Research on semantic representation based on syntactic dependency structure and type

Qiao Kang [#1,*1], Can Zeng [#1], Pengfei Hu [#1], Fangyan Dong [#1],Kewei Chen[#1,*2]

#1 Faculty of Mechanical Engineering & Mechanics, Ningbo University, Ningbo, 315211,China
*1 E-mail: 2011081055@nbu.edu.cn
*2 E-mail: chenkewei@nbu.edu.cn

Abstract: Long short-term memory (LSTM) has become the state-of-the-art for modeling semantic similarity computation. However, it cannot fully encode hierarchical syntactic dependencies and cannot identify differences in meaning caused by changes in the semantic roles of words or phrases. Therefore, a new relative syntactic distance dependency type long short-term memory (RSD-DT-LSTM) neural model is proposed in this paper. This model modifies the gating mechanism in Dependency type long short-term memory (DT-LSTM) with relative syntactic distance and provides a transformed gate to more directly model syntactic structure. Make sentence structure dependencies and dependency types embed sentence meanings into dense vectors. The RSD-DT-LSTM model was used to perform semantic similarity scoring experiments on the SICK dataset and the MRPC dataset. The Pearson correlation coefficient obtained by the model on the sick data set is very close to the DT-LSTM model. The mean square error (MSE) between the actual similarity score y and the predicted score y ^ 'is 0.1% better than the DT-LSTM model. In the MRPC dataset, its accuracy is 0.9% better than that of DT-LSTM. And its recall is 0.3% better than that of DT-LSTM model.

Keywords: semantic similarity; LSTM; relative syntactic distance

1. INTRODUCTION

Computing sentence semantic similarity[1] is an important fundamental problem in natural language processing. This technology is widely used in information retrieval, question answering systems, and machine translation. In information retrieval, semantic similarity can calculate the match between a query term and a document in a database. In question answering systems, semantic similarity can calculate the degree of matching between query questions and candidate answers. The basic technology of calculating the semantic similarity of sentences directly affects the performance of the upper-layer application system. Therefore, researching and promoting the development of sentence semantic similarity calculation is the top priority[2,3].

At present, the calculation methods of semantic similarity are mainly based on word overlap, corpus statistics, linguistics, and neural network[4]. The method based on word overlap is to calculate the similarity of sentences by some words shared by a sentence pair. Among them, the Jaccard similarity coefficient method calculates the similarity of sentences by calculating the ratio of the intersection of words and the union of words in two sentences[5]. The method based on corpus statistics is to take the set of words appearing in the sentence pair as the feature set. The similarity is expressed as the cosine of the angle between the corpus-based vectors[6]. Jie Shi calculated the semantic similarity between sentences through the overlapping information of the semantic hierarchical tree structure in the WordNet dictionary[7]. Xin Li calculated the semantic similarity of sentence pairs by constructing a text relatedness subspace and a text sentiment subspace[8]. Linguistics-based methods use the semantic relationship and grammatical components between words to determine the similarity of sentences[9]. Xu Li normalized the sum of the similarity between words and words in the sentence pair as the similarity value of the sentence pair[10].

Most of the deep learning models currently used for sentence semantic matching tasks use recurrent neural networks (RNN) or convolutional neural networks (CNN). Recurrent neural networks have the ability to process sequences of arbitrary length. Long short-term memory (LSTM) can solve the problem of long-term dependence. But these models fail to capture non-linear dependencies between words. Dependency type long short-term memory (DT-LSTM) model was proposed by Jeena Kleenankandy. It can use the control input to model the relationship between the two inputs of the sequence. The model uses dependency types to obtain high-quality semantic representations of sentences[11]. Yikang Shen proposed parse-read-prediction network (PRPN). This model introduces the concept of syntactic distance. It can utilize syntactic structure information to form better semantic representation and modeling[12]. But it cannot comprehensively encode hierarchical syntactic dependencies, and cannot identify differences in meaning

caused by changes in the semantic roles of words or phrases.

