LSTM-Based Neural Network Model for Semantic Search

Xiaoyu Guo, Jing Ma and Xiaofeng Li

Abstract To improve web search quality and serve a better search experience for users, it is important to capture semantic information from user query which contains user's intention in web search. Long Short-Term Memory (LSTM), a significant network in deep learning has made tremendous achievements in capturing semantic information and predicting the semantic relatedness of two sentences. In this study, considering the similarity between predicting the relatedness of sentence pair task and semantic search, we provide a novel channel to process semantic search task: see semantic search as an atypical predicting the relatedness of sentence pair task. Furthermore, we propose an LSTM-Based Neural Network Model which is suitable for predicting the semantic relatedness between user query and potential documents. The proposed LSTM-Based Neural Network Model is trained by Home Depot dataset. Results show that our model outperforms than other models.

Keywords LSTM · Deep learning · Semantic search · RNN

1 Introduction

Having a better understanding of user's intention is very important to improve search engine's quality and optimize the user experience. Traditional search engine is mainly based on matching keywords in documents with search queries. However, this technology cannot distinguish synonymous words from different sentences. Moreover, users have to consider a lot about how to organize query words in order to get the right information they want. This brings too much inconvenience to users. As a result, semantic search engine emerges in order to better serve users.

X. Guo · J. Ma (✉) · X. Li
College of Economics and Management, Nanjing University of Aeronautics and Astronautics,
Nanjing, Jiangning District 211106, China
e-mail: majing5525@126.com

H. Yang et al. (eds.), *Smart Service Systems, Operations Management,*
and Analytics, Springer Proceedings in Business and Economics,
https://doi.org/10.1007/978-3-030-30967-1_3

Recently, Long short-term memory (LSTM) networks have shown remarkable performance in several kinds of Natural Language Processing (NLP) tasks, including image captioning [1], machine translation [2] and semantic search [3]. LSTM networks have also been used to predict semantic relatedness score [4] in order to find the relevant questions from the existing questions and find the relevant answers to a new question for Community Question Answering forums. Inspired by this task, we propose an LSTM-Based Neural Network Model reformed from predicting semantic relatedness score model based on similarities between predicting the relatedness of sentence pair and semantic search. We perform experiments to validate the utility of our model in Home Depot dataset, and compare the method to other models.

The contributions of this paper are as follows:

(1) Explore a novel method to process semantic search task. In this field, user query contains less information than search result does, so this paper proposed to split search result document and splice user query and related documents many times to balance the information quantity.
(2) Based on the method mentioned in (1), this paper built a novel neural network model for semantic search field.
(3) Perform experiments to validate the utility of our model, and compare our model to other model using different variants of LSTM architecture.

2 Related Work

The traditional language model, without taking sequence factor into account, cannot capture semantic information thoroughly. For example, “look after” and “after looking” have different meaning and traditional language model cannot tell any differences from each other. Mikolov et al. [5] proposed Recurrent Neural Network Language Model (RNNLM) in order to process sequence data. Recurrent neural network (RNN), which is different from normal neural network, introduces constant circulation into its model so that it could process sequence information. RNN shows remarkable performance in processing many tasks concerning sequence, but RNN has a Long-Term Dependencies problem [6] when it processes longer passages that contain too much information. The LSTM architecture, proposed by Hochreiter and Schmidhuber [7], addresses this problem of Long-Term Dependencies by introducing a memory cell that is able to store state information over long period of time into RNN structure. LSTM has recently been used in information retrieval field for extracting sentence-level semantic vectors [8] and context-aware query suggestion [9].

Commonly used variants of LSTM architecture are the Bidirectional Long Short-Term Memory (BLSTM) Network and the Multilayer Long Short-Term Memory (MLSTM). Several scholars pay much attention to exploring novel LSTM variants, including Tree-Structured Long Short-Term Memory Networks proposed by

Tai et al. [10] and multiplicative LSTM proposed by Stephen Merity et al. [11] in order to obtain a better performance on NLP tasks. In this paper, we conducted our experiments using different variants of LSTM.

