Applied Deep Learning
final report

r05922021, r05922006, r05922027, b01902106

January 14, 2017

1 Introduction

Introduction In this assignment, we were given two datasets, including Stanford Question Answering Dataset(SQuAD) and TOEFL. The goal is to solve the multiple choice question answering problem. We need to model the correlation between the context and the question, and to find out the answer in the choices. Here we use attention mechanisms to focus on a target area of sentence in the context. The experiment result shows that attention mechanisms can well model the right answer.

2 Model

2.1 BiDAF

This model is a hierarchical muti-stage process. Each layer model the different feature of our training data, and use difference of the start position and the end position of the ground truth with the prediction as the loss function. Then, the error will backpropagate through the layers to update the weight in each layer.

Character Embedding Layer In this layer, the model will map each word to a low-dimension vector by using character level CNN. First, characters are embedded into vectors, and use CNN to model the fix-size representation of each word.

Word Embedding Layer The word embedding layer will directly model the relation of a word and its surrounding words. Here, we use pre-train GloVe word embedding to be our fix-size representation of each word. The final representation of each word will be the concatenation of the character embedding and the word embedding.

Then, we can use these word representations to represent the context and the query, and pass into a two-layer highway network to get the final representation matrix of context and query.

Phrase Embedding Layer In phrase embedding layer, we take the previous layer output as input and use this layer to model the sequential information in the context and the query with a bi-direction Long Short-Term Memory. The concatenation of bi-direction LSTM model output (forward and backward) is the output of phrase embedding layer. Finally, we will get two phrase-level embedding, including context's phrase-level embedding H and query's phrase-level embedding U.

Attention Flow Layer In this layer, we compute a bi-direction attention from context to query as well as from query to context. We calculate the attention from the similarity matrix $S \in R^{T \times J}$, between the context phrase-level embedding H and query phrase-level embedding U.

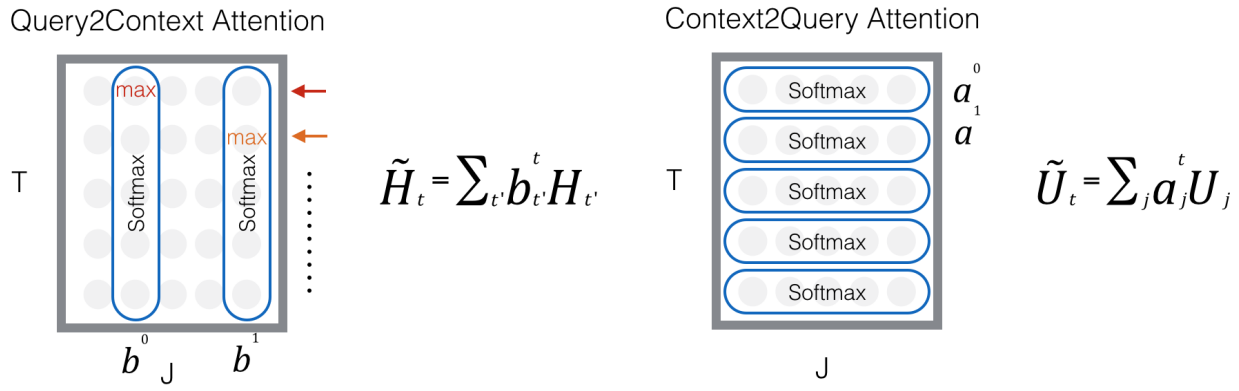
$$S_{tj} = \alpha(H_t, U_j) \in R$$

Alpha is a linear model which compute the similarity of two input vectors H_t and U_j.

$$\alpha(h, u) = w_{(S)}^T [h; u; h \circ u]$$

$w_{(S)}^T$ is the weight of the similarity model. $[;]$ is vector concatenation and $\circ$ is an elementwise multiplication.

Figure 1: Attention Flow Layer



Context to Query Attention C2Q attention focus on which query words are more relevant to each context word. For each context word t, we can get the weighted distribution α_t over query words. The final representation vectors $\tilde{U}$ will be the weighted sum of query vector for each context word.

Query to Context Attention Q2C attention focus on which context words have the closest similarity to one of the query words and hence critical for answering on the query. For each context word t, we need to find the most similar word in query and use the weighted distribution of the query over context length. The final representation vector $\tilde{H}$ will be the weighted sum of context vector for each context word.

Finally, the representation G will be the output of a linear model β function. The input is the concatenation of the attention vector, the phrase-level embedding vector and their elementwise multiplication vector.

$$G_t = \beta(H_t, \tilde{U}_t, \tilde{H}_t)$$

$$\beta(h, \tilde{u}, \tilde{h}) = w_{(G)}^T [h; \tilde{u}; h \circ u; h \circ \tilde{h}]$$

Modeling Layer In modeling layer, the input is the representation from the previous layer. we use this layer to model the interaction among the context words and the query words. Since, we have captured the attention representation from previous layer, so this layer model the different information to the phrase embedding layer. We also use a bi-direction Long Short-Term Memory to construct this layer. The output the this model M is expected to contain the information interaction between the context and the query.

