A QA System Based on Bidirectional LSTM with Text Similarity Calculation Model

Wenhua Xu, Hao Huang, Hao Gu, Jie Zhang, and Guan Gui(✉)

College of Telecommunication and Information Engineering,
Nanjing University of Posts and Telecommunications, Nanjing 210003, China
guiguan@njupt.edu.cn

Abstract. The development of deep learning in recent years has led to the development of natural language processing [1]. Question answering (QA) system is an important branch of natural language processing. It benefits from the application of neural networks and therefore its performance is constantly improving. The application of recurrent neural networks (RNN) and long short-term memory (LSTM) networks are more common in natural language processing. Inspired by the work of machine translation, this paper built an intelligent QA system based on the specific areas of the extension service. After analyzing the shortcomings of the RNN and the advantages of the LSTM network, we choose the bidirectional LSTM. In order to improve the performance, this paper add text similarity calculation in the QA system. At the end of the experiment, the convergence of the system and the accuracy of the answer to the question showed that the performance of the system is good.

Keywords: QA system · Deep learning · RNN · LSTM

1 Introduction

With the development of deep learning in recent years, the field of natural language processing has also developed rapidly [2]. QA system is a very popular research direction in the field of natural language processing. People can submit problems expressed in natural language to the QA system, and the system will return compact and accurate answers instead of just returning a collection of pages like a search engine. In other words, the QA system saves resources with maximum efficiency to find the answers most needed by users. The history of intelligent QA system can be traced back to the beginning of artificial intelligence (AI). Alan M. Turing, father of artificial intelligence, proposed an imitating game at the beginning of the book [3], which can be considered as the beginning of QA system. Turing test showed that a computer was intelligent if it can communicate in natural language like humans. Therefore, the field of natural language processing was popular in the world. This prompted a large number of researchers to explore language techniques by studying QA system.

In recent years, researchers have applied the sequence-to-sequence model in machine translation [4]. The model is further optimized in [5], and the authors used the LSTM model to obtain better performance than RNN model. Some scholars have used TFIDF to design a question and answer system [6].

S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 340–347, 2019.
https://doi.org/10.1007/978-3-030-19086-6_38

This paper proposed a QA system that combines the bidirectional LSTM model with a text similarity model. There are search systems and generative systems; this paper combines these two systems together. After training, the model's loss and the results of the test have achieved good performance.

The structure of the paper is as follows. Section 2 introduces some model structures and our model in the QA system. Section 3 shows the structure of the QA system. Section 4 reports our results include the system's convergence and partial results of question-answer testing compared with a single Bidirectional LSTM model.

2 Model Structure

2.1 Encoder-Decoder Model

In most cases, the encoder-decoder model is used to process natural language problems [4]. One of the most significant features of the encoder-decoder framework is that it is an end-to-end learning algorithm, which is a model for sequence-to-sequence problems. Briefly, it is just an input sequence $\mathbf{x} = [x_0, x_1 \cdots x_N]^T \in R^N$, to generate another output sequence $\mathbf{y} = [y_0, y_1 \cdots y_N]^T \in R^N$. Sequence-to-sequence model has many applications, such as translation, document harvesting, QA system, and so on [2] (Fig. 1).

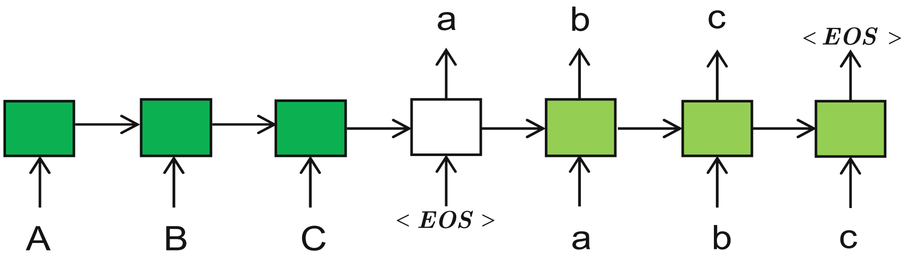


Fig. 1. An encoder-decoder model. This is the process of encoding the input and then decoding it. The encoder is to convert the input sequence into a vector of fixed length; decoder is to convert the previously generated fixed vector into an output sequence. The input is "A, B, C", and after encoding and decoding output "a, b, c".

