Global Span Representation Model for Machine Comprehension on SQuAD

Sunmi Lee
Department of Computer Science
Stanford University
Stanford, CA 94305
sunmilee@stanford.edu

Jaebum Lee
Department of Computer Science
Stanford Univeristy
Stanford, CA 94305
jblee94@stanford.edu

Abstract

Machine comprehension of text is an important problem in natural language processing. A recently released dataset, the Stanford Question Answering Dataset (SQuAD), offers a large number of real questions, relevant context and answers created by humans through crowdsourcing. Given this more realistic dataset, we focus on the question answering task of machine comprehension. For this problem, we implemented a neural architecture that globally considers all answer spans. The architecture at its core implements the Recurrent Span Representation (RaSoR), with some modifications. Our model achieves a F1 score of 38 % and an EM score of 30 % with a potential for higher scores with additional hyper parameter tuning.

1 Introduction

Machine Comprehension (MC) is often thought of as one of the most challenging tasks in both machine learning and natural language processing research. The goal of machine comprehension is to train machines to understand a given passage and a question, then answer the question based on the related passage. The application of machine comprehension using natural language processing goes far and wide, with machine comprehension being used as tools for task handling and information processing in different consumer and enterprise technologies.

Before the introduction of the SQuAD dataset, other datasets such as RCTest and Cloze style MC examples were used when developing QA methods. However, the limited size and the lack of complexity of the datasets made it difficult for the developed end-to-end deep neural network models to successfully demonstrate robustness of the system. Instead, SQuAD provides a challenging testbed for evaluating machine comprehension algorithms, partly because compared with previous datasets, in SQuAD the answers do not come from a small set of candidate answers and they have variable lengths.

The SQuAD dataset presents a large number of real questions and their answers created by humans through crowdsourcing, and it is comprised of 100K ¡question, context, answer¿ triplets. Using the dataset, our problem is to train our model so that it can accurately answer a given question by returning the start and end indices of the answer span that is embedded within the context.

Many of the state-of-the-art question answering algorithms use answer extraction with separate probability calculations for start and end indices of the answer span. Our approach is different in that we enumerate all possible answer spans and consider all possible pairs of start and end indices globally. This would mean that we need to generate a $O(m^2)$ candidate answer span matrix for document length m, with a network that is cubic size with respect to the passage length, which leads to an unreasonably computationally expensive model. To overcome this, we restrict our maximum answer span in order to reduce the overall size of the fixed-length span representations significantly.

2 Related Work

There are many previous works that are relevant to our method. Rajpurkar et al. (2016) made SQuAD, on which they analyzed the dataset to understand the types of reasoning required to answer the questions, leaning heavily on dependency and constituency trees. Their logistic regression model achieved an F1 score of 51.0%, a significant improvement over a simple baseline, which was 20% at the time.

More closely related to our work is Wang & Jiang's end-to-end neural architecture using Match LSTM which substantially outperform the best results obtained by Rajpurkar et al. (2016) They also further adopt the Pointer Net (Ptr-Net) model developed by Vinyals et al. (2015), which enables the predictions of tokens from the input sequence only rather than from a larger fixed vocabulary and thus allows them to generate answers that consist of multiple tokens from the original text. There are two version of the model: a sequence model and a boundary model. Both models consists of an LSTM preprocessing layer, a match-LSTM layer and an Answer Pointer layer. More closely related to our model is the boundary model due to it similarly using recurrent networks and capture interaction between endpoints. However, the match-LSTM model have greedy training and evaluation, making them susceptible to search errors when decoding.

3 Proposed Model

In this section, we will review the structure of our model and the rationale behind it. Our model is based on the Recurrent Span Representation model (RaSoR) but we simplify it to boost the performance and efficiency of the learning process.