Output Layer In output layer, our loss function is designed to model the start position (p^1) and the end position(p^2). First, we pass the concatenation vector of attention layer and modeling layer to model the start position. Then, pass the modeling layer representation M to another LSTM model to obtain the modeled representation M^2, and pass concatenation vector of attention layer and second modeling layer M^2 to model the end position.

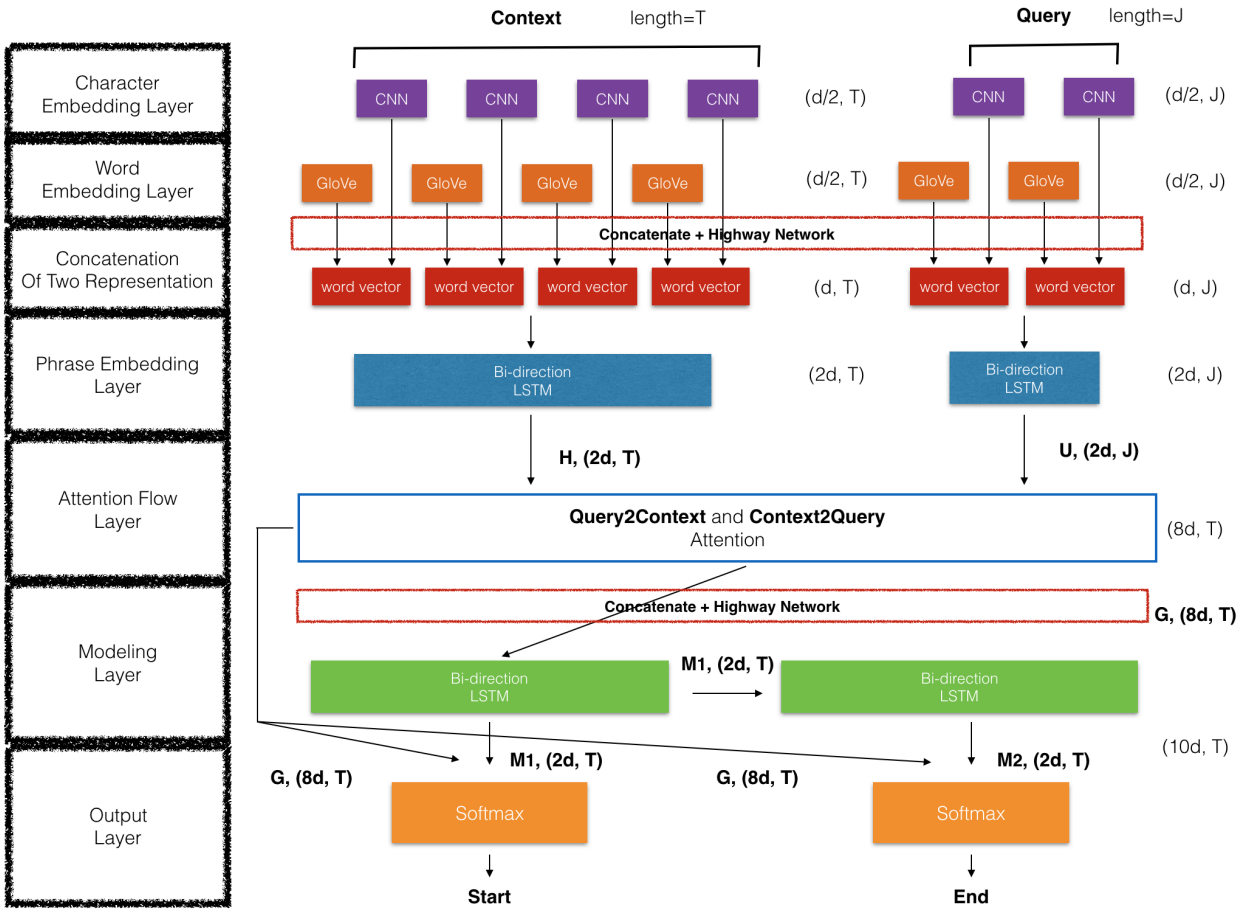
Finally, the loss function is the sum of the log probabilities of the true start position (y^1) and the end position (y^2) by the predicted distributions, average over training data N.

$$p^1 = softmax(w^T_{(p_1)}[G; M])$$

$$p^2 = softmax(w^T_{(p_2)}[G; M^2])$$

$$Loss = -\frac{1}{N}\sum_i^N (log(P^1_{y^1_i}) + log(P^2_{y^2_i}))$$

Figure 2: Bi-direction Attention Flow Model(d is the embedding dimension size)



2.2 Random Forest

This model ensembles the results of decision trees to do classification job. Each decision tree uses selected features to classify problems.

Method We use pre-trained word embedding from GloVe. For each problem, we use mean value of embedding to represent a sentence. We concatenate the embedding of question, context, real answer, wrong answer as positive training example and concatenate the embedding of question, context, wrong answer, real answer as a negative training example.