A novel model of RSD-DT-LSTM is proposed. The model exploits sentence structure dependencies and dependency types. Embed sentence meaning into dense vectors to obtain more comprehensive and accurate sentence semantic similarity.

2. DT-LSTM MODEL

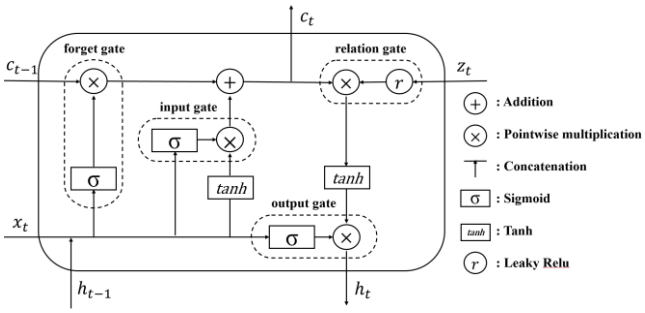


Fig.1 DT-LSTM model structure diagram

The dependency type long short-term memory (DT-LSTM) is an enhanced version of the LSTM model. It consists of four gates (forget gate, input gate, output gate, relation gate) and two memory states (unit state and hidden state). For each time t, the model structure of DT-LSTM is shown in Fig.1.

Each DT-LSTM$^{(t)}$ cell in the model corresponds to a node t. DT-LSTM$^{(t)}$ in x_t as input, the word vector is represented by w_t and the relation vector is $z_t = e^{j}$ (binary vector with 1 at dimension j and 0 elsewhere). The output of DT-LSTM$^{(t)}$ is unit state c_t and hidden state h_t. They propagate to their parent DT-LSTM$^{(t)}$.

The definition of h_k is previously retained memory:

$$h_{C(t)} = \Sigma_{k \in C(t)} h_k \tag{1}$$

Here, $C(t)$ represents node t is a collection of all child nodes, $k \in C(t)$.

The definition of f_{tk} is updated memory of forgotten gate:

$$f_{tk} = \sigma(W^{(f)} x_t + U^{(f)} h_k + b^{(i)}) \tag{2}$$

Here, x_t represents the vector representation of the word. i_t represents updated memory:

$$i_t = \sigma(W^{(i)} x_t + U^{(i)} h_{c(t)} + b^{(i)}) \tag{3}$$

The definition of u_t is the data distribution after remodeling:

$$u_t = tanh(W^{(u)} x_t + U^{(u)} h_{c(t)} + b^{(u)}) \tag{4}$$

The definition of c_t is current latest memory:

$$c_t = i_t \odot u_t + \Sigma_{k \in C(t)} f_{tk} \odot c_k \tag{5}$$

The definition of r_t is dependency type memory:

$$r_t = ReLU(W^{(r)} z_t + b^{(r)}) \tag{6}$$

Here, z_t represents input dependency type vector.

The definition of o_t is output probability vector:

$$o_t = \sigma(W^{(o)} x_t + U^{(o)} h_{c(t)} + b^{(o)}) \tag{7}$$

The definition of h_t is output vector:

$$h_t = o_t \odot tanh\,(c_t \odot r_t) \tag{8}$$

Relational gates are especially important in semantic tasks. Its words can share different types of dependencies. And these dependency types affect the semantic composition between words. The DT-LSTM model models text sequentially. Therefore, hierarchical syntactic dependencies cannot be fully encoded.

3. RSD-DT-LSTM MODEL

This section introduces proposed relative syntactic distance dependency type long short-term memory (RSD-DT-LSTM) model.

3.1. Parsing the relative syntactic distance

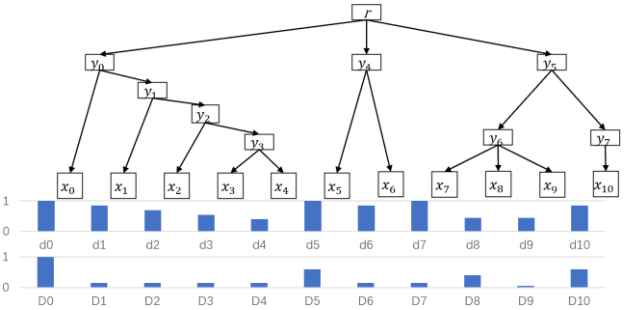


Fig.2 Syntax tree structure diagram

In syntax trees, syntactic distance is defined as the height of the lowest common ancestor between adjacent words in the syntax tree. Syntactic distance can model the syntactic relationship between words in a sentence. It is calculated by convolutional layer and fully connected layers, v_t is represented as input word vector.