The setup of our work is closely related to what was proposed already by Nassif H et al. [4]. Nassif et al. aim to obtain the semantic relatedness score between two questions as shown in Fig. 1. The model was built on two bidirectional LSTM whose output can be augmented with extra features and fed into the multilayer perceptron. Other similar methods include the combination of recurrent and convolutional models [12]. In this paper, we regard semantic search as predicting semantic relatedness between user query and potential document aiming to find the closest result in semantic meaning. The main difference of our method compared to these models is that we balance the quantity of information between inputs by exploiting user query several times. We also compare our method to these methods.

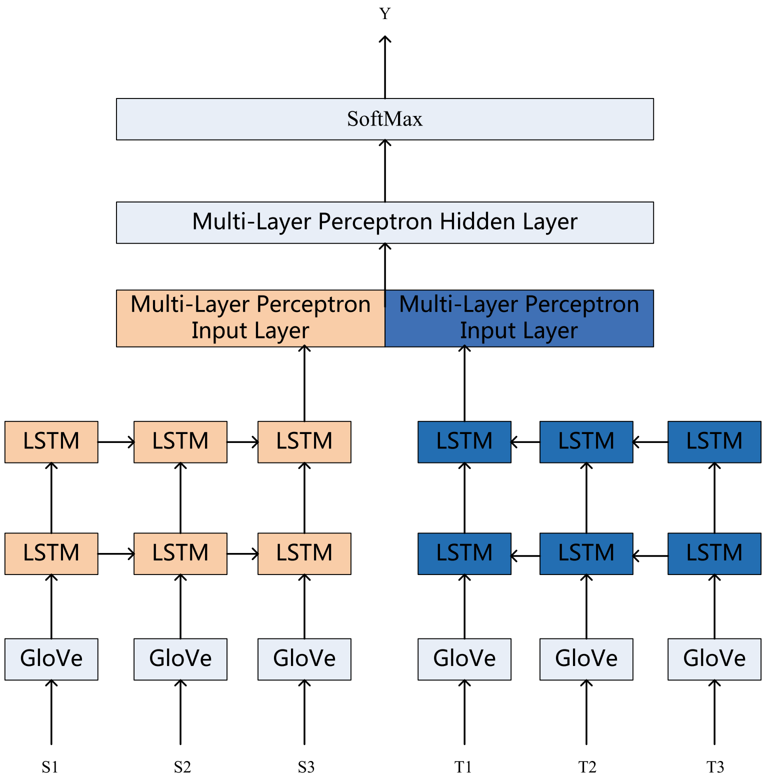
Y
SoftMax
Multi-Layer Perceptron Hidden Layer
Multi-Layer Perceptron Input Layer
Multi-Layer Perceptron Input Layer
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
GloVe
GloVe
GloVe
GloVe
GloVe
GloVe
S1
S2
S3
T1
T2
T3

Fig. 1 The general architecture of predicting semantic similarity score model

3 Method

The essence of search engine is to calculate the relatedness score between user query and possible documents. Hence, predicting the semantic similarity bears strong resemblances to semantic search. We could see a semantic search task as described below:

$$f(Query, Doc) = SemanticScore\{a|a \in R, a \in [1, 3]\} \tag{1}$$

where Query is user input query words, Doc is a document and SemanticScore is a real number between 1 and 3.

That is to say, given user query and documents, processed by functional transformation and then output a real number between 1 and 3, in which 1 denotes "not relevant" and 3 denotes "extremely relevant".

Based on the model mentioned in 2, we proposed a LSTM-Based Neural Network Model which is suitable for semantic search, as is shown in Fig. 3. If we directly apply the model mentioned in 2 into semantic search, we would see a model shown in Fig. 2. Apparently, there exists an imbalance in information quantity between user queries and documents. That is to say, documents contain more information than user queries do. Therefore, if we directly use model in Fig. 2, we could not gain remarkable performance theoretically.

To balance the information quantity, we reform the model in Fig. 2 and the model in Fig. 3 is the general architecture of our model. There are three differences between Figs. 2 and 3. (1) We splice user query and related documents many times. The input layer of our model consists of two parts. One is LSTM representation of user query and another is LSTM representation of sentences in related documents. In this way, we make the best use of user query so that we balance the information quantity. (2) For every pair of user query and a document, we gain several semantic scores. (3) We use Ridge Regression as the last step to output the final semantic relatedness score.

