A Hybrid Deep Learning System for Machine Comprehension

Gang Wu
Stanford Center for Professional Development
Stanford, CA 94305
gangwu@stanford.edu

Abstract

In this paper, a hybrid deep learning system for the machine comprehension task is presented. Particularly, I have combined the state-of-art bi-directional attention and self-attention mechanisms in to my model. In addition, different practical techniques, such as more accurate span prediction, adding extra input features etc., are applied in order to further reduce memory usage, help speedup runtime and boost performance. Experiments based on the SQuAD is performed on the proposed model. The F1 score 74.397 and EM score 64.418 is achieved using a single model on the test leader board.

1 Introduction

Machine comprehension is the ability for the machine to read text and then answer questions related to the text. It is a challenging topic while has become a very hot research field nowadays, due to the recent breakthrough in deep learning neural networks [1]-[6].

Recently, techniques based on the idea of attention has shown large performance improvement on the machine comprehension systems. The basic idea of attention is to make the system focus on a particular part of the context during the decoding process. Therefore, it can overcome the bottleneck issue encountered on the recurrent neural network [2]. In [2], the authors have proposed an enhanced attention scheme, which makes the attention to flow both ways: from question to context and from context to question. In [3], the authors have proposed a self-attention scheme which makes the content attends to itself. In [6], the authors show that machine comprehension can be done by just applying attention techniques without using recurrent neural network (RNN), as RNN is not able to be computed in parallel and therefore has a longer runtime.

In addition to different attention schemes, various other approaches have been proposed to improve the machine comprehension system. In [5], the authors show the performance can be greatly improved by adding extra input features such as exact match or POS. In [4], the authors have proposed an answer pointer network to better predict the starting and ending probabilistic distribution of the answer.

In this paper, I combined the state-of-art attention schemes in my machine comprehension system. I also added practical techniques to improve the performance and reduce the memory usage and runtime of the model. Experiments are performed using SQuAD[1] and the final results are evaluated on the test leaderboard. Detailed error analysis is also done on the proposed model.

2 Problem Definition

I formally define the machine comprehension problem in this section. The input to the machine comprehension problem is a sequence of context words denoted by $\mathbf{c} = \{c_1, c_2, .., c_n\}$ and a sequence

of question words denoted by $\mathbf{q} = \{q_1, q_2, ..., q_m\}$. Given the input, the machine comprehension problem outputs a pair of starting and ending position denoted by $\{p_s, p_e\}$ with the constraint $1 < p_s < p_e < n$. The segments of context words within this starting and ending position will indicate the answer to the question.

3 The Proposed Hybrid Model

In this section, I present the proposed hybrid deep learning system in more details. An overview of the proposed hybrid model is shown in Fig. 1.