The definition of z_t is updated memory:

$$z_t = ReLU\left(W_c \begin{bmatrix} v_{t-L} \\ \vdots \\ v_t \end{bmatrix} + b_c\right) \tag{9}$$

d_t is defined as syntactic distance between x_{t-1} and x_t:

$$d_t = sigmoid(W_d z_t + b_d) \tag{10}$$

Fig. 2 is a syntactic tree structure diagram. x_t represents the input words, d_t represents the syntactic distance between words, and D_t represents the relative syntactic distance between words. The syntactic distance d_0 of x_0 is the height of y_0, and the syntactic distance d_1 of x_1 is the height of y_1.

The larger the syntactic distance represents the smaller the syntactic dependence between the two words. For example, the dependency path of x_4 is r→y_0→y_1→y_2→y_3, and the dependency path of x_5 is r→y_4. The dependency paths of x_4 and x_5 are completely different. Therefore its syntactic distance is close to 1. So are x_6 and x_7. The dependency path of x_0 is r→y_0, and the dependency path of x_1 is r→y_0→y_1. x_0 and x_1 have very similar dependency paths. There is a strong dependence between them. But due to the large height of y_1, the syntactic distance between x_0 and x_1 is large. Therefore, the use of syntactic distance alone cannot well represent the degree of dependence between words. Relative syntactic distance is used.

The definition of D_t is the relative syntactic distance:

$$D_t = min(d_t,\ |d_t - d_{t-1}|) \tag{11}$$

3.2. Gating mechanism

In ordinary RNNs, multiple multiplications of gradients during backpropagation can cause information to disappear over many time steps. Gating mechanisms are used to control the information paths passed in neural networks. LSTMs use input and forget gates to achieve long-term memory. This model modifies the gating mechanism so that syntactic information can be efficiently updated between words.

The definition of h_k is previously retained memory:

$$h_{C(t)} = \Sigma_{k\in C(t)}\widetilde{h_k} \tag{12}$$

Here, $C(t)$ represents node t is a collection of all child nodes, $k \in C(t)$.

The definition of f_{tk} is updated memory of forgotten gate:

$$f_{tk} = \sigma((W^{(f)}x_t + U^{(f)}h_k + b^{(i)}) \cdot (2D_t)) \tag{13}$$

Here, x_t represents the vector representation of the word. D_t represents relative syntactic distance.

The definition of i_t is updated memory:

$$i_t = \sigma((W^{(i)}x_t + U^{(i)}h_{c(t)} + b^{(i)}) \cdot (2 - 2D_t)) \tag{14}$$

The definition of u_t is the data distribution after remodeling:

$$u_t = tanh(W^{(u)}x_t + U^{(u)}h_{c(t)} + b^{(u)}) \tag{15}$$

The definition of c_t is current latest memory:

$$c_t = i_t \odot u_t + \Sigma_{k\in C(t)} f_{tk} \odot c_k \tag{16}$$

The definition of r_t is dependency type memory:

$$r_t = ReLU(W^{(r)}z_t + b^{(r)}) \tag{17}$$

Here, z_t represents input dependency type vector.

The definition of o_t is output probability vector:

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{c(t)} + b^{(io)}) \tag{18}$$

The definition of h_t is the output vector:

$$h_t = o_t \odot tanh\,(c_t \odot r_t) \tag{19}$$

The transition gate δ is used to transition the hidden state h_t. When $D_t > \frac{1}{2}$, it indicates that the degree of dependence between the two words is weak, and the conversion gate $\delta_t = 0$. This means that the previous state does not provide useful information for the current time step. When $D_t = \frac{1}{2}$, it means that the degree of dependence between two words is general, and the conversion gate $\delta_t = 0$. When $D_t < \frac{1}{2}$, it means that the dependency of grammatical paths between two words is strong. The current hidden state depends on the previous grammatical structure.