The input of our model consists of two parts: one is user query words (QW) and another is Related Document. For QW, the model directly transforms them to vectors using GloVe and then encodes vectors using LSTM. For Related Document, the model splits the documents into sentences first and then processes them using the same methods as QW. After LSTM encoding, the model splice user query to every sentence twice. Then we get several semantic scores and our goal is to gain a final semantic score. Therefore, we see this task as a regression problem:

$$\begin{aligned}\widehat{\mathrm{S}} &= Xw \\ &= X(X^T X + \lambda E)^{-1} X^T S\end{aligned}$$

where S denotes actual semantic scores, $\widehat{\mathrm{S}}$ denotes the final forecasting semantic score, and X denotes a matrix consisted of sub-semantic scores.

In the last step, this model uses Ridge Regression to predict the final score. The output of our model is a final semantic score representing the semantic relatedness

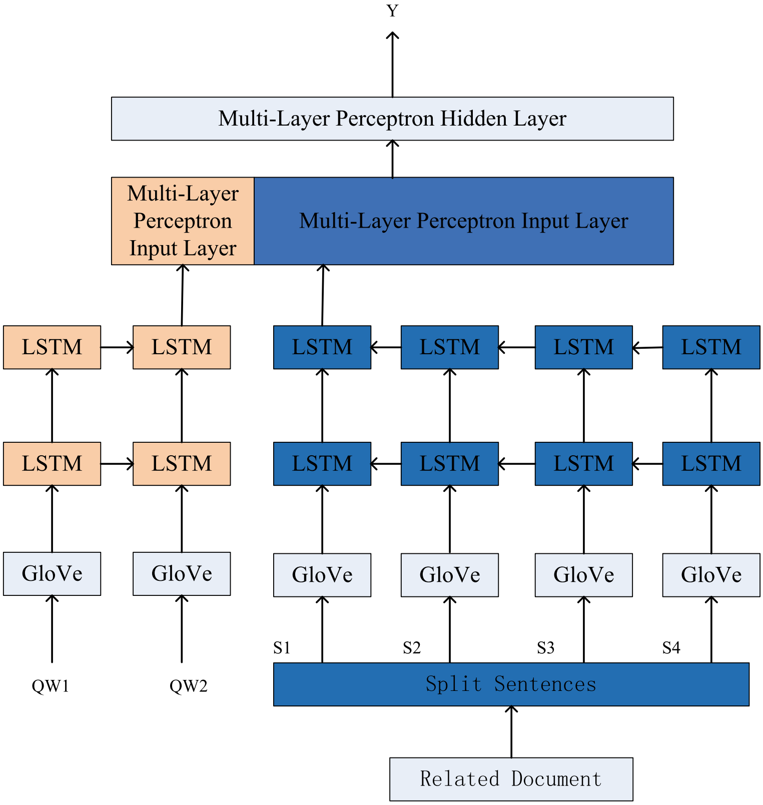

Fig. 2 Directly apply predicting semantic score model into semantic search

between user query and document. This model is designed to process English semantic search task. If you want to apply this model to other languages, all you have to do is to change the input part of the model. Take Chinese into example, we can split words first and then uses this model.

4 Experimental Setup and Results

In this section, we describe our experimental setup. The dataset and Evaluation methodology is described in Sects. 4.1 and 4.2. The experiments that we conduct to test the performance of our model is given in Sect.4.3.