Since there are 4 choices and only in each problem, we can get 6 training example in each problem and total 6*N in N problems. By using those training examples, we can know if a choice is more probable than other choices.

Inference After we get the prediction of choice pairs, we use voting to choose the answer, if a pair (question, context ,choice (A), choice (B)) is predicted as label 1, it means that choice (A) is a more probable answer than choice (B), then choice (A) will get one point. Finally, we choose the most votes choice as the answer.

2.3 Information Retrieval Based Search Engine

This model is an IR based unsupervised model, we use traditional information retrieval model and add some rule based to retrieve the most probable answer.

Method We view each question as a query, sentences in the corresponding context as documents and the collection of those sentences as the corpus, so different problems will have different corpus and documents. For each question, we first specify the type of question and types of answers, then give the higher weight to the answer if it has the same type with the question.Second, we use BM25 to retrieve most similar sentences and compare those sentences with answer list to choose the answer.

$$score(D,Q) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|D|}{\mathrm{avgdoclen}})}$$

where $f(q_i, D)$ is query q_i's term frequency in the document D, $|D|$ is the length of the document D in words, and avgdoclen is the average document length in the corpus. k_1 and b are free parameters.

Inference We define that if the answer choice appear in the retrieved documents, then this answer choice is the truth positive. We use F1-score to evaluate the retrieved documents and choose the highest score as the answer.

2.4 Local Word Embedding

Method In this model, we use GloVe to load pre-trained word embeddings, and extract key words in a question, then training key words and choices word embedding in the context by fixing other words' word embedding. So words in different question might have different word embeddings .By training local word embeddings, we can catch the local meaning of those key words and choices.

Inference We use mean value of words' embedding in the question to represent the question, and calculate cosine similarity of choices and question, choose the highest score as the answer.

3 Experiment

In this section, we evaluate this model performing on the multi-choice question answering dataset SQuAD (Rajpurkar et al., 2016) and the TOEFL listening.

Dataset SQuAD is a machine comprehension dataset on a large set of Wikipedia articles, with more than 100,000 questions. The answer to each question is always a span in the context. Different from SQuAD dataset, the answer to each question is not a span in the context. Therefore, the TOEFL dataset is more difficult than SQuAD, since we might not find the answer to each question in the context. The evaluation of our prediction is just the error to the true answer, and calculate the accuracy over the test data.

Model detail We use pre-trained 100 dimension GloVe word embedding to be the word representation, then tokenize each word and replace with its token index in the training and testing data. In the training phrase, the optimizer is AdaDelta and the mini-batch is set to 20.

Prediction Since the optimize target of this model is not optimize for the choices, we need to use the prediction result to select a best choice. Here we calculate the F1 score of the prediction sequence with each answer in the choices. To calculate the F1 score of two sequence, we need to tokenize our sequence, use the tokenize prediction as the relevant words. Then, we The answer that have the highest F1 score will be selected to be our final choice.

Setting We have tried different experiment settings to test this model under different condition. In the following, we describe these setting in detail, including supervised learning method, IR base unsupervised learning, ensemble and also the improvement result.

Supervised BiDAF Supervised BiDAF is the multi-layer attention flow model, which model the interaction between question and context words. The final objective function is to model the start and end position of the true answer.

Ensembled supervised BiDAF We ensemble several supervised BiDAF models' result to improve our prediction. Each model predict a choice for each question, questions will choose the choice with the highest votes.

IR based approach In IR base approach, we build a search engine for each context and question pair. The query will be the input to our search engine and return the most similar sentence in the context. Then, we use the return sentence to compare with each answer using F1 score.

Local word embedding We also try to use unsupervised approach to make the prediction. First, we load the pre-train word embedding for whole data. For each answers in answer list and key words in question, we will train a local embedding for them based on the pre-train embedding. Finally, calculate the cosine similarity with the question embedding and the answers' embedding, and choose the most similar on as the prediction.

Table 1: The accuracy for different models

Model	SQuAD	TOEFL ($)
Single BiDAF	0.7041	0.3346
Ensembled BiDAF	0.7252	0.3688
IR based approach	0.6308	None
Local word embedding	0.4323	None

Improvement To improve our prediction result, we try to use a weighted way to ensemble our models. We found that we can use the F1 score of each prediction and answer pair to be the weight. To our surprise, the weighted ensemble approach greatly improve the performance on both tasks. The following table shows the improvement result.

Table 2: The accuracy for different models

Model	SQuAD	TOEFL ($)
Ensembled BiDAF	0.7252	0.3688
Weighted ensembled BiDAF	0.7886	0.4180

4 What we learned

This is the first time that we try to use multi-layer model and use the last layer's error to back-propagate through each layer to update the parameters in different models. We have learned how to combine different models to extract the features and information to achieve out final goal. Since multi-layer model is too difficult to only learn it in a theoretical way, this implementation help us learn how to construct a complicate model in practical.

References

[1] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi & Hananneh Hajishirzi, *BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION*, University of Washington, 2016.

[2] Poonam Gupta, Vishal Gupta, *A Survey of Text Question Answering Techniques*, Panjab University, Chandigarh.