The definition of δ_t is the output conversion gate:

$$\delta_t = ReLU\left(\frac{1}{2} - D_t\right) \tag{20}$$

The definition of $\widetilde{h_t}$ is the transformed hidden state:

$$\widetilde{h_t} = \delta_t * h_{t-1} + (1 - \delta_t) * h_t \tag{21}$$

3.3. The overall structure of the RSD-DT-LSTM model

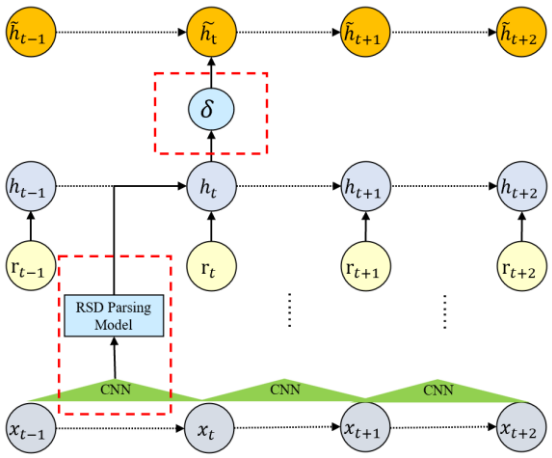


Fig.3 RSD-DT-LSTM model structure diagram

The structure of the relative syntactic distance dependency type long short-term memory (RSD-DT-LSTM) model is shown in Fig.3. Input word vector x_t, and use convolutional neural network to calculate the relative syntactic distance between two words. Input the relative syntactic distance and dependency type into the hidden state h_t. The transformed hidden state $\widetilde{h_t}$ is obtained through the transition gate δ. The gating mechanism of this model depends not only on the information of the word embeddings, but also on the pattern of the syntax tree. An auxiliary module δ is also added to the gating mechanism of this model to connect the previous hidden state with the current hidden state. The current hidden state is then fed to another transition gate based on the relative syntactic distance to determine the final output at the current time step. In this way, the model can more accurately cover all possible dependencies.

4. EXPERIMENTS

In order to verify the effectiveness of the model, two public datasets are used for experiments. SICK (Sentences Involving Compositional Knowledge) dataset and MRPC dataset (Microsoft Research Paraphrase Corpus). The SICK data set consists of 9927 sentence pairs. Its label is the similarity score between two sentences. The score range is 1-5. 5 points indicate complete similarity and 1 point indicates complete dissimilarity. MRPC data set is a standard data set opened by Microsoft, with a total of 5800 short text pairs. The label of the text pair with the same semantics is 1, and the label with different semantics is 0.

4.1. Experimental analysis of SICK data

The training and evaluation method proposed by Tai [13] were followed. The SICK dataset was split into 4500/500/4927 train/dev/test for experiments. Sentences in the dataset were parsed using the Stanford Dependency Parser[14]. And use the pre-trained Glove word embeddings[15] to initialize word representations. Table 1 lists the range of hyperparameters used. Choose the best hyperparameter values based on the experimental results.

Table 1. Hyperparameter ranges used to tune the model

Parameter	SICK-R
Learning rate	0.01/**0.025**/0.05/0.1/0.2/0.3
Batch size	**25**/50/100
Memory dimension	120/**150**/100
Weight decay	**0.0001**
Optimizer	**adagrad**/adam/nadam

For the semantic similarity score, the Pearson correlation coefficient and the mean squared error between the actual similarity score y and the predicted score y' is used to evaluate the performance of the model. The experimental results are compared with the LSTM model and DT-LSTM. The program for the LSTM model is obtained from the github website. The DT-LSTM model procedure is restored according to the procedure of the paper written by Jeena Kleenankandy et al[9].