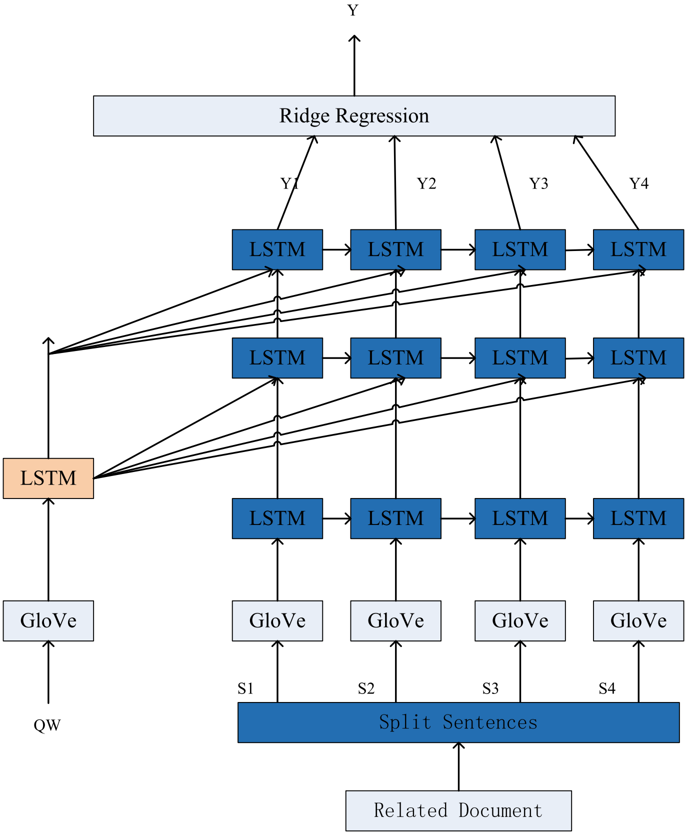

Fig. 3 LSTM-based neural network for semantic search

4.1 Dataset

In this paper, the dataset [13] we used to test the performance is from The Home Depot, an e-commerce website in America. This platform sells building materials. As a user who wants to buy products from this website, all you have to do is to type your objective and then click “search” button. For example, type “I want to lay the foundation” in input field and then click “search”. The results will show tools for laying the foundation. Therefore, this dataset is suitable for semantic tasks.

This dataset contains 124429 query–document pairs and 37034 unique product description documents. Each query–document pair contains a human-generated relevance label, which is the average of 5 ratings assigned by different human annotators. We split the query sessions into three parts, and use 60% to train LSTM-Based Neural Network Models, 10% to trial models (aim to choose a best one then apply it to the test dataset) and 30% as test data to evaluate the prediction performance.

4.2 Evaluation Methodology

We use root mean squared error (RMSE) as evaluation metrics. Actually, RMSE score indicates the gap between prediction and real value. Therefore, we aim to gain a smaller RMSE value.

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2} \tag{3}$$

4.3 Experiments

For a given pair of user query and document, our task is to predict a human-generated rating of the similarity of user query and document in meaning.

Here, we use the similarity model described in Fig. 3. We initialized our word representations using publicly available 300-dimensional Glove vectors [14]. We trained our models with a learning rate of 0.05 and a minibatch size of 25, mentioned by AdaGrad [15].

The last step is Ridge Regression [16], in which we predict the final relatedness score using sub-relatedness scores. In this step, we choose a proper Alpha value to get the best result. Figure 4 shows the relationship between Alpha and Cross-validation (CV) value of different LSTM variants.

We compare our model with predicting semantic similarity model using the same dataset and the results are summarized in Table 1. The first group is the performances of predicting semantic similarity model and the second group is the performances

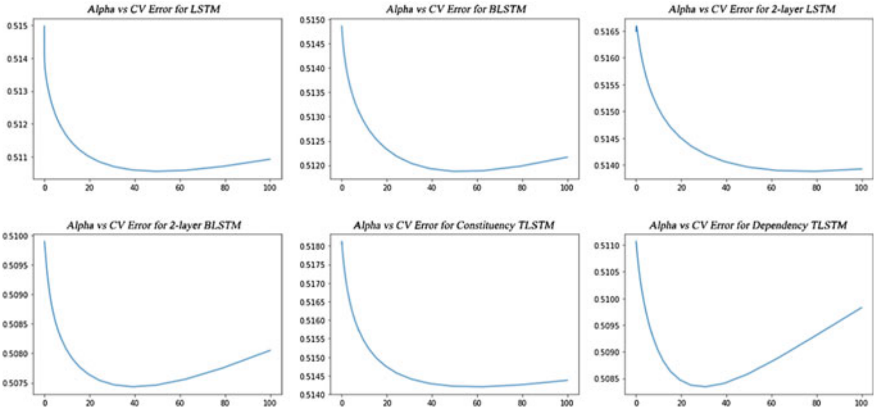

Fig. 4 The relationship between alpha and Cross-Validation (CV) value of different LSTM variants

