Cross-Lingual Semantic Textual Similarity Modeling Using Neural Networks: 14th China Workshop, CWMT 2018, Wuyishan, China, October 25-26, 2018, Proceedings

Chapter · January 2019

DOI: 10.1007/978-981-13-3083-4_5

CITATIONS
0

READS
572

3 authors, including:

Xia Li
Guangdong University of Foreign Studies
40 PUBLICATIONS **123** CITATIONS
SEE PROFILE

Minping Chen
13 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Cross-Lingual Semantic Textual Similarity Modeling Using Neural Networks

Xia Li[1,2(✉)], Minping Chen[2], and Zihang Zeng[2]

[1] Key Laboratory of Language Engineering and Computing, Guangdong University of Foreign Studies, Guangzhou, China
[2] School of Information Science and Technology/School of Cyber Security, Guangdong University of Foreign Studies, Guangzhou, China
shelly_lx@126.com, minpingchen@126.com, raymondtseng0912@126.com

Abstract. Cross-lingual semantic textual similarity is to measure the semantic similarity of sentences in different languages. Previous work pay more attention on leveraging traditional NLP features (e.g., alignment features, syntactic features) to evaluate the semantic similarity of sentences. In this paper, we only use word embedding as basic features without any handcrafted features and build a model which is able to capture local and global semantic information of the sentences to evaluate semantic textual similarity. We test our model on SemEval-2017 and STS benchmark datasets. Our experiments show that our model improves the performance of the semantic textual similarity and achieves the best results compared with the baseline neural-network based methods reported on the two datasets.

Keywords: Cross-lingual semantic textual similarity · SemEval-2017 Neural networks

1 Introduction

Cross-lingual semantic textual similarity measures the degree to which two sentences in different languages are semantically equivalent to each other, which is a fundamental language understanding problem in many fields, such as information retrieval, information extraction, machine translation and so on. Evaluation of sentence semantic similarity in English has achieved great success, but there are still some challenges in modeling sentences similarity in different languages due to lacking of enough training data for a particular language [1, 2]. In this paper, we focus on building a model to evaluate the semantic similarity between cross-lingual sentence pairs. We translated all the foreign languages into English by Google translator[1] following the state-of-the-art works in SemVal-2017 [3–5].

[1] https://cloud.google.com/translate/.

J. Chen and J. Zhang (Eds.): CWMT 2018, CCIS 954, pp. 52–62, 2019.
https://doi.org/10.1007/978-981-13-3083-4_5

Most previous works focus on leveraging traditional NLP features to evaluate the semantic similarity between cross-lingual sentence pairs [3, 4]. These hand-crafted features helped improve the performance of the models, but also are expensive for the system.

Few work only used neural features without any hand-crafted features in modeling semantic textual similarity between cross-lingual sentence pairs. Shao et al. [5] is one of the few exceptions. They use convolutional neural network to capture the semantic representation of the source and target sentences, and then fed both of them into the fully connected layer to get the semantic similarity scores of the two sentences. The model achieved good results in the task of SemVal-2017.

However, the model only captured the local semantic information of a sentence by CNN. The local n-grams information captured by CNN sometimes doesn't work well when modeling the semantic textual similarity in cross-lingual sentences. This is because when a language is translated into another language[2], the words order of the translated sentence may change or be inaccurate due to translation errors. Therefore, obtaining local n-grams information of a sentence through CNN may be insufficient due to the wrong words order caused by the translation.

On the other hand, in the task of cross-language semantic textual similarity, we not only need to pay attention to the local semantic matching information between sentences, but also need to consider the global semantic information of the sentences. Therefore, in order to better measure the semantic similarity of two sentences, we should integrate the semantic information of the source and target sentences in both local and global aspects as much as possible.

Based on these observations, this paper proposes a model which is able to capture the local and global semantic information between cross-lingual sentence pairs. Firstly, we use CNN with kernels of multiple sizes to capture the local matching information between sentence pairs. Then, we use the recurrent neural network to capture the semantic dependences of the sentence words and use this accumulation of the semantic relationship as the global semantic information of the sentence and output an accumulated vector of the sentence. Finally, the local information and the global information are concatenated to represent the final semantic of the sentence.

The architecture of our model is showed as Fig. 1. We do several experiments on SemVal-2017 data and STS benchmark data and the experimental results show that our model outperforms the current best neural network model without any manual NLP features.

2 Model

2.1 Sentence Encoding

Each source sentence and target sentence are initially represented by a fixed-size word-embeddings matrix. We use padding operation in our model. Supposing the maximum length of the sentence in our model is L, then if the number of words in a sentence is

[2] In this paper, all the other languages are translated into English.

less than L, we will add 0 to it. If the number of words in a sentence exceeds L, we will remove the extra words. All the words in the sentence are converted into corresponding word embedding, and then we can get a matrix representation of the sentence. Source sentence matrix and target sentence matrix are inputted into the model as sentence initial encoding.

2.2 Local Semantic Information Extraction

In order to capture the local matching information of the sentence pair sufficiently, we use convolution blocks of different convolution kernel sizes to convolve the sentences. As shown in Fig. 1, we convolve the sentences using three convolution blocks with convolution kernel windows of 1, 2, and 3, each with 300 convolution kernels. We also use ReLu function as activation function in CNN layer and maximum down-sampling for extracting the most informative vector from the output of convolution. The outputs of the three convolutional blocks pooling are jointed as a 900-dimensional semantic vector which represents the local semantic information of the sentences.

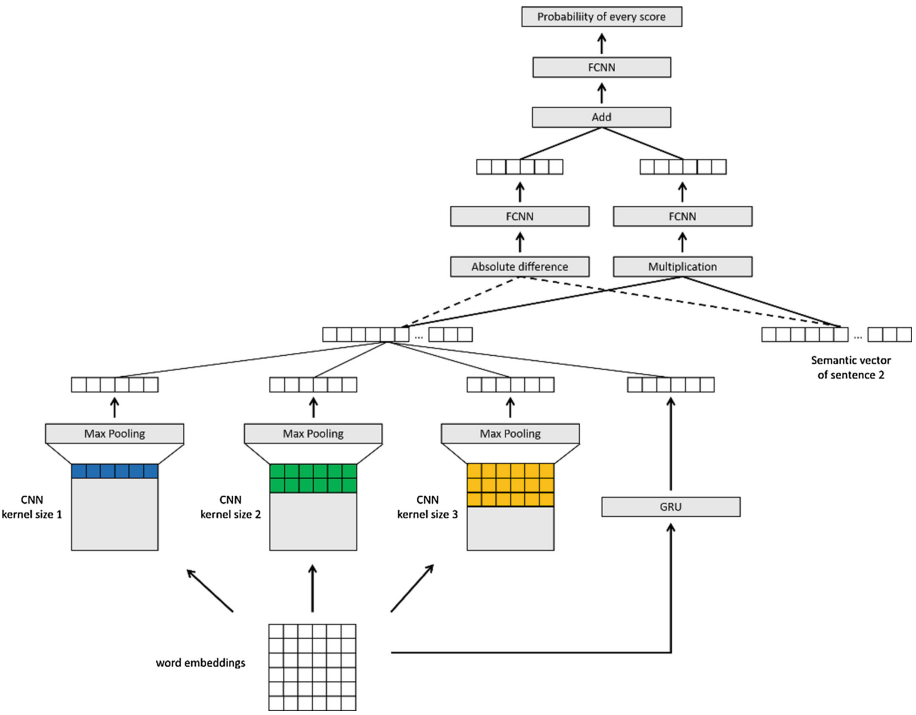

Fig. 1. Structure of our model based on convolutional neural network.

2.3 Global Semantic Information Extraction

In the cross-lingual sentence semantic similarity evaluation task, although the word order of the translated sentence may change, the semantic information of most words is correct.

Therefore, our motivation is to capture the global semantic information of the sentence through the information accumulation operation of the recurrent neural network.

Different from the previous work, our model does not take the average of the output of each GRU [6, 7] unit as the final output, but accumulates the semantic information of each word from left to right and takes the output of the last GRU unit as the final representation of the global semantics of the sentence (Fig. 2). Assuming that a sentence consists of m words S = $\{w_1, w_2, \ldots, w_m\}$. We use GRU to produce the internal states for the sequence at each timestep t: (y_1, y_2,…, y_{m-1}, …, y_m). As for our task, we use the last hidden state output y_m as the result of RNN with GRU units. We believe that this output accumulates the global semantic information of the sentence. As the output of GRU is also a 300-dimensional semantic vector, we concatenate the 900-dimensional local semantic vector obtained in Sect. 2.2 and the 300-dimensional global semantic vector as a 1200-dimensional vector and take it as a final semantic representation of the comprehensive information of the sentence.

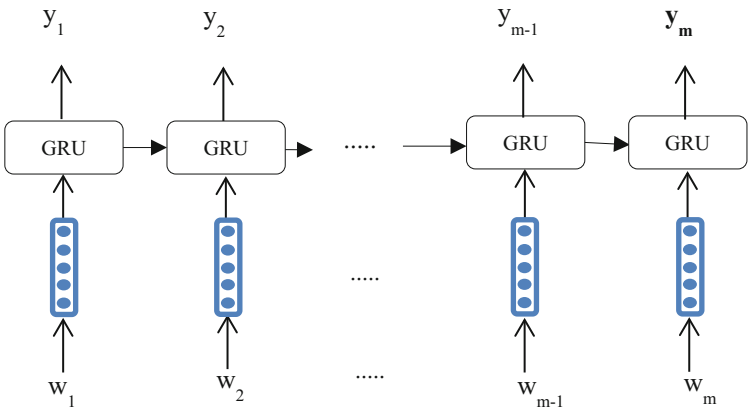

Fig. 2. Global semantic information extraction from GRU.

2.4 Representation of Semantic Similarity of Sentence Pairs

In order to calculate the text semantic similarity of two sentences, following Shao et al. [5], we carry two kinds of operations to the semantic representations of two sentences: absolute difference and multiplication. Here, the absolute difference operation for two sentences can be considered as getting the difference information between two semantic vectors, and the multiplication operation for two sentences can be considered as capturing the similarity information of two semantic vectors.

Different from the work of Shao et al. [5], we do not directly concatenate these two vectors of absolute difference operation and multiplication operation to get a final vector to represent the semantic similarity of sentence pairs. Instead, we firstly input the difference result vector and the multiplication result vector to a fully connected layer respectively. Then we add the two output vectors from these two fully connected layers

and take it as a final representation of semantic similarity of the source sentence and target sentence. The formula is shown in Eqs. (1)–(3).

$$h\_sub = W_1 \times \left|\overrightarrow{SV_1} - \overrightarrow{SV_2}\right| + b_1 \tag{1}$$

$$h\_mul = W_2 \times \left|\overrightarrow{SV_1} \cdot \overrightarrow{SV_2}\right| + b_2 \tag{2}$$

$$\overrightarrow{\mathrm{SDV}} = h\_sub + h\_mul \tag{3}$$

Here $\overrightarrow{SV_1}$ and $\overrightarrow{SV_2}$ are the final semantic vectors of the two sentences. W_1, b_1, W_2, b_2 are weights used in fully connected layers. The number of neurons used here is 900. $\overrightarrow{\mathrm{SDV}}$ is the final representation of semantic similarity between two sentences. We put $\overrightarrow{\mathrm{SDV}}$ into a fully connected layer and the Softmax layer. The fully connected layer has 900 neurons and use the Tanh activation function followed by a dropout layer with a dropout rate of 0.5. At the last layer, the number of neurons is 6 as the range of semantic similarity score is 0–5 and a Softmax function is used to get the probability value of each score.

2.5 Loss Function

We use KL divergence as a loss function and use ADAM [8] as a gradient descent optimizer. Because the similarity score is a value range from 0 to 5 and our model will get the probability value of each score, we need to convert the similarity score into a fractional probability distribution. The conversion formula is as Eq. (4).

$$p_i = \begin{cases} y - |y|, & i = |y| + 1 \\ |y| - y + 1, & i = |y| \\ 0, & otherwise \end{cases}, i \in [0, K] \tag{4}$$

Where K is the maximum similarity score. Loss function is the KL divergence between the standard fractional probability distribution p and the predicted fractional probability distribution $\hat{p}_\theta$. KL divergence can effectively measure the difference between two distributions and when two distributions are the same, the value is zero. The loss function of our model is shown in Eq. (5).

$$\mathrm{J}(\theta) = \frac{1}{m}\sum_{k=1}^{m} KL(p^{(k)}||\hat{p}_\theta^{(k)}) + \frac{\lambda}{2}\|\theta\|_2^2 \tag{5}$$

Here m is the total number of sentence pairs in the training set, k represents the kth sentence pair in the training set, θ_2^2 is the L2 loss of parameters in the network, λ is coefficient of L2 loss.

3 Experiments

3.1 Datasets

In our experiment, we use two datasets as our training data and test data. Because of the need for large size of training data, following work of [3–5], we collect the English sentence similarity dataset in the text semantic similarity task from SemEval-2012 to SemEval-2016 as our training data[3]. We randomly divide the collected data into 10 parts. 90% of the data is used as training data which contains 13,191 sentence pairs in total, and the other 10% of the data is used as a development set which contains 1,465 sentence pairs. We use multi-lingual and cross-lingual dataset [2] in the text semantic similarity task in SemEval-2017 as our test data[4]. In this task [2], there are 7 tracks with total 1750 sentence pairs, each track has 250 sentence pairs. The details of the SemEval-2017 data are shown in Table 1.

Table 1. Details of SemEval-2017 dataset.

Track	Language	Test pairs
1	Arabic (ar-ar)	250
2	Arabic-English (ar-en)	250
3	Spanish (es-es)	250
4a	Spanish-English (es-en)	250
4b	Spanish-English (es-en)	250
5	English (en-en)	250
6	Turkish – English (tr-en)	250
Total	–	1750

We also use the SST benchmark dataset in SemEval-2017[5] to test our model. The training data in SST benchmark contains 5749 sentence pairs, the development set contains 1500 sentence pairs, and the test set contains 1379 sentence pairs. Details of the two datasets are shown in Table 2.

Table 2. Details of two datasets used in our experiments.

Task	Training data	Test data	Development data
SemEval2017 Task1	13191	1465	1750
STS benchmark	5749	1500	1379

[3] http://ixa2.si.ehu.es/stswiki/index.php/Main_Page.

[4] http://alt.qcri.org/semeval2017/task1/index.php?id=data-and-tools.

[5] http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark.

3.2 Experimental Setup

We use the Pearson correlation coefficient as our evaluation metric in our experiment following SemEval-2017 task. The Pearson correlation coefficient can be used to reflect the degree of linear correlation between the two variables in a range of [− 1,1]. In our experiment, the Pearson correlation coefficient is calculated between the gold similarity score graded by human and the predicted similarity score graded by our model.

We implement some preprocessing operations for the data which include: (1) Remove all punctuations in the sentences; (2) Transform all words into lowercase; (3) Use NLTK segmentation to segment words; (4) Use pretrained paragram[6] word embedding [9] to represent the words of sentences. If the word does not exist in the paragram vocabulary, it is set to 0. (5) The length of all sentences is converted to a fixed size of 30. Those sentences with size larger than 30 will be cut, and shorter than 30 will be filled in zero. The hyperparameters used in our experiments are shown in Table 3. The initial leaning rate of our model is set to 0.001 and the batch size is 64. Besides, a dropout rate of 0.5 and regularization of 0.004 are used to avoid overfitting.

Table 3. Parameters of our model

Steps	Paramters	Value
Preprocessing	Sentence size	30
	Dimension of paragram	300
Similarity computing model	Convolution kernel size	1,2,3
	Convolution kernels	300
	Convolution neural network activation function	Relu
	Fully connection layer neuros (first layer)	900
	Fully connection layer neuros (second layer)	6
Training	Optimizer	ADAM
	Minimum batch size	64
	Learning rate	0.001
	Dropout rate	0.5
	Regularization	0.004

3.3 Experimental Results and Analysis

We do several experiments on SemEval-2017 Task1 and STS benchmark datasets. In our experiments, we use top 3 systems on SemEval-2017 Task1 and top 4 systems on STS benchmark as our baselines [3–5, 9]. Among these four systems, the Rank 1 system (ECNU) [3] ensembled several machine learning methods and three neural network models with rich traditional NLP features. These features include a total of 67 manual features such as sentence pair matching features, n-gram overlaps features, sequence features, syntactic parse features, machine translated based features,

[6] https://drive.google.com/file/d/0B9w48e1rj-MOck1fRGxaZW1LU2M/view.

alignment features and single sentence features etc. The Rank 2 system (BIT) [4] was completely based on traditional features. The system introduced semantic information space (SIS), which is constructed based on the semantic hierarchical taxonomy in WordNet, to compute non-overlapping information content (IC) of sentences. The system ranks 2nd on SemEval-2017 Task1. And the Rank 3 system (HCTI) [5] was a neural-network based method which only used convolutional neural networks to capture the semantic information of the sentences without any handcrafted features.

Results on SemEval-2017. As shown in Table 4, the average Pearson correlation coefficient of our model in the SemEval-2017 Task1 dataset is 69.61 which is higher than the rank 2 system BIT model [4] for 1.72% and higher than the rank 3 system HCTI [5] for 3.64%. Compared with neural-network based system [5], our model outperforms on 6 tracks, especially on track 4b which is Spanish to English, our model outperforms for 9.55%. Although our model achieves better results than the neural-network based systems HCTI [5] and also outperforms than traditional-features based system BIT [4], the results of the work ECNU [3] are better than ours. As ECNU used 67 rich traditional NLP features and ensembled different machine learning methods and neural network models, it is more powerful to model the sentence pairs and can integrate the advantages of different models. However, compared with neural-network-based models, which can automatically learn the features of the sentences, the work to extract the features of ECNU is very heavy and time-expensive.

Table 4. Results of different methods on SemEval-2017 Task1 dataset.

Models	Primary	Track1 AR-AR	Track2 AR-EN	Track3 SP-SP	Track4a SP-EN	Track4b SP-EN-WMT	Track5 EN-EN	Track6 EN-TR
ECNU [3]	73.16	74.40	74.93	85.59	81.31	33.63	85.18	77.06
BIT [4]	67.89	74.17	69.65	84.99	78.28	11.07	84.00	73.05
HCTI [5]	65.98	71.30	68.36	82.63	76.21	14.83	81.13	67.41
Our model	69.61	74.37	72.96	81.13	80.22	24.38	83.34	70.90

Results on STS Benchmark Dataset. As shown in Table 5, on the STS benchmark dataset, the Pearson correlation coefficient on the development data is 85.41% and is 79.38% on the test data. According to the STS benchmark Wiki, our model achieves the best performance on the dev data and the rank 3 on the test data. However, compared with neural-network-based models without any feature engineering, our model achieves the best results.

Table 5. Results of different methods on STS benchmark dataset.

	Dev	Test
ECNU [3]	84.70