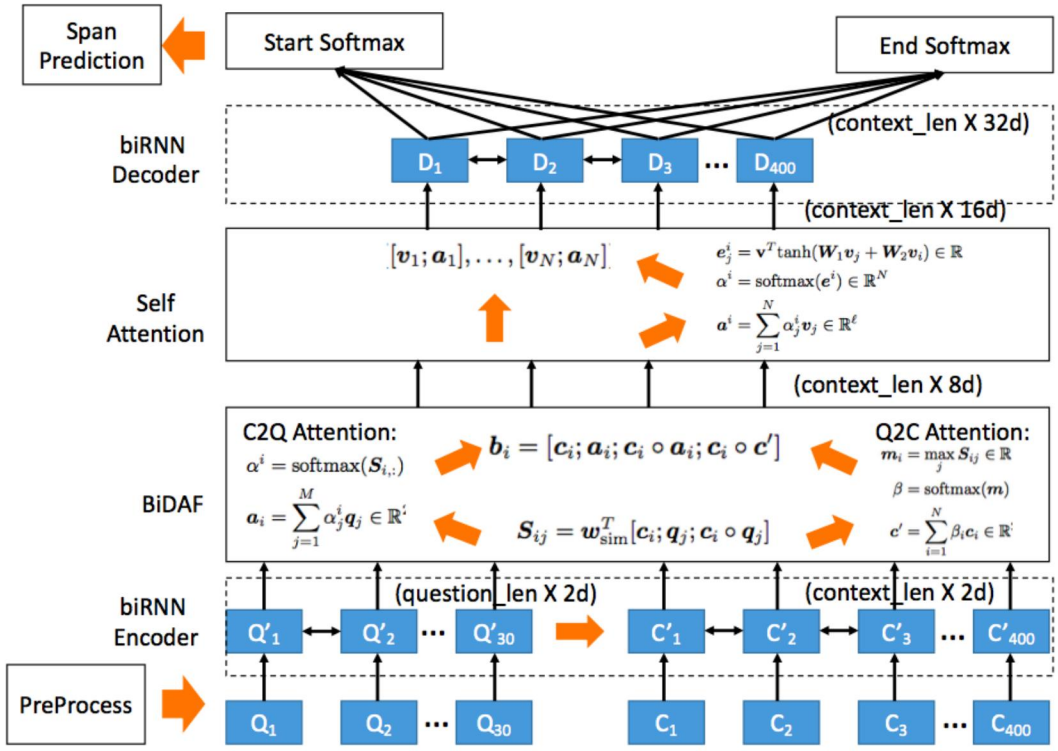

Figure 1: Overview of the proposed model.

3.1 Pre-process

Besides the basic pre-process used in the baseline model, a special pre-process method is added in order to implement the "Exact Match" idea in DrQA[5]. Particularly, during the pre-process stage, I extracted the three binary feature indicating whether the context word can exactly match to one of the question word in either its original, lower-case or lamma form. The extracted features are dumped out as extra train/dev.feature files, which will later be load in during the training and then concatenated with the context word embedding.

3.2 Bi-RNN Encoder

The purpose of the Bi-RNN encoder layer is to make the question / context embedding more "context aware". In addition, to make the context to be depended on the question, the baseline model shares the RNN weight between the question RNN and context RNN. To further enhance this, I used the final state of the question RNN as the initial state of the context RNN.

3.3 Bi-Directional Attention

The idea of BiDAF is to make the attention flow both ways: from the context to the question and from the question to the context [2]. After calculating the similarity matrix, the rest of the BiDAF implementation is quite straightforward. I use the BiDAF to replace the basic attention layer in the baseline model.

3.4 Self-Attention

Following the idea of R-Net [3], a self-attention layer is added after the bi-directional attention layer. To reduce the memory usage, I use the multiplicative attention instead of the addictive attention. As addictive attention consumes a lot more memory. Also, an extra normalization dividend is added based on the idea in [6].

3.5 Bi-RNN Decoder

After self-attention, a bi-directional RNN layer is added similar to [2][3].

3.6 Span Predictor

Instead of taking argmax as the baseline model, I implemented better span predictor to limit the answer span to be the one giving the best p[i]*p[j] within 10 words.

4 Implementation Details

In this section, I will present more implementation details about the proposed machine comprehension system. Particularly, I will discuss different practical techniques I used to reduce memory usage and runtime in Section 4.1 and Section 4.2. In addition, I will discuss the strategies I used to improve the system performance in Section 4.3.

4.1 Reduce Memory Usage

Neural network based machine comprehension system tend to have heavy memory usage. With only 8GB memory of Nvidia M60, it is very easy to run out of memory. Besides using multiplicative attention to save memory, as mentioned in Section 3.4, I listed below of the other practical techniques that I have used to reduce the memory usage:

- Reduce context length: Based on the statistics of the input data, a majority of the contexts is within 400 words. Therefore, a lot of memory can be saved by simply ignoring the few outliers and just use 400 as the context length.
- Reduce batch size: It can help to fit the bigger model, but it also affects the training speed. Therefore, I just used the default batch size 100 for most of my experiments.
- Reduce hidden layer size: I reduced the hidden layer size to 75, as I did not see major performance degradation by using a smaller hidden layer size.
- Use Nvidia K80: Instead of M60, K80 has 12GB memory. However, since this is an older generation GPU, the training process takes longer than Nvidia M60.
- Use multiple GPUs: With the interface of Tensorflow, I easily mapped different components of my model into different GPUs. This enables me to build a model with both BiDAF and self-attention without reducing the batch size or hidden layer size.

