MultiBiDAF - A Multi-Sentence Reading Comprehension Model

Eitan-Hai Mashiah and Noam Barda

September 4, 2018

1 Introduction

Machine comprehension is a task of natural language processing wherein the algorithm is presented with a context paragraph and a query, and must answer the query using the data in the paragraph. While this is one of the classical tasks of NLP (Mccarthy 1976), interest in it has surged in recent years thanks to algorithmic advances (such as recurrent neural networks) and new benchmark datasets such as CNN/daily mail (Hermann et al. 2015), SQuAD (Rajpurkar et al. 2016), NewsQA (Trischler et al. 2016) and multiRC (Khashabi et al. 2018).

Despite its rising popularity, performance in this task remains challenging, and algorithms' capabilities are often below those of young children (Clark and Etzioni 2016).

2 Datasets

We will use two datasets in this project: SQuAD for pre-training and multi-RC as the actual challenge. Both will now be described.

SQuAD, the Standford Question Answering Dataset (Rajpurkar et al. 2016) was released in 2016. It is a large dataset with over 100,000 questions written by crowdsourced workers on wikipedia articles.

It has several defining characteristics:

1. A single context paragraph has several questions attached to it.
2. An answer is always a span of text in the paragraph. This is critical.
3. There isn't a predetermined list of answers to choose from.

Version 2.0 of this dataset (Rajpurkar, Jia, and Liang 2018) also adds questions for which no answer exists in the paragraph, requiring the answering algorithm to also know when to abstain from answering.

Currently, human performance on SQuAD (F1 = 89.452) far surpasses best machine performance (F1 = 74.422), leaving a wide gap for researchers to fill.

multiRC (Khashabi et al. 2018) is a fairly recent addition to machine comprehension datasets. It consists of ~6000 questions on

~800 paragraphs from 7 different knowledge domains.

It's defining characteristics are:

1. Several answers are proposed for each question.
2. Of which one or more can be true.
3. Answering requires reasoning over several sentences (specifically, 2-4) in the paragraph that are not a single span. This is the defining characteristic of this dataset.

The dependence on several sentences was created to force the algorithm to better "understand" the context paragraph, and thus allow better generalization.

Again, we see that human performance (F1 = 86.4) is significantly better than existing algorithms (F1 = 66.1), leaving a wide gap to be addressed.

3 Baseline Model

Our attempt at improving performance on the multiRC dataset will use an existing model as the baseline, altering it in ways that should improve its performance when applied to multiRC. The model we chose for this task is the Bidirectional Attention Flow for Machine Comprehension (BIDAF) model (Seo et al. 2016). We'll begin by describing the model at large, and will then focus on its main novelty, which is as the name suggests, its attention mechanism. See figure 1 for the complete model diagram as originally published.

The layer-wise structure of the BIDAF model is:

- Character-level embedding using a character-level convolutional neural network.
- Word embedding layer using pre-trained word embeddings (GLoVe).
- A standard bi-directional multilevel LSTM dubbed the contextual embedding layer.
- The bi-directional attention layer, to be described in the next paragraph.
- Another standard bi-directional multilevel LSTM dubbed the modeling layer.
- An output layer with two parts: One for predicting the start token and one for predicting the end token.

More accurately, the first two layers are input to the third layer, which is then input to the attention layer. This entire process, up to the attention layer, is executed separately for the context and the query. The output of the attention layer together with the unattended output from the contextual layer are both passed to the modeling layer, and that is used to predict the outputs.

The novel part of this model is the attention layer. Generally speaking, classic attention mechanisms (Bahdanau, Cho, and Bengio 2014):

- Summarize the entire context paragraph into a single dense vector by weighting the hidden representations of the RNN.
- Are dynamic, allowing the attention weights learned in a previous time step to affect the current time step.

- Are uni-directional, with the query attending the paragraph.

In the BIDAF model, on the other hand, the attention mechanism:

- Calculates attention for each time step and passes it on to the next layer, preventing early summarization.

- Is static, now allowing the attention weights from the previous time-step to affect the current time-step.

- Is bi-directional, from context to query and from query to context.

More concretely, the attention mechanism utilizes a T (context length) by J (query length) learnable similary matrix between the context and query words, whereby for each attending (context or query) word, the attended words are weighted using a softmax function based on the corresponding row or column of the similarity matrix, and then summed.

This model has been shown to achieve state-of-the-art results on multiple datasets, including SQuAD and CNN/DailyMail.

