Machine Comprehension with Bi-directional and Self-attention Flow

Tianpei Qian
Department of Statistics
Stanford University
Palo Alto, CA 94305
tianpei@stanford.edu

Ji Yu
Department of Materials Science
Stanford University
Palo Alto, CA 94305
jiyu318@stanford.edu

Abstract

For the CS224n default project, we implement an end-to-end machine comprehension system that performs well on the SQuAD. Our model features bidirectional and self-attention flow, which combines the architectures of two high-performing SQuAD models, R-Net and BiDAF. Our model achieves 76.5% F1 and 66.3% EM on the test set.

1 Introduction

Machine comprehension, the task of teaching machines to read text and answer relevant questions, is a crucial research topic in natural language processing (NLP). There has recently been significant progress in this field with the rise of deep learning in NLP research and the availability of large-size, high-quality machine comprehension datasets. In particular, the Stanford Question Answering Dataset (SQuAD) [1], which was released in 2016, contains 100,000+ question-answer pairs on 500+ articles and makes the training of deep neural networks possible. Since its release, a number of research groups have built various machine comprehension models on the SQuAD and the current best model has outperformed human performance in one of the evaluation metrics (F1).

In this project, we build a high-performing machine comprehension model on the SQuAD. Specifically, our model features bidirectional and self-attention and achieve 76.5% F1 and 66.3% EM on the test set.

The rest of the paper is organized as follows. In Section 2, we discuss previous high-performing models on the SQuAD. In Section 3, we describe our model in details. We then present in Section 4 the results of the experiments we have run on the SQuAD. In this section, we also visualize the attention layers of our model to illustrate the effectiveness of our model. Lastly, we make a detailed error analysis of our model and suggest possible improvements.

2 Background

There have been many high-performing SQuAD models, such as R-Net [2], BiDAF [3] and Dynamic Coattention Network [4]. Most of them involve innovative use of different attention structures, with self-attention and two-way attention being the two most successful ones. Specifically, R-Net contains a self-attention layer; BiDAF and Dynamic Coattention Network both contain a two-way attention layer, although differing in their specific structure. It is also worth mentioning that all of the attention structures are memoryless. In other words, the attention calculated at each time step are independent of previous or later steps. This allows the attention at each time step to learn independently and remain unaffected by possible mistakes made by other attendances. In our paper, we also adopt memoryless attention and combine the attention structure of BiDAF and R-Net to create a bi-directional and self-attention layer.

Apart from attention structure, those high-performing SQuAD models also use different word encoding techniques. BiDAF and R-Net both combine wording embedding and character embedding to represent text. However, R-Net uses RNN to further process character embedding into word representation while BiDAF uses CNN. For faster training, we choose CNN to process character embedding.

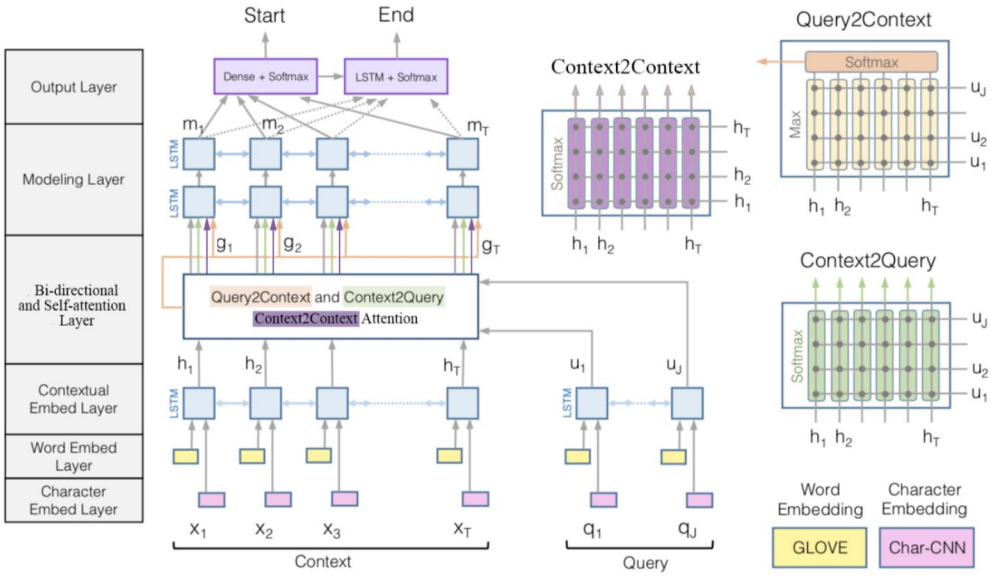

Figure 1: Model architecture

3 Approach

We give a detailed description of our model in this section. Our model consists of a character-embedding layer, a word-embedding layer, a contextual-embedding layer, a bi-directional and self-attention layer, a modelling layer and an output layer. The architecture is summarized in Figure 1.