4.2 Speedup Training Time

Initial baseline model takes about 5 hr to train. However, with the increasing model size, the training time can easily increase to 12 hr or even more than 24 hr. In order to reduce the experiment turnaround time, it is very critical to improve the training time. Here I listed few techniques I used for the training time improvement:

- Use CudnnRNN: CudnnRNN is a different implementation of RNN in tensorflow. Since it is directly optimized based on Nvidia GPU architecture, it has a much faster runtime and a smaller memory usage compared with traditional RNN implementation. However, a major drawback I find is that the weights trained by CudnnRNN is not directly loadable into the system without a GPU. Even though Tensorflow has listed few APIs (e.g. CudnnCompatibleLSTMCell, CudnnLSTM Saveable) which supposed to help loading back the parameter into a regular RNN, but the lacking of documentation makes it very difficult to use. In order to submit my results to dev / test leaderboard, I used regular RNN for the final training.
- Use multiple GPU and increase the batch size: By mapping my model into two GPUs, I'm able to increase my batch size to 120 or even 150 during some experiments.

4.3 Performance Boost

Apart from having good machine comprehension model, some small tweaks can be very helpful to further improve the performance:

- Use glove840B300d: The default glove6B100d setting is trained based on a smaller data set and therefore can suffer from the out-of-vocabulary words. Thus, I used glove840B300d Common Crawl as the word embedding database during my training.
- Apply dropout and regularization: dropout and regularization help reducing overfitting in different ways: dropout randomly zero-out activations during the training and regularization restricts all the weights to be small. Since dropout are inserted in many places of my model, setting a large dropout value could zero-out too many activations too early during the training process. In my model, I applied both dropout and regularization.
- Use LSTM: I use LSTM RNN instead of the default GRU in my model and it usually gives better performance. However, LSTM also takes longer compile time and consumes more memory.
- Apply answer pointer network: I implemented the answer pointer network based on the idea in [4]. The initial hidden state I used for predicting the starting point is based on the discerption in [3]. However, this more complicated answer pointer network gives very similar performance compared with just using a simple fully-connected layer during my experiment. Therefore, a fully-connected layer is used in my final model to predict the probabilistic distribution of the starting and ending point.

5 Experimental Results

The proposed hybrid model is evaluated using SQuAD [1]. I use the AdaDelta optimizer [7]. More detailed parameters configuration is presented in Table 1. The training is performed using two M60 GPUs on the Azure platform. Particularly, the RNN Encoder, BiDAF, self-attention is mapped into one GPU. RNN Decoder and the rest of the model is mapped into another GPU.

Parameter	Value
learning rate	1.0
ρ	0.95
ϵ	1e-6
dropout	0.15
regularization	1e-5
batch size	100
hidden size	75
context len	400
embedding size	300

Table 1: Parameter configuration of the proposed model.

	Exact Match		F1	
	Dev	Test	Dev	Test
BiDAF [2]	67.7	68.0	77.3	77.3
R-Net [3]	-	68.4	-	77.5
DrQA [5]	69.5	78.8	70.0	79.0
Ours	**64.437**	**64.418**	**74.744**	**74.397**

Table 2: Performance comparision of various models.

	Dev EM	Dev F1
baseline	29.47	39.72
with self attention + biRNN decoder	47.90	62.53
with Bi-directional attention	50.11	64.41
with LSTM + span prediction	53.05	68.10
with glove840B300d	54.34	68.90
with regularization	54.35	69.30

Table 3: Comparing the performance impact of adding different components.

The proposed model has achieved EM score of 64.418 and F1 score of 74.397 with a single model on the test leaderborad. The convergence of the EM and F1 score is shown in Figure 2. In addition, the performance comparision between my model and several state-of-art models is presented in Table 2.