Table 2. test results of SICK dataset

Model	Pearson	MSE
LSTM	0.846106	0.290754
DT-LSTM	**0.849289**	0.283802
RSD-DT-LSTM	0.848517	**0.282682**

As shown in Table 2, the RSD-DT-LSTM model is used for semantic similarity scoring. Its mean square error (MSE) between the actual similarity score y and the predicted score y' is better than other models. But its Pearson correlation coefficient did not outperform the other models.

4.2. Experimental analysis of MRPC data

The MRPC dataset was divided into 2160/290/2900 train/dev/test for experiments. The sentences in the dataset are parsed using Stanford dependency parser. And initializes the word representation using pre trained glove word embedding. Table 3 lists the range of hyperparameters used. The optimal hyperparametric value was selected according to the experimental results.

Table 3. Hyperparameter ranges used to tune the model

Parameter	SICK-R
Learning rate	0.01/**0.025**/0.05/0.1/0.2/0.3
Batch size	**30**/50/100
Memory dimension	120/150/**200**
Weight decay	**0.0001**
Optimizer	**adagrad**/adam/nadam

Table 4. test results of MRPC dataset

Model	Precision	Recall	F_1
LSTM	0.838104	0.9177491	0.882104
DT-LSTM	0.849685	0.920542	**0.886685**
RSD-DT-LSTM	**0.858936**	**0.923278**	0.885936

As shown in Table 4, the experimental results were compared with LSTM model and DT-LSTM model. Through the comparative experiment in Table 4, it can be seen that the precision and recall of RSD-DT-LSTM model are better than the other two models in the field of text similarity calculation, and the F_1 value is slightly worse than the DT-LSTM model.

4.3. Research Cases and Visualization

In this section, the performance of the RSD-DT-LSTM model is researched through experiments. In addition, the relative syntactic distance of different sentences is analyzed by us. In Figure 4 and Figure 5, Two different sentences are read from SICK data set and visualize their relative syntactic distance.

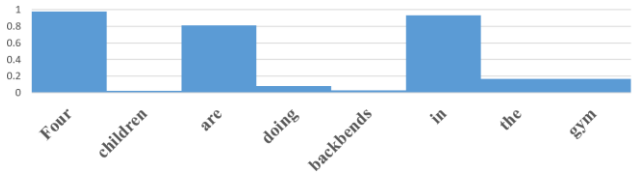


Fig.4 relative syntactic distance between words (short sentences)

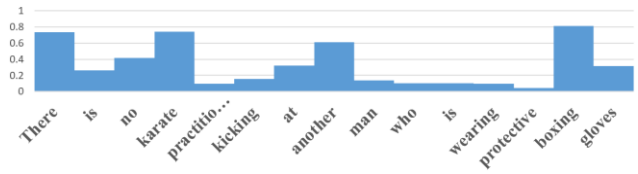


Fig.5 relative syntactic distance between words (long sentences)

As shown in Figure 4 and Figure 5, a column chart is used by us to visualize the relative syntactic distance computed by the RSD-DT-LSTM model. It is observed from the figure that if two words are in a phrase then they have a strong dependency in the syntax tree. Their relative syntactic distances tend to be small ("doing", "backbends" in shorter sentences). Relative syntactic distance tends to be large for two words that do not have strong dependencies ("no" and "karate" in longer sentences). This proves that this model is able to understand latent dependencies in sentences.

The "Four children are doing backbends in the gym" syntax tree is relatively simple. This model can easily model the dependencies between components. "There is no karate practitioner kicking at another man who is wearing protective boxing gloves" consists of multiple phrases. The hierarchy of the syntax tree of this sentence is more diverse than the previous short sentence, and the dependencies are more complex. From the visualization results, it can be seen that the relative syntactic distance can still describe the dependencies between most words whether it is a sentence with a simple structure or a complex structure.