3.1 Character embedding layer

Character-level encoding [5] has gained much popularity recently. It is known to handle out-of-vocabulary words well through encoding the internal structure of word.

For each word, we apply trainable character embeddings to each character. We then pass the character embedding to a 1-D CNN layer, treating the embedding dimension as the channel dimension. Lastly, we apply a max-pooling layer over the characters to get the final representation of the word.

3.2 Word embedding layer

We use the 100-d GloVe [6], the state-of-art pretrained word embeddings to get another word representation.

Unlike the original BiDAF, which further process character-level embeddings and word embeddings by applying a two-layer Highway Network, we simply concatenate the two embeddings to get the final word representation. In our implementation, this results in a better performance than if we use the highway network layer.

3.3 Contextual embedding layer

The final embeddings are then passed to a bi-directional LSTM layer. This produces hidden states that represent the contextual information of questions and contexts.

3.4 Bidirectional and self-attention layer

We use both directional attention and self-attention in this layer.

The bidirectional attention involves two types of attention, namely context-to-question attention and question-to-context attention. Suppose we denote the hidden states of a context by $\boldsymbol{h}_1, ..., \boldsymbol{h}_N \in \mathbb{R}^{2h}$ and denote the hidden states of a question by $\boldsymbol{u}_1, ..., \boldsymbol{u}_M \in \mathbb{R}^{2h}$. We define the *similarity* matrix by

$$\boldsymbol{S}_{ij} = \boldsymbol{w}_{sim}^T [\boldsymbol{h}_i; \boldsymbol{u}_j; \boldsymbol{h}_i \circ \boldsymbol{u}_j] \in \mathbb{R}$$

where $\boldsymbol{w}_{sim}^T$ is a weight vector of dimension $6h$. This matrix encodes the similarity between each question and context pair.

Based on the similarity matrix, we perform context-to-question attention as follows.

$$\begin{aligned}
\boldsymbol{\alpha}^i &= softmax(\boldsymbol{S}_{i,:}) \in \mathbb{R}^M \quad \forall i \in \{1,\ ...,\ N\} \\
\boldsymbol{a}^i &= \sum_{j=1}^{M} \alpha_j^i \boldsymbol{u_j} \in \mathbb{R}^{2h}\ \forall i \in \{1,\ ...,\ N\}
\end{aligned}$$

Next, we perform question-to context attention of the following form.

$$\begin{aligned}
m_i &= \max_j \boldsymbol{S_{ij}} \in \mathbb{R} \quad \forall i \in \{1,\ ...,\ N\} \\
\boldsymbol{\beta} &= softmax(m) \in \mathbb{R}^N \\
\boldsymbol{c} &= \sum_{i=1}^{N} \beta_i \boldsymbol{h}_i \in \mathbb{R}^{2h}
\end{aligned}$$

Apart from bidirectional attention, we also add an extra self-attention layer. Basically, we perform a context-to-context multiplicative attention. We denote it by $\boldsymbol{s}_1, ..., \boldsymbol{s}_N \in \mathbb{R}^{2h}$.

Finally, we get the output of this attention flow layer.

$$\boldsymbol{b}_i = [\boldsymbol{h_i}; \boldsymbol{a_i}; \boldsymbol{s_i}; \boldsymbol{h_i} \circ \boldsymbol{a_i}; \boldsymbol{h_i} \circ \boldsymbol{c}] \in \mathbb{R}^{8h} \quad \forall i \in \{1,\ ...,\ N\}$$

3.5 Modelling layer

We pass the question-aware and self-aware representations of the contexts to another 2 bi-directional LSTM layers to produce the final encoding of the contexts $\boldsymbol{m}_1^{(s)}, ..., \boldsymbol{m}_N^{(s)}$ for the start index prediction.

3.6 Output layer

A softmax layer follows the modelling layer to produce final probability prediction of the start/end index.

Specifically, we predict the probability distribution of the start index by

$$\boldsymbol{p}_i^s = softmax(\boldsymbol{w}_{p^{(s)}}^T [\boldsymbol{b}_i; \boldsymbol{m}_i^s])$$

Next, we pass $\boldsymbol{m}_1^{(s)}, ..., \boldsymbol{m}_N^{(s)}$ to another LSTM layer and obtain $\boldsymbol{m}_1^{(e)}, ..., \boldsymbol{m}_N^{(e)}$, which is then used to predict the probability distribution of the end index in the following form

$$\boldsymbol{p}_i^e = softmax(\boldsymbol{w}_{p^{(e)}}^T [\boldsymbol{b}_i; \boldsymbol{m}_i^e])$$

4 Experiments

In this section, we first present exploratory analysis of the SQuAD dataset. We then give the hyperparameter choices of our model. Finally, we evaluate our model from different perspectives, both quantitatively and qualitatively.