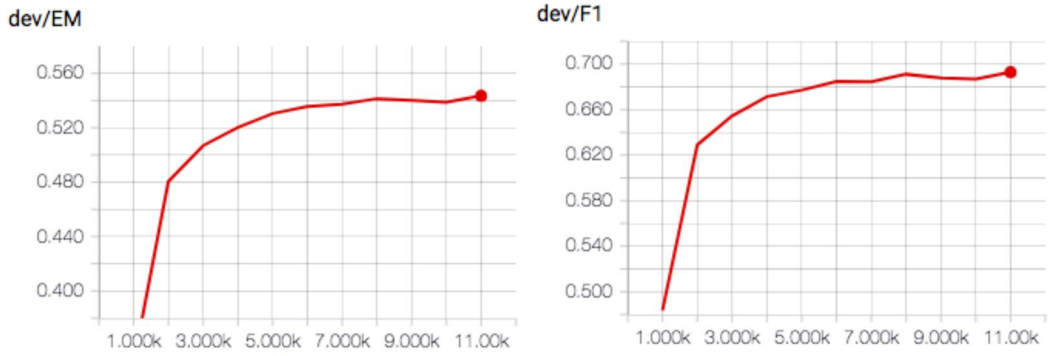

Figure 2: Convergence of EM and F1 score on dev set.

In Table 3, I compare the performance impact after adding different components and applying different techniques. Initially, the F1 score is 39.72 in the baseline mode. I get a huge performance boost and reached F1 score of 62.53 after implementing the self-attention and adding bi-directional RNN decoder. Another big performance improvement happens after replacing GRU with LSTM and adding the better span prediction, which reached F1 score of 68.10.

6 Error Analysis

6.1 Incorrect Boundaries

A common type of miss prediction is the miss alignment of the boundaries, i.e. the predicted answer contains more words than needed or having not enough words. A better layer to generate the probabilistic distribution of the starting and ending position might be helpful in this case.

- **Context:** To remedy the causes of the fire, changes were made in the Block II spacecraft and operational procedures, the most important of which were use of a nitrogen/oxygen mixture instead of pure oxygen before and during launch, and removal of flammable cabin

and space suit materials. The Block II design already called for replacement of the Block I plug-type hatch cover with a quick-release, outward opening door. NASA discontinued the manned Block I program, using the Block I spacecraft only for unmanned Saturn V flights. Crew members would also exclusively wear modified, fire-resistant Block II space suits, and would be designated by the Block II titles, regardless of whether a LM was present on the flight or not.

- **Question:** What type of materials inside the cabin were removed to help prevent more fire hazards in the future?
- **Prediction:** removal of flammable cabin and space suit materials
- **Answer:** flammable cabin and space suit materials

The following is a clear example that the span selection algorithm is not doing a good job, given the correct probabilistic distribution of each word. A simple improvement could be making the algorithm to choose a shorter answer give two answers which have the same probabilistic results.

- **Context:** A job where there are many workers willing to work a large amount of time (high supply) competing for a job that few require (low demand) will result in a low wage for that job. This is because competition between workers drives down the wage. An example of this would be jobs such as dish-washing or customer service. Competition amongst workers tends to drive down wages due to the expendable nature of the worker in relation to his or her particular job. A job where there are few able or willing workers (low supply), but a large need for the positions (high demand), will result in high wages for that job. This is because competition between employers for employees will drive up the wage. Examples of this would include jobs that require highly developed skills, rare abilities, or a high level of risk. Competition amongst employers tends to drive up wages due to the nature of the job, since there is a relative shortage of workers for the particular position. Professional and labor organizations may limit the supply of workers which results in higher demand and greater incomes for members. Members may also receive higher wages through collective bargaining, political influence, or corruption.
- **Question:** What is the potential earnings for a job where there are few skilled workers but many available positions?
- **Prediction:** high demand), will result in high wages
- **Answer:** high wages

6.2 Missing Question Information

In this example, part of the question ignored by the model and therefore a wrong answer is generated. This shows the model needs better encoding / more attention to the question.