2.2 RNN Encoder-Decoder Model

Because of the inconsistency of the input and output, it is difficult to separate these different sequences into separate samples for training, but RNN can deal with this problem. The input sequence is encoded using a recursive neural network (RNN) and a variable length sequence output is generated using another set of decoder RNN [7]. Then sent it to the network to training, this architecture has been proven perform better than the traditional phrase-based models (Fig. 2).

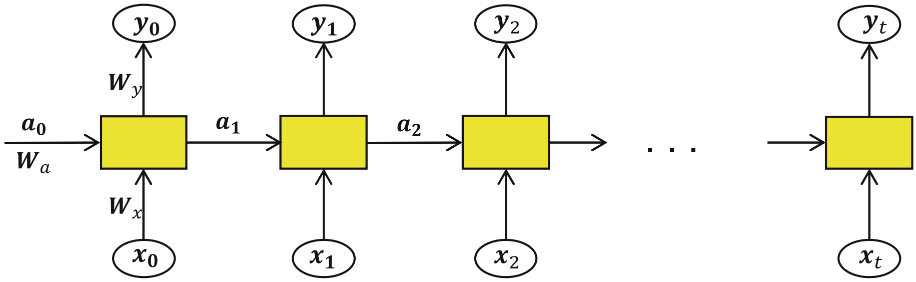


Fig. 2. A typical RNN model, where a_0 is an artificially fabricated activation value, usually a zero vector.

$$\boldsymbol{a}_t = g\left[\boldsymbol{W_a}(\boldsymbol{a}_{t-1}, \boldsymbol{x}_t)^T + \boldsymbol{b_a}\right] \tag{1}$$

$$\boldsymbol{y}_t = g\left(\boldsymbol{W_y}\boldsymbol{a}_t + \boldsymbol{b_y}\right] \tag{2}$$

$$\boldsymbol{a}_t = [a_{t0}, a_{t1}, \ldots a_{tk}]^T \tag{3}$$

$$\boldsymbol{W_y} = \left[W_{y0}, W_{y1}, \ldots W_{yk}\right]^T \tag{4}$$

$$\boldsymbol{W_x} = [W_{x0}, W_{x1}, \ldots W_{xk}]^T \tag{5}$$

$$\boldsymbol{W_a} = [W_{a0}, W_{a1}, \ldots W_{ak}]^T \tag{6}$$

Here, '$\boldsymbol{x_t}$', '$\boldsymbol{y_t}$' and '$\boldsymbol{a}_t$' denote the input, output and the initial value in the t-th moment, '$\boldsymbol{W}$' and '$\boldsymbol{b}$' denote the weight and bias. g denotes the activation function.

Take the weight W_0 update of as an example. $\boldsymbol{L}$ is the loss function. We use the cross-loss entropy. According to the chain derivation rule, the weight update formula is:

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}_0} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W_t}} \cdot \frac{\partial \boldsymbol{W_t}}{\partial \boldsymbol{W_{t-1}}} \cdots\cdots \frac{\partial \boldsymbol{W}_1}{\partial \boldsymbol{W}_0} \tag{7}$$

It can be seen from the formula, that if the gradient is bigger than 1, the gradient will exponentially increase with the number of iterations; if the gradient value is smaller than 1, with the increase in the number of network layers, the gradient will gradually disappear, and the RNN's memory will fade slowly [8]. This is the problem of the disappearance of the RNN gradient.

2.3 Bidirectional LSTM Encoder-Decoder Model

Because of the gradient disappearance, the RNN cannot achieve the real memory characteristic when address long sequence [9]. If we make the gradient is equal to 1 at all the time that the gradient disappearance will be solved. Therefore, we should make a constraint to ensure that the gradient value is equal to 1 all the time. The LSTM model made improvements to RNN and solved the problems of gradient disappearance and gradient explosion [10].

