad b = \sum_{i=1}^{N} \beta_i c_i$$

Our approach, shows that when an answer candidate has sufficient context, it improves the quality of inference. Specifically, we implemented Self-Matching Attention (SMA) to mitigate the information loss and derive an aggregate context representation that extracts evidence from the entire context w.r.t. current word and query.

Results

Model	F1	EM	AvNA	F1 (Test)	EM (Test)
Baseline	61.52	58.27	68.27		
Char Embedding	63.84	60.46	70.24		
SMA	66.92	63.62	72.14	65.67	62.47

Figure 3: Performance scores on SQuAD 2.0 dev and test sets

We saw modest improvements for both enhancements. Integrating learnable character embeddings lead to considerable improvement to the baseline. This improvement is attributable to the enhanced ability of the model to receive extra bit of signal to learn word meanings, and thus match words and infer answer spans better. The results also highlight the effectiveness of Self-Matching Attention. Scanning the entire context and aggregating signal relevant to the current context word and query, limits the information loss and produces better predictions.

Conclusions

The results highlight the effectiveness of Self-Matching Attention as described in R-Net. Scanning the entire context and aggregating signal relevant to the current context word and query limits the information loss and produces better predictions. Our model achieved 66.92 F1 and 63.62 EM scores on the dev set which is an 8% improvement over the baseline.

In future work we would like to explore different network structures such as GNNs to handle questions that require complex inferences.

References

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. CoRR, abs/1611.01603, 2016

Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. May 2017

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. CoRR, abs/1804.09541, 2018

Acknowledgments

Professor: Chris Manning
TA Mentor: Christopher Wolff