Table 1 Test set results on the home depot dataset

Method		RMSE
Predicting semantic similarity model (Henry Nassif et al.,2016)	LSTM	0.6664
	BLSTM	**0.6747**
	2-layer LSTM	0.6830
	2-layer BLSTM	0.6608
LSTM-based neural network for semantic search	LSTM	0.4682
	BLSTM	**0.4611**
	2-layer LSTM	0.4643
	2-layer BLSTM	0.4668
	Constituency TLSTM	0.4646
	Dependency TLSTM	0.4645

of LSTM-Based Neural Network for Semantic Search. Overall, the performances of LSTM-Based Neural Network is better than predicting semantic similarity model's, with approximately 0.2 improved. More specifically, BLSTM shows best performance among all the LSTM variants on both two models, while there are slight differences within each group.

5 Conclusion and Future Work

In this paper, we propose an LSTM-Based Neural Network model, which is reformed from an existent predicting relatedness score task model. We see the semantic search problem as an atypical predicting relatedness score task. The results show that our model has a remarkable performance in predicting semantic relatedness score between user query and potential documents without adding any additional man-made feature engineering. Therefore, our model saves time to make feature engineering and reduce the influence of human factors for sorting search results.

We may foresee that it will serve a better search experience for users if we use our model in semantic search field because there is no need to think too much about search words. For example, people who have little knowledge of search engine principle may consider too much about which word should be included and which word should not be included in query because the extra word may have a bad influence on search results. Therefore, our model may save too much time that users used to consider constructing query and provide a better search service for users.

In the future, we would pay much attention to the output period. In this paper, we just use Ridge Regression to output our semantic relatedness score. But Ensemble learning is famous for its performance in machine learning field. Consequently, we would try to combine Ridge Regression with other machine learning methods to output the final result in future work. In addition, we just use one English dataset to test model because it is hard to obtain semantic search-related dataset. Therefore, we will dedicate to obtaining different datasets and testing our model.

References

1. O. Vinyals, A. Toshev, S. Bengio et al., *Show and Tell: A Neural Image Caption Generator [J]* (2014)
2. I. Sutskever, O. Vinyals, Q.V. Le, *Sequence to Sequence Learning with Neural Networks [J]* (2014)
3. P.S. Huang, X. He, J. Gao et al., Learning deep structured semantic models for web search using clickthrough data [C], in *Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management* (ACM, 2013)
4. H. Nassif, M. Mohtarami, J. Glass, Learning semantic relatedness in community question answering using neural models [C], in *Proceedings of the 1st Workshop on Representation Learning for NLP* (2016), pp. 137–147
5. T. Mikolov, *Statistical Language Models Based on Neural Networks [J]* (Presentation at Google, Mountain View, 2012), p. 80
6. R. Pascanu, T Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks [C], in *International Conference on Machine Learning* (2013), pp. 1310–1318
7. S. Hochreiter, J. Schmidhuber, Long short-term memory[J]. Neural Comput. **9**(8), 1735–1780 (1997)
8. H. Palangi, L. Deng, Y. Shen et al., Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval [J]. IEEE/ACM Trans. Audio Speech Lang. Process (TASLP) **24**(4), 694–707 (2016)
9. A. Sordoni, Y. Bengio, H. Vahabi et al., A hierarchical recurrent encoder-decoder for generative context-aware query suggestion[C], in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (ACM, 2015), pp. 553–562
10. K.S. Tai, R. Socher, C.D. Manning, *Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks [J]* (2015). arXiv:1503.00075, 2015
11. B. Krause, L. Lu, I. Murray et al., *Multiplicative LSTM for Sequence Modelling [J]* (2016) arXiv:1609.07959
12. H.S. Joshi, *Finding Similar Questions in Large-scale Community QA Forums [D]*. Massachusetts Institute of Technology (2016)
13. https://www.kaggle.com/c/home-depot-product-search-relevance/data
14. J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation [C], in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543
15. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization[J]. J. Mach. Learn. Res. 2121–2159 (2011)
16. A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems [J]. Technometrics **12**(1), 55–67 (1970)