In Fig. 3, '$\boldsymbol{f}_t$', '$\boldsymbol{i}_t$' and '$\boldsymbol{o}_t$' denote the output of forget gate, output of input gate and output of output gate. '$\widetilde{\boldsymbol{c}(t)}$' denotes the intermediate variable in the t-th moment and σ denotes the sigmoid function. '$\boldsymbol{c}_t$' denotes the input in the t-th moment.

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W_f} \cdot [\boldsymbol{y}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b_f}\right) \tag{8}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W_i} \cdot [\boldsymbol{y}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b_i}) \tag{9}$$

$$\widetilde{\boldsymbol{c}(t)} = \tanh(\boldsymbol{W} \cdot [\boldsymbol{y}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b_C}) \tag{10}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W_o} \cdot [\boldsymbol{y}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{11}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \cdot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \cdot \widetilde{\boldsymbol{c}(t)} \tag{12}$$

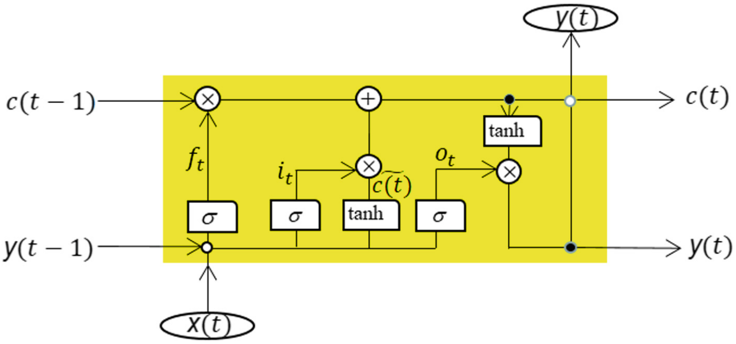


Fig. 3. LSTM encoder-decoder Model.

In order to make the system performance better, this paper used bidirectional LSTM network. Compared with unidirectional networks, bidirectional LSTM network can remember more information [12].

In Fig. 4, bidirectional LSTM is superimposed by traditional LSTM and performs better than traditional LSTM networks. The bidirectional LSTM consists of LSTM in both directions. The forward LSTM network can remember the information in the previous sequence, and the reverse can remember the information behind.

2.4 Text Similarity Calculation

A text similarity calculation is added to this QA system. TF-IDF is used to calculate the word frequency set of words in text, and combines them with vectors to calculate the similarity by comparing the cosine distances between different sets of vectors in linear space.

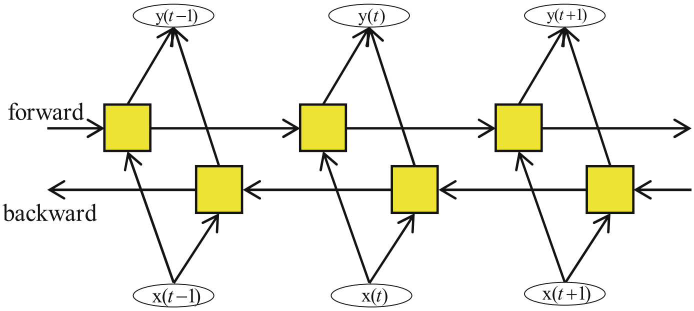


Fig. 4. A bidirectional LSTM network

This is mainly used to calculate the similarity between the input question and the question in the training set. If they are very similar, the answer to the input question is the answer to the stationery in the training set.

3 Structure of QA System

The structure of the QA system is shown in Fig. 5. Compared with the general seq2seq model, this model added a text similarity algorithm. The model can be roughly divided into two parts.

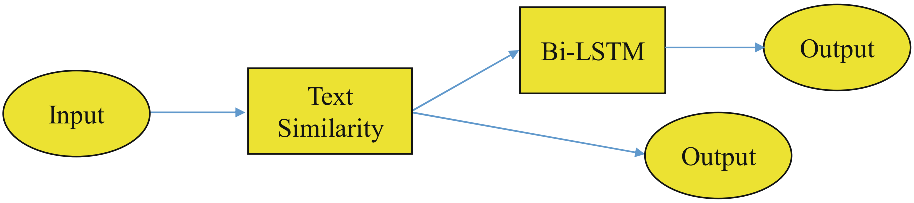


Fig. 5. The structure of the QA system

First, the input sentence is matched with the sentence in the training set by the text similarity algorithm. If the similarity with the corpus's question is extremely high, the corpus answer is output directly. If not very similar, it will enter the Bidirectional LSTM model for training to get the final output.