The goal of our model is that from the passage word vectors $P = (p_1, p_2, \cdots, p_m)$ and question words vectors $Q = (q_1, q_2, \cdots q_n)$, we need to retrieve correct answer span from the passage words vectors $p_s, p_{s+1}, \cdots, p_e$. To do so, instead of pin-pointing the starting index and the ending index of the answer span, we consider the starting and ending index of the answer span as one pair. This means that we have to generate pairs of all candidate answer spans. Since it is computationally expensive to consider entire m^2 possible answer spans, we limit the length of the answer span as maximum 15 words. That corresponds to more than 90 percent of entire data set.

3.1 Building attention matrix

Often the word vectors in answer spans and questions have a lexical similarity. To capture such relationship, we first build a large attention matrix between the word vectors in the passage and the question. We first feed the passage and question word vectors into a feed forward neural network and then multiply together to create a large attention matrix S. To get the normalized similarity between words in context and question word vectors, we apply softmax function to S.

$$\tilde{P} = W_p P + b_p \tag{1}$$

$$\tilde{Q} = W_q Q + b_q \tag{2}$$

$$S = \text{Softmax}(PQ) \tag{3}$$

$$\tag{4}$$

Then, S_{ij} now contains the similarity score between p_i and q_j from the passage and question word vectors respectively, to align each question word vectors to the passage word vectors, for each i, where $0 \leq i \leq m$,

$$q_i^* = \sum_{j=1}^{n} S_{ij} \tilde{q}_j \tag{5}$$

Finally, we concatenate each passage-aligned question word vectors to the corresponding passage word vectors to create

$$p_i^* = [p_i; q_i^*]$$

for the entire passage word vectors.

3.2 Answer Span layer

Now from $P^* = [p_i^*, p_2^*, \cdots p_m^*]$, we use bidirectional LSTM and concatenate the outputs from the forward and backward RNNs. We use two layers of bidirectional LSTMs over P^* by using the output of the first bidirectional LSTM layer as an input to the second bidirectional LSTM layer. The reason why we chose to use two layers of BiLSTM over P^* is because we are using the output vector at starting and ending index of the answer to represent corresponding answer span. To make sure that the two output vectors of answer span convey enough information about the context inside and outside the answer span, we have to use multiple layers of biLSTM to incorporate the context information into the answer span.

$$(h_1, h_2, \cdots, h_m) = \text{BiLSTM}(\text{BiLSTM}(p_1^*, p_2^*, \cdots p_m^*))$$

After running it through Bi-LSTM, we get the hidden representation for each paragraph word vectors $(h_1, h_2, \cdots, h_m)$. Now from those word vectors, we create the possible combinations of all answer spans. For example, the answer span that starts with i and ends with j is represented by the concatenation of h_i and h_j, $[h_i; h_j]$. Since the longest possible answer span is 15 words, there are $15m$ entries in the answer span layer.

$$S^{\text{Ans}} = [(h_0; h_0), (h_0; h_1), \cdots (h_0; h_{14}), (h_1; h_1), \cdots (h_1; h_{15}) \cdots]$$

Then, we apply feed forward neural network to S^{Ans} to get the score layer. Finally, we use another feed forward neural network to get the score for the each possible answer spans.

$$\widetilde{S^{\text{Ans}}} = W_{\text{Ans}} S^{\text{Ans}} + b_S \tag{6}$$

$$S^{\text{score}} = W_{\text{score}} \widetilde{S^{\text{Ans}}} \tag{7}$$

3.3 Learning

We simply use softmax cross entropy function to maximize the probability of the correct answer span.

4 Experiments

The hyperparameters we have tried to tune are: the dimensions of GloVe embeddings, the hidden size of LSTM cells, the hidden size of the matrices in Feed forward neural network, the dropout rate and learning rate. After many different combinations, the best result was achieved with 200 dimensional GloVe embeddings, 50 dimensional matrices for hidden states in LSTM and 150 dimensional matrix for feed forward neural networks, 0.1 dropout rate and 5 percent decay in learning rate in every 100 iterations, which is about 10k words.

We implemented our model in Tensorflow and trained on SQuAD training set using ADAM optimizer with a mini batch size of 100. We used Standard NV6 on Microsoft Azure and one epoch took about 40 minutes to run.