5. CONCLUSION

In this paper, a model called relative syntactic distance-dependency type long short-term memory network (RSD-DT-LSTM) is proposed by us. Syntactic structure dependencies and dependency types are integrated into LSTMs without requiring any annotation data. The gating mechanism in DT-LSTM is modified according to relative syntactic distance. This mechanism serves as a scalar that controls the rate at which information is updated. Hierarchical syntactic dependencies can be fully encoded. A transition gate is provided to model syntactic structures more directly. This creates connections between hidden units. Finally, RSD-DT-LSTM model is applied to SICK data set and MRPC data set. The results show that the sentence representation obtained by RSD-DT-LSTM model is correct in the semantic similarity experiment. In the SICK dataset, the mean square error (EMS) between the actual correlation score y and the predicted score y ^ 'was 0.1% higher than that of the DT-LSTM model. The Pearson correlation coefficient obtained by RSD-DT-LSTM model is very close to that obtained by DT-LSTM model. In the MRPC dataset, its accuracy is 0.9% better than that of DT-LSTM model. And its recall is 0.3% better than that of DT-LSTM model.

The experimental results of semantic similarity score show that RSD-DT-LSTM language model can obtain sentence semantic similarity more accurately than other models. However, in sentences (especially long sentences), the relative syntactic distance can not fully understand the potential dependencies in sentences, and there are still some deviations. Therefore, the model is still insufficient in calculating the semantic linear relevance between two sentences, and will be improved in the following research. Finally, the visualization of the case further illustrates whether it is a sentence with simple structure or a sentence with complex structure. Relative syntactic distance can still describe the dependency between most words.

Although this model outperforms other language models in some aspects. However, the stability of convolutional neural network in calculating relative syntactic distance is not high. In the following research, the sentence representations obtained by this model are then used to conduct implicit discourse relations research.

REFERENCES:

[1] Han M, Zhang X, Yuan X, et al. A survey on the techniques, applications, and performance of short text semantic similarity[J]. Concurrency and Computation Practice and Experience, 2020.

[2] 郭浩, 许伟, 卢凯. 基于 CNN 和 BiLSTM 的短文本相似度计算方法[J]. 信息技术与网络安全, 2019, 38(6):5.

[3] 李景玉. 基于 BERT 的孪生网络计算句子语义相似度[J]. 科技资讯, 2021, 19(32):4.

[4] 杨萌, 李培峰, 朱巧明. 一种基于 Tree-LSTM 的句子相似度计算方法[J]. 北京大学学报：自然科学版, 2018, 54(3):6.

[5] Zhou Y, Liu C, Pan Y. Modelling Sentence Pairs with Tree-structured Attentive Encoder[C]. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers，2016.

[6] Arroyo-Fernandez I, Mendez-Cruz C F, Sierra G, et al. Unsupervised sentence representations as word information series: Revisiting TF-IDF[J]. Computer Speech & Language,2019,56(JUL.): 107-129.

[7] 石杰, 周兰江, 线岩团,等. 基于 WordNet 的中泰文跨语言文本相似度计算[J]. 中文信息学报, 2016, 30(4):6.

[8] 李欣, 李旸, 王素格. 面向情感聚类的文本相似度计算方法研究[J]. 中文信息学报, 2018, 32(5):8.

[9] Chambua J, Niu Z, Yousif A, et al. Tensor Factorization Method based on Review Text Semantic Similarity for Rating Prediction[J]. Expert Systems with Applications, 2018, 114(DEC.): 629-638.
[10] Li X, Liu N, Yao C, et al. Text similarity measurement with semantic analysis[J]. International Journal of Innovative Computing Information & Control, 2017, 13(5):1693-1708.
[11] Kleenankandy J, Abdul N. An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies[J]. Information Processing & Management, 2020:102362.
[12] Shen Y, Lin Z, Huang C W, et al. Neural Language Modeling by Jointly Learning Syntax and Lexicon[J]. International Conference of Learning Representation ,2017.
[13] Tai K S, Socher R, CD Manning. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks[J]. Computer Science, 2015, 5(1):: 36.
[14] Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP Natural Language Processing Toolkit[C].Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014.
[15] Pennington J , Socher R , Manning C. Glove: Global Vectors for Word Representation[C]. Conference on Empirical Methods in Natural Language Processing. 2014.