The specific implementation we'll use is the one in the allennlp project (Gardner et al. 2017). Allennlp is an open source NLP research library based on pytorch.

4 Our Model and Training Process

The two changes we need to make to better match BiDAF to multiRC are:

1. Change the output layer to predict up to 4 sentences as matches for the query.

2. Add a mechanism to decide which answers "fit" the chosen sentences sufficiently to be voted as positive.

We'll elaborate on each in turn.

4.1 Multiple Sentences

While the above-described output layer is a good fit for a dataset such as SQuAD, in which the answer is always a single span from the context paragraph, it is not fitting for multiRC, in which 2-4 sentences participate in the answer.

To handle this setting, we'll replace the original output layer by a new output layer with the following changes:

- Instead of predicting one start token and one end token, it will generate a probability distribution over all tokens in the dataset.

- 2-4 start tokens will then be chosen from the top 4 most probable words, while the cumulative probability of the chosen words does not exceed a learnable threshold parameter, T_s.

- We'll change the loss function to be the negative log probabilities of the correct start tokens, $\sum_i P_i$.

- Predicting the end tokens is now unnecessary, as they are by definition the end of the sentences whose start are the chosen words.

4.2 Multiple Answers

Identifying the pertinent sentences is not sufficient to decide which answers are right. Those answers must be chosen based on their agreement with the sentences. To do so, we'll make the following changes:

- We'll calculate a similarity metric between each answer and each chosen sentence. We'll use cosine similarity between TF-IDF weighted word counts in the sentences and the answers.
- We'll take the maximum such metric for each answer among all sentences.
- All answers whose maximum similarity exceeds a learnable threshold T_a will be flagged as true.

4.3 Training

As the number of questions in the multiRC is clearly not sufficient to properly train a complete neural network, even with pre-trained word embeddings, we'll opt for a multi-stage training process.

1. In the first stage, the model will be pre-trained on the SQuAD dataset.
2. In the second stage, the model will be fine-tuned on the multiRC dataset.
3. The thresholds for the sentences (T_s) and the answers (T_a) will be decided using exhaustive grid search using cross-validation.

Actual training will be done a Nvidia Tesla P100 GPU on google cloud.

5 Results

We report final results and learning curves both for the sentences found and for the correct answers found.

Following Rajpurkar, Jia, and Liang 2018 , we define $A(q)$ as the set of correct items and $\hat{A}(q)$ as the set of selected items. Precision (positive predictive value) is $\frac{A(q)\cap\hat{A}(q)}{\hat{A}(q)}$ and recall (sensitivity) is $\frac{A(q)\cap\hat{A}(q)}{A(q)}$.

Then, (macro-average) $F1_m$ is the harmonic mean of average precision and average recall of all questions, and $F1_a$ the harmonic mean of sensitivity and recall measured by first "OR"ing all the correct and chosen answers.

The chosen thresholds following cross-validation were $T_s =$ and $T_a =$

Pre-training on the squad dataset was carried for 20 epochs, with best-epoch performance of @@@. See figure 2 for a complete learning curve.

Training on the multiRC dataset was carried for 20 epochs, with eventual best-epoch validation performance of @@@. See figure 3 for a complete learning curve.

As the multiRC test set is not publicly available, test (as opposed to validation) results can only be had by sending the model's results to the dataset authors. This process takes a few weeks, and test results were not available in time for submission.