For the similarity evaluation, we set a threshold. The value of this threshold is set to 0.8 based on a large number of experiments. Only when the similarity between the input and the question in the corpus is higher than this threshold, the answer to the question in the corpus is output. If it is below this threshold, the final output will be obtained through the training results of Bidirectional LSTM.

4 Experiment and Discussions

4.1 Dataset

In order to complete this specific QA system, we created a new dataset which is closely related to our mission. By manual tagging, the questions and answers of the corpus are separated into two documents. The 12 categories questions and answers are mainly selected, each of which contains about 1000 answers and 1000 questions. The goal of training is to make reasonable and accurate answers to certain questions.

4.2 Simulation Results

Figure 6 displays the training process of the model, and it describes the continuous reduction of loss as the number of iterations increases. It can be obviously seen that loss shows a downward trend and eventually stabilizes. The moment loss tend to be stabilize when model tend to converge. The simulation result shows that after about 2000 iterations, model is convergent enough to 0.11. This showed that the bidirectional LSTM could be used to build the QA system.

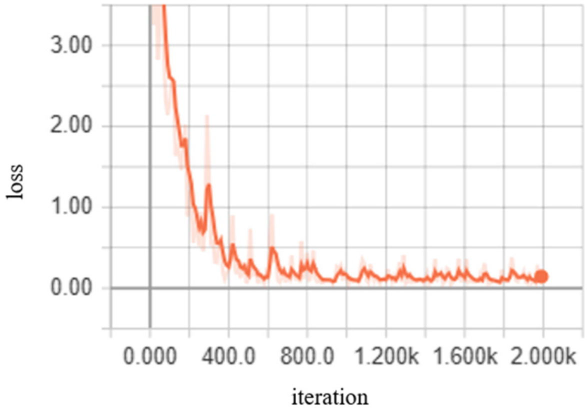


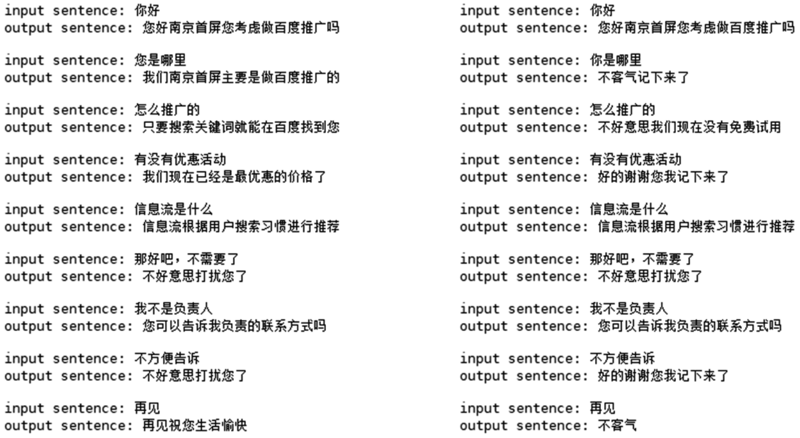
Fig. 6. Performance curve for Bidirectional LSTM model from tensor board.

4.3 Performance Comparison with Other Model

Figure 7 shows that under the same training set, both models can complete the task. It is obviously that both two models can give answers when asked. But after carefully comparing the results, we can find that the model with text similarity calculation gives the better answers which are more humanize. As far as the final display results are concerned, the effect of the model with text similarity calculation did exceed the single effect of using only the bidirectional LSTM network in some questions.