- **Context:** The success of the first two landings allowed the remaining missions to be crewed with a single veteran as Commander, with two rookies. Apollo 13 launched Lovell, Jack Swigert, and Fred Haise in April 1970, headed for the Fra Mauro formation. But two days out, a liquid oxygen tank exploded, disabling the Service Module and forcing the crew to use the LM as a "life boat" to return to Earth. Another NASA review board was convened to determine the cause, which turned out to be a combination of damage of the tank in the factory, and a subcontractor not making a tank component according to updated design specifications. Apollo was grounded again, for the remainder of 1970 while the oxygen tank was redesigned and an extra one was added.
- **Question:** What happened to the Apollo program in for the rest of 1970 after the incident regarding Apollo 13?
- **Prediction:** oxygen tank was redesigned and an extra one was added
- **Answer:** grounded

Similarily, in the following example, the part "continent" is not well understood in the model and therefore a wrong answer is generated.

- **Context:** A resurgence came in the late 19th century, with the Scramble for Africa and major additions in Asia and the Middle East. The British spirit of imperialism was expressed by Joseph Chamberlain and Lord Rosebury, and implemented in Africa by Cecil Rhodes. The pseudo-sciences of Social Darwinism and theories of race formed an ideological underpinning during this time. Other influential spokesmen included Lord Cromer, Lord Curzon, General Kitchner, Lord Milner, and the writer Rudyard Kipling. The British Empire was the largest Empire that the world has ever seen both in terms of landmass and population. Its power, both military and economic, remained unmatched.
- **Question:** In which continent besides Asia were major gains made by the British Empire in the late 19th century ?
- **Prediction:** Middle East
- **Answer:** Africa

6.3 Out of Vocabulary Words

The quesiton containts incorrect words (which is out of vocabulary) and lead to wrong prediction of the model. Implement character level embedding might be able to help on this issue.

- **Context:** 20th Century Fox, Lionsgate, Paramount Pictures, Universal Studios and Walt Disney Studios paid for movie trailers to be aired during the Super Bowl. Fox paid f or Deadpool, X-Men: Apocalypse, Independence Day: Resurgence and Eddie the Eagle, Lionsgate paid for Gods of Egypt, Paramount paid for Teenage Mutant Ninja Turtles: Out of the Shadows and 10 Cloverfield Lane, Universal paid for The Secret Life of Pets and the debut trailer for Jason Bourne and Disney paid for Captain America: Civil War, The Jungle Book and Alice Through the Looking Glass.
- **Question:** Paramount paid fo, 10 Cloverfield Lane and which other film trailer to be aired during the game?
- **Prediction:** the Super Bowl
- **Answer:** Teenage Mutant Ninja Turtles: Out of the Shadows

7 Conclustion

A hybrid deep learning system for the machine comprehension task is presented in this paper. The proposed system combines the advanced machine comprehension architectures such as bi-directional attention and self-attention. In addition, various techniques are explored and implemented to improve the performance, memory usage and runtime of the model. The proposed system is evaluated on SQuAD and achieves competitive results compared with the state-of-art models.

References

[1] Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "Squad: 100,000+ questions for machine comprehension of text." *arXiv preprint arXiv:1606.05250* (2016).

[2] Seo, Minjoon, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. "Bidirectional attention flow for machine comprehension." *arXiv preprint arXiv:1611.01603* (2016).

[3] Wang, Wenhui, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. "Gated self-matching networks for reading comprehension and question answering." *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 189-198. 2017.

[4] Wang, Shuohang, and Jing Jiang. "Machine comprehension using match-lstm and answer pointer." *arXiv preprint arXiv:1608.07905* (2016).

[5] Chen, Danqi, Adam Fisch, Jason Weston, and Antoine Bordes. "Reading wikipedia to answer open-domain questions." *arXiv preprint arXiv:1704.00051* (2017).

[6] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, ukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in Neural Information Processing Systems*, pp. 6000-6010. 2017.

[7] Matthew D Zeiler. "Adadelta: an adaptive learning rate method." *arXiv preprint arXiv:1212.5701*, 2012.