4.1 Exploratory analysis of the SQuAD

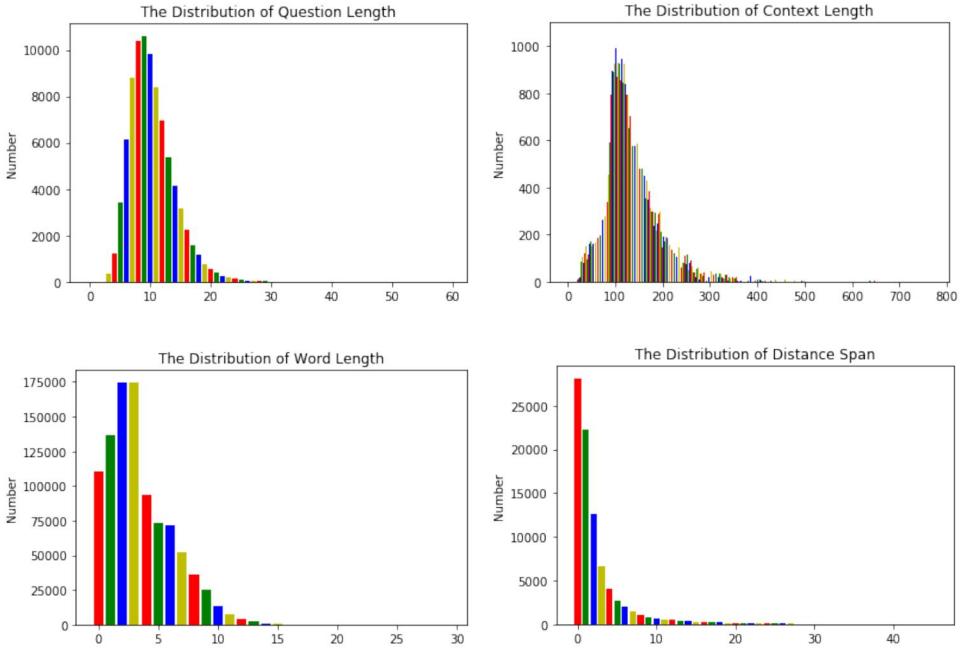

Figure 2: Exploratory analysis of the SQuAD

We look at the distributions of question length, context length, word length, and answer length. The results can be found in Figure 2.

We can see that most questions in the SQuAD are shorter than 30 words and most contexts are shorter than 400 words. For the words that appear in the SQuAD, most of them are short than 15 characters. Finally, the length of answers are almost always shorter than 30.

From our exploratory analysis, we decide to pad (or truncate) every context to length 400 and pad (or truncate) every question to length 30. Also, we choose the word length in character embedding to be 15. Moreover, we restrict the length of answer to be less than or equal to 30. That is, if we denote the start index by i, end index by j, then we must have $i \leq j \leq i + 30$.

4.2 Training details

Based on the original BiDAF paper, advice from TAs and our own experiments, we make the following choices of hyperparameter.

We use character embeddings of size 20, a window size of 5 and 100 filters for the character-level

Table 1: Performance of different models on the SQuAD

Model	dev		test	
	F1	EM	F1	EM
Baseline	41.3%	32.1%	-	-
BiDAF w/o char embed	73.7%	63.3%	-	-
BiDAF with char embed	74.4%	63.7%	-	-
BiDAF + self-attention	75.5%	65.2%	76.5%	66.3%
BiDAF (original paper)	77.3%	67.7%	77.3%	68.0%
R-Net	80.6%	72.3%	80.7%	72.3%

CNN. We choose 100 as the Glove embedding size. We have also experimented with larger embedding sizes but achieve little improvement. The sizes of all hidden states are 100. We use AdaDelta (Zeiler, 2012) optimizer with an initial learning rate of 0.5 and set the batch size to be 100. We also apply gradient clipping with a cap at 5. Finally, a dropout of rate 0.2 is applied to every CNN, LSTM and attention layers, as well as the fully connected layer in the output stage. The training is done on a NV6 Standard machine for 30k iterations.