5 Results

We achieved 38 percent of F1 score and 31 percent of EM score on training set. Also 22 percent of F1 score and 17 percent of EM score on a testing set.

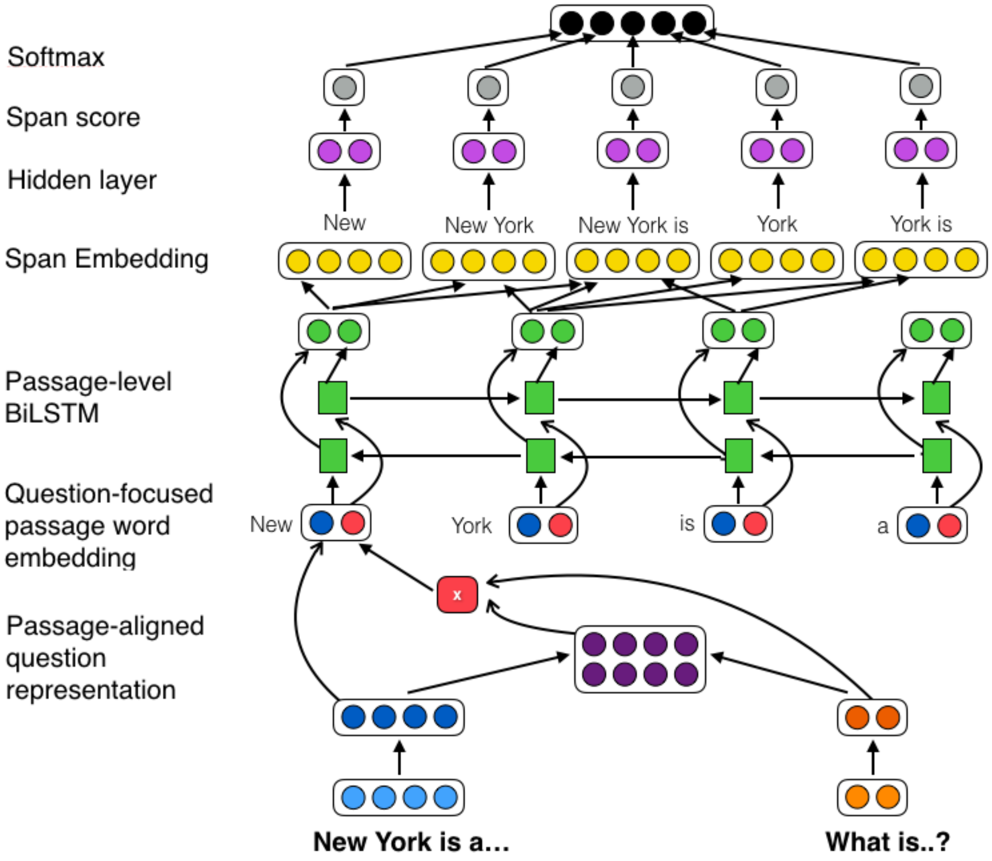

Figure 1: The outline of the proposed model.

	Train		Test	
	F1	EM	F1	EM
Baseline	28	19	16	11
Our model	38	31	22	17

Table 1: Result of our model

6 Analysis

6.1 Performances

The main optimization done on our model was considering all possible answer spans. This is beneficial because explicitly representing each answer span allows the model to be globally normalized during training and decoded exactly during evaluation.

Our model was able to correctly guess the questions that contain word vectors which are similar to the those in the context. Also, because our model specifically searches for the correct answer span rather than guessing the start and end index of the answer span independently, it could achieve a relatively high exact match score with a relatively low F1 score.

Examples showing correct guesses made by our model is shown in Table 1. We can see that our model was able to correctly guess ”Arthur Hutchings”, ”Texas Medical Center” and ”September 16, 1810”. These are exact matches to the correct answer, which our model is stronger at due to the global answer span search system.