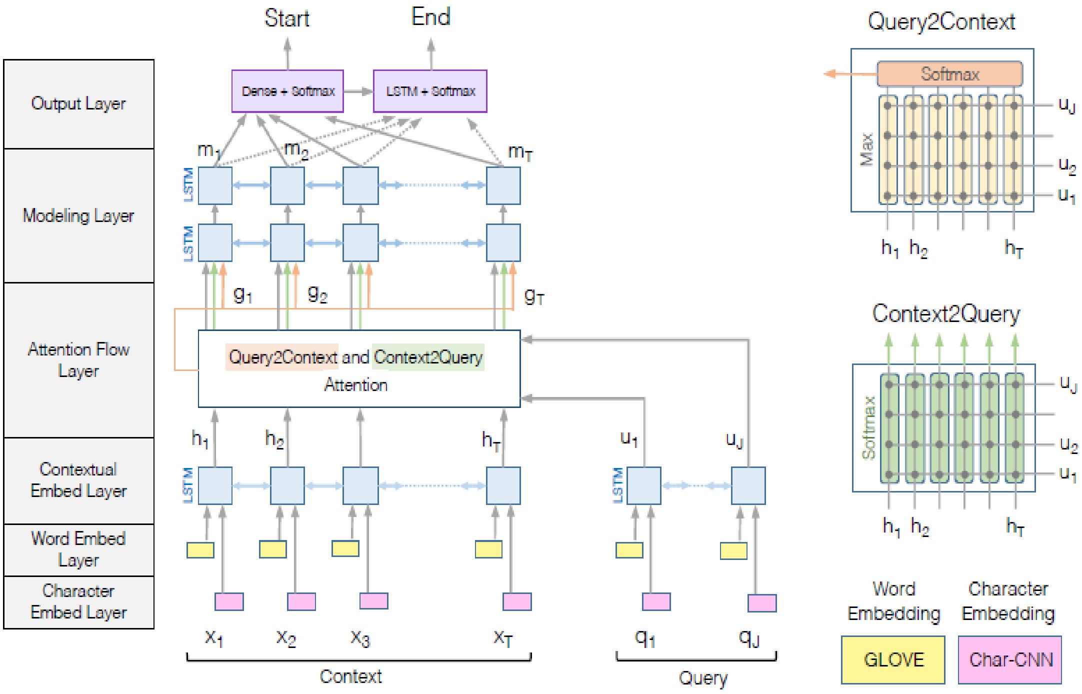

Figure 1: BIDAF Model Structure

6 Conclusion

Our validation results of @@@ fall short of the state-of-the-art results of $F1 = 66.1$, but not by much. As there is a significant decline in performance between finding the correct sentences and finding the current answers, it is possible that a better similarity calculation could achieve better results.

In this project we illustrated that a neural network model, utilizing a bi-directional attention flow mechanism, can achieve good performance on the multiRC dataset. To achieve this, the model has to undergo a modification of the output layer and subsequent fine-tuning.

Future directions of research could involve either similar changes to the output layer (e.g. perhaps a NER type layer, tagging spans as belonging to the answer), changes to the similarity calculation between answers and sentences or changes to the actual network.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (Sept. 1, 2014). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: arXiv: `http://arxiv.org/abs/1409.0473v7 [cs.CL]`.

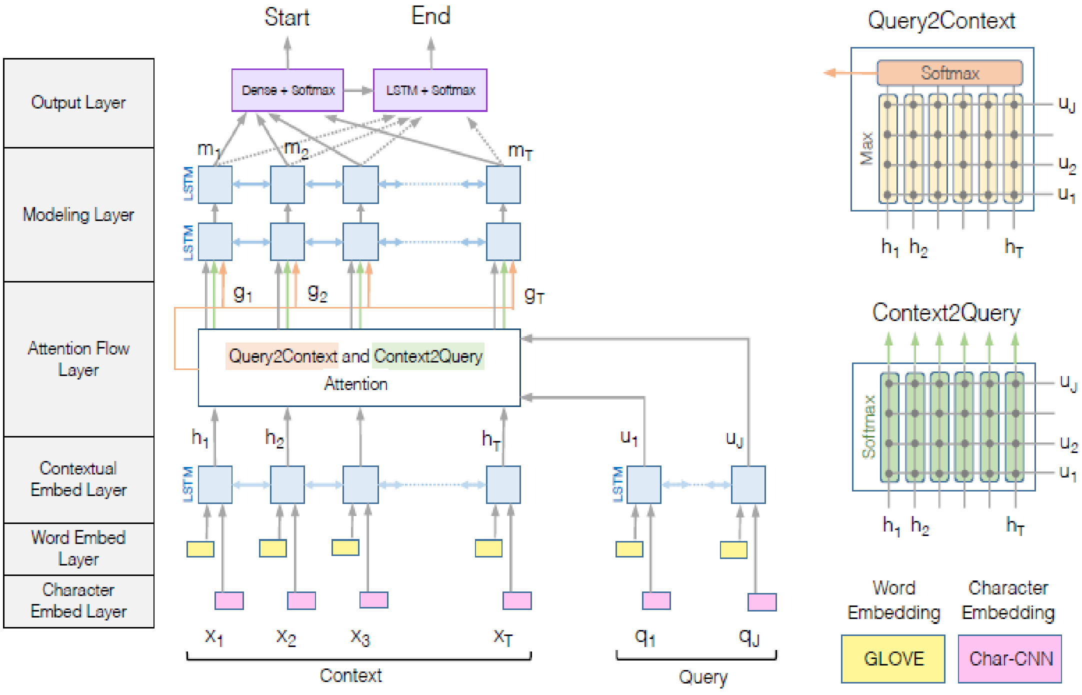

Figure 2: SQuAD Pre-training Learning Curve

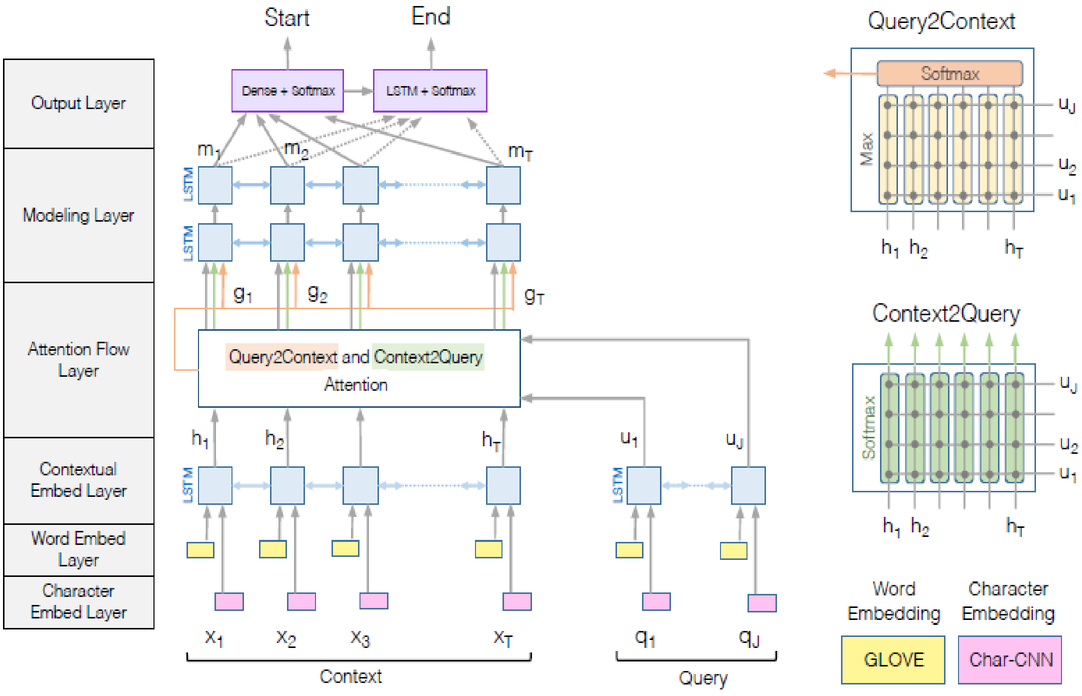

Figure 3: multiRC training Learning Curve

Clark, Peter and Oren Etzioni (2016). “My Computer Is an Honor Student — but How Intelligent Is It? Standardized Tests as a Measure of AI”. In: *AI Magazine* 37.1, p. 5. DOI: 10.1609/aimag.v37i1.2636.

Gardner, Matt et al. (2017). “A Deep Semantic Natural Language Processing Platform”. In:

Hermann, Karl Moritz et al. (June 10, 2015). “Teaching Machines to Read and Comprehend”. In: arXiv: http://arxiv.org/abs/1506.03340v3 [cs.CL].

Khashabi, Daniel et al. (2018). “Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 252–262. URL: http://aclweb.org/anthology/N18-1023.

Mccarthy, John (Sept. 1976). “An example for natural language understanding and the AI problems it raises”. In:

Rajpurkar, Pranav, Robin Jia, and Percy Liang (June 11, 2018). “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: arXiv: http://arxiv.org/abs/1806.03822v1 [cs.CL].

Rajpurkar, Pranav et al. (June 16, 2016). “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: arXiv: http://arxiv.org/abs/1606.05250v3 [cs.CL].

Seo, Minjoon et al. (Nov. 5, 2016). “Bidirectional Attention Flow for Machine Comprehension”. In: arXiv: http://arxiv.org/abs/1611.01603v6 [cs.CL].

Trischler, Adam et al. (Nov. 29, 2016). “NewsQA: A Machine Comprehension Dataset”. In: arXiv: http://arxiv.org/abs/1611.09830v3 [cs.CL].