4.3 Model performance

The performance for different versions of our model is summarized in Table 1. Although our model does not perform as good as either one of the BiDAF or R-Net implementation in their originla paper, it is still a huge improvement over the baseline. Also, adding self-attention to the BiDAF model does boost the model performance.

4.4 Attention analysis

We further illustrate the effectiveness of our model through analysis of the attention layer.

We select two questions that are paired with the same context.

- **Context**: tesla was generally antagonistic towards theories about the conversion of matter into energy . :247 he was also critical of einstein 's theory of relativity , saying :
- **Question1**: what was tesla 's attitude toward the idea that matter could be turned into energy ?
- **Question2**: which theory of einstein 's did tesla speak critically toward ?

All three types of attention (Context2Question, Question2Context and Context2Context) exhibits interesting patterns. Here we choose to illustrate the context-to-question attention (see Figure 3 & Figure 4).

Firstly, we observe that the attention exhibits very distinct patterns for the two questions, although they share the same context. Secondly, if an word appears both in the context and the question, the corresponding attention score is very high, which is expected since they have the same word embedding (e.g. energy in question 1, einstein in question 2). There are also other high attention scores that we find interesting. In question 1, antagonistic pays most attention to attitude. In question 2, relativity attends most to einstein. Finally, we can see that the overall attention is most activated in the first half of the context for question 1 while most activation appears in the second half of the context for question 2. Indeed, the first half of the context is more relevant for answering question 1 while the second half is more relevant for answering question 2. The above analysis shows that our attention is doing its job.

4.5 Error analysis

We analyze the errors our model make from both macro and micro perspectives.

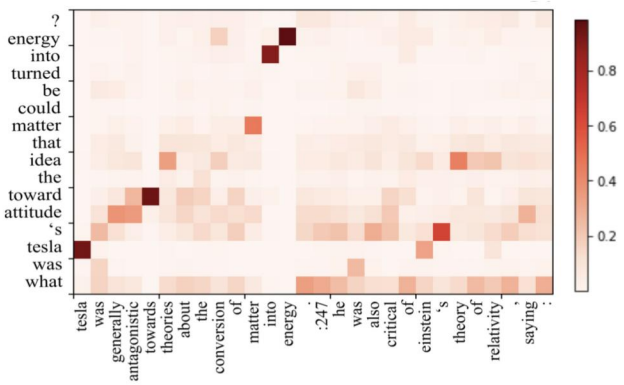

Figure 3: Visualization of C2Q attention for question 1

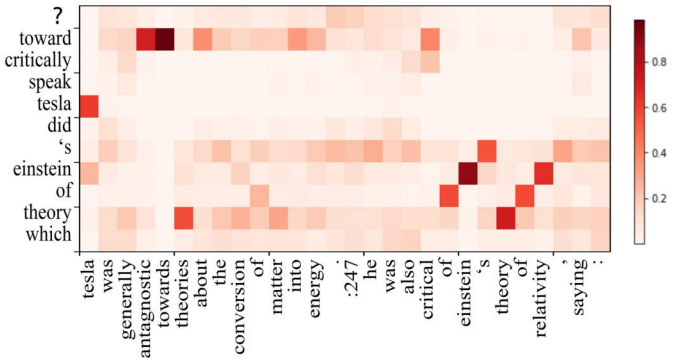

Figure 4: Visualization of C2Q attention for question 2

Firstly, we look at the model performance for each question type (see Table 2). Notice that we only use 1 golden answer to calculate the scores. The scores would be higher if we use all 3 golden answers.

We find out that our model performs best on "when" questions but does badly on "why" questions. It is possible that the those "why" questions more often involve complicated reasoning. Moreover, there is also an obvious negative correlation between model performance and length of the golden answers. The longer the true answer, the more difficult it is to get the prediction correct.

Next, we look at individual examples that our model has made wrong answer predictions on. We classify them into 4 types: (1) Wrong place of attention; (2) Imprecise answer boundaries; (3) Lack of common sense; (4) Complicated logical reasoning.

4.5.1 Wrong place of attention

In a few cases, our model pays attention to a completely wrong part of the context.

- **Context**: six-time grammy winner and academy award nominee lady gaga performed the national anthem , while academy award winner marlee matlin provided american sign language (asl) translation .
- **Question**:what actor did sign language for the national anthem at superbowl 50 ?
- **True Answer**: marlee matlin
- **Predicted Answer**:lady gaga

Table 2: Performance breakdown by question type

TYPE	F1	EM	Number	Ans Avg Length
When	85.1%	74.0%	684	2.30
Who	76.9%	67.9%	1237	2.86
Which	72.7%	57.3%	670	2.69
How	69.2%	50.7%	1123	3.05
What	67.5%	52.2%	5981	3.32
Where	65.0%	46.0%	458	3.39
Why	57.6%	22.1%	149	7.05
Other	50.5%	28.1%	89	5.06

Our model obviously pays attention to the wrong part of the sentence. We suspect our model is wrongly guided by the phrase "national anthem" and hence have its attention drawn to the first half of the sentence. This calls for a smarter or more complicated attention layer. If we could work on this project for more time, we would experiment with other attention forms, such as co-attention.