Context	Question	Predicted
... as well as his musical genius , also fuelled his contemporary and later reputation . While his illness and his love-affairs conform to some of the stereotypes of romanticism , the rarity of his public recitals (as opposed to performances at fashionable Paris soires) led **Arthur Hutchings** to suggest that ”his lack of Byronic flamboyance [and] his aristocratic reclusiveness make him exceptional” among his romantic contemporaries , such as Liszt and Henri Herz	Who said Chopin was unlike his romantic contemporaries Liszt and Henri Herz ?	Arthur Hutchings
Houston is the seat of the internationally renowned **Texas Medical Center**, which contains the world 's largest concentration of research and healthcare institutions . All 49 member institutions of the Texas Medical Center are non-profit organizations . They provide patient and preventive care ...	Where in Houston is the world 's largest grouping of health-care institutions ?	Texas Medical Center
..., the day to celebrate it varied between September 16 , the day of Hidalgo 's Grito , and September 27 , the day Iturbide rode into Mexico City to end the war. Later , political movements would favor the more liberal Hidalgo over the conservative Iturbide , so that eventually **September 16, 1810** became the officially recognized day of Mexican independence ...	Which day eventually became the official day of Mexican Independence ?	September 16, 1810

Table 2: The examples of the correct guess

Context	Question	Predicted
... **Replays were traditionally played three or four days after the original game** , but from 199192 they were staged at least 10 days later on police advice. This led to penalty shoot-outs being introduced , the first of which came on 26 November 1991 when Rotherham United eliminated Scunthorpe United .	When are replays played?	26 November 1991
But in statistical mechanics things get more complicated . On one hand, **statistical mechanics** is far superior to classical thermodynamics , in that thermodynamic behavior, such as glass breaking , can be explained by the fundamental laws of physics paired with a statistical postulate ...	What are superior to classical thermodynamics ?	classical thermodynamics

Table 3: The examples of the correct guess

6.2 Error Analysis

There are several aspects where our model fails in particular. Our model fails to recognize subtle lexical differences in the word usage. For the first example in Table 2, the question is ‘When are replays played?’. The word ‘when’ does not refer to the date or time but rather means ‘cases’. Due to this ambiguity, our model searched for a date in the context and answered ‘26 November 1991’

Another prevalent error is that sometimes our model detects a ”strong relationship” between words in the question and words in the context even though those two are not relevant to the answer, and our model fails to distinguish between them. For example, the question is ‘What are superior to classical thermodynamics’ and the answer was ‘statistical mechanics’ according to the passage. However, since ‘statistical mechanics’ has been repetitively used in the context and has a high similarity to the actual question, the model answered ‘statistical mechanics’.

7 Conclusion

We proposed an architecture that efficiently builds fixed length representations of all spans in the dataset document with a recurrent network, which improves performance over some baseline implementations. However, in order to achieve a higher evaluation score, we could further extend and tune the parameters of the model such as increasing word dimensions and increasing hidden layer size.

Beyond simple parameter tuning, we could increase the answer span size (which is 15 right now) to 30 or more. This would allow for a more robust search since the model will now consider a larger permutation of start and end indices, leading to a greater sample size to take the answer span with the maximum probability.

Another change we could implement is using a better GloVe dataset. We worked with the given default dataset, but there are larger and more comprehensive GloVe datasets available, and using them would help with increasing the accuracy and performance.

A significant future changes in implementation could be considering using a convolutional neural network. Since the model only considers start and end indices, it would be beneficial to have a deeper layer of context matching in order to further encode the context information to the answer spans. This would allow for a more learned representation of span embeddings and hence would lead to a higher success rate.

8 References

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

[2] Kenton Lee, Tom Kwiatkowski, Ankur Parikh and Dipanjan Das. Learning recurrent span representations for extractive question answering. *In Proceedings of ICLR*, 2017

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

[4] Oriol Vinyals, Meire Fortunato, Navdeep Jaitly. Pointer Networks. *arXiv preprint arXiv:1506.03134*, 2015.