4.4 Performance Analysis

It can be seen from the Fig. 7 that adding text similarity calculation to the model can improve the performance of the QA system.



```
input sentence: 你好
output sentence: 您好南京首屏您考虑做百度推广吗

input sentence: 您是哪里
output sentence: 我们南京首屏主要是做百度推广的

input sentence: 怎么推广的
output sentence: 只要搜索关键词就能在百度找到您

input sentence: 有没有优惠活动
output sentence: 我们现在已经是最优惠的价格了

input sentence: 信息流是什么
output sentence: 信息流根据用户搜索习惯进行推荐

input sentence: 那好吧，不需要了
output sentence: 不好意思打扰您了

input sentence: 我不是负责人
output sentence: 您可以告诉我负责的联系方式吗

input sentence: 不方便告诉
output sentence: 不好意思打扰您了

input sentence: 再见
output sentence: 再见祝您生活愉快
```

```
input sentence: 你好
output sentence: 您好南京首屏您考虑做百度推广吗

input sentence: 你是哪里
output sentence: 不客气记下来了

input sentence: 怎么推广的
output sentence: 不好意思我们现在没有免费试用

input sentence: 有没有优惠活动
output sentence: 好的谢谢您我记下来了

input sentence: 信息流是什么
output sentence: 信息流根据用户搜索习惯进行推荐

input sentence: 那好吧，不需要了
output sentence: 不好意思打扰您了

input sentence: 我不是负责人
output sentence: 您可以告诉我负责的联系方式吗

input sentence: 不方便告诉
output sentence: 好的谢谢您我记下来了

input sentence: 再见
output sentence: 不客气
```

Fig. 7. This is a small part of the results. The picture on the left is the effect of adding text similarity calculation. The picture on the right is not.

When asked about greetings like ‘hello’, both models have the same answers because of the same training data and the input was short. However, when asked about other question as it shows in the fifth pair, they give the different answers and it is obviously the answer that model with text similarity calculation gives is better than that bidirectional LSTM gives.

As a result, by comparing the answers the models give, it is obviously that model with text similarity calculation has the better performance and it is easily accepted by people. Because the generated model is based on a large number of corpora, our corpus may not be enough. If the corpus is large enough, the model will work well. However, using text similarity model to make a QA system, as long as the input question can find a matching sentence in the corpus, it can output a relatively accurate answer. Conversely, for questions that are not in the corpus, they may not be handled well enough. In this case, we can use the generated model to get the answer. Combining the generated model with the text similarity model can find the most appropriate answer and build a relatively good performance QA system.

5 Conclusion

The work of this paper was mainly to obtain inspiration from machine translation and apply the model of machine translation to the QA system in the extension service field. After analyzing the memory defects of the RNN model from a mathematical view, we chose a bidirectional LSTM network and created a specific data set for this area. And this paper add a text similarity model in the QA system. According to the results of the final experiment, the QA system showed that model with text similarity calculation has the better performance. However, there is still much room for improvement in this QA system. Next, we may try to add attention mechanism to the model and looking for a better text similarity method.

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
2. Spyns, P.: Natural language processing in medicine: an overview. Methods Inf. Med. **35**(04), 285–301 (1996)
3. Turing, A.M.: Computing machinery and intelligence. In: Epstein, R., Roberts, G., Beber, G. (eds.) in Parsing the Turing Test, pp. 23–65. Springer, Dordrecht (2009). https://doi.org/10.1007/978-1-4020-6710-5_4
4. Van Merri, B.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014, pp. 1724–1734 (2014)
5. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Neural Information Processing Systems, Kuching, Malaysia, 3–6 November 2014, pp. 3104–3112 (2014)
6. Zhao, S.H., Li, J.-Y., Xu, B.-R., et al.: Improved TFIDF-based question similarity algorithm for the community interlocution systems. Trans. Beijing Inst. Technol. **37**(9), 982–985 (2017)
7. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015, pp. 1–6 (2015)
8. Tran, K.M., Bisazza, A., Monz, C.: Recurrent memory networks for language modeling. In: North American Chapter of the Association for Computational Linguistics, San Diego, CA, USA, 12–17 June 2016, pp. 321–331 (2016)
9. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78** (10), 1550–1560 (1990)
10. Bengio, Y., Simard, P.Y., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A Search Space Odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2222–2232 (2017)
13. Graves, A., Fernandez, S., Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks, Warsaw, Poland, 11–15 September 2005, pp. 799–804 (2005)