4.5.2 Imprecise answer boundaries

There are several cases where our model pays correct attention but gives imprecise predictions on answer boundaries.

In some questions, we think the predicted answer is quite reasonable.

- **Context**: on 7 january 1900 , tesla left colorado springs . [citation needed] his lab was torn down in 1904 , and its contents were sold two years later to satisfy a debt .
- **Question**: what happened to the things inside the lab after it was torn down ?
- **True Answer**: sold
- **Predicted Answer**: its contents were sold two years later to satisfy a debt

It seems that the SQuAD usually favor concise answers when there is ambiguity. We can try adding a small penalty for answer length. Alternatively, we can set a comparison threshold in the answer selection stage. As long as the probability of two answer spans are within that threshold, we always favor the shorter one.

That being said, there are also cases where the golden answer is more verbose than the predicted answer.

- **Context**:luther 's rediscovery of " christ and his salvation " was the first of two points that became the foundation for the reformation . his railing against the sale of indulgences was based on it .
- **Question**: how many points are there in the foundation of the reformation ?
- **True Answer**: two points
- **Predicted Answer**: two

Such inconsistency adds difficulty to the precise selection of answer boundaries.

4.5.3 Lack of common sense

In some questions, common sense is necessary to predict the correct answer.

For example, when there are multiple potential answers in the context, it is almost always that one answer should be favored over another. Let us illustrate the situation with the following example.

- **Context**: for exercise , tesla walked between 8 to 10 miles per day . he squished his toes one hundred times for each foot every night , saying that it stimulated his brain cells .

- **Question**: why did he walk ?
- **True Answer**: exercise
- **Predicted Answer**: stimulated his brain cells

In fact, we find the predicted answer to be more meaningful than the true answer, only that it is not grammatically sensible to select this answer. Given the limited number of similar cases, we think it is difficult for our model to learn such grammatical knowledge from the SQuAD directly.

The necessity of common sense is even more manifest in the following example.

- **Context**: the league eventually narrowed the bids to three sites : new orleans ' mercedes-benz superdome , miami 's sun life stadium , and the san francisco bay area 's levi 's stadium .
- **Question**: which louisiana venue was one of three considered for super bowl 50 ?
- **True Answer**: new orleans ' mercedes-benz superdome
- **Predicted Answer**: san francisco bay area 's levi 's stadium

To answer this question, the model needs to know that new orleans is a city in louisiana. We suppose that such relationship should be encoded in the similarity of their word embeddings but from this example it seems that word embeddings alone do not work. An external knowledge base may help in teaching our model such common sense knowledge.

4.5.4 Complicated logical reasoning

A few number of contexts contain logical reasoning that is too complicated for our model.

- **Context**: super bowl 50 featured numerous records from individuals and teams . denver won despite being massively outgained in total yards (315 to 194) and first downs (21 to 11) ...
- **Question**: how many yards did denver have for super bowl 50 ?
- **True Answer**:194

To answer this question correctly, the model should first understand the meaning of "outgained" and then choose the smaller one of the two numbers (315 and 194) following it. This presents a huge challenge for our model and currently we still do not see any simple modifications that would work for such reasoning.

5 Conclusion

We build an end-to-end machine comprehension model that performs well on the SQuAD. The model combines two high-performinig SQuAD models, R-Net and BiDAF and features bi-directional attention and self-attention.Our model achieves 76.5 % F1 and 66.3 % EM on the test set. From error analysis, we find out that our model still occasionally pays wrong attention and lacks common sense and the ability of making complex reasoning. Future work may focus on more complicated attention structures and the inclusion of external knowledge base to further boost the model performance.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text.

[2] Wang, W., Yang, N., Wei, F., Chang, B., & Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering.In Empirical Meth- ods in Natural Language Processing (EMNLP).

[3] Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603.

[4] Xiong, Caiming, Victor Zhong, and Richard Socher. "Dynamic coattention networks for question answering." arXiv preprint arXiv:1611.01604 (2016).

[5] Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W Cohen, and Ruslan Salakhut- dinov. Words or characters? fine-grained gating for reading comprehension. arXiv preprint arXiv:1611.01724, 2016.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Lan- guage Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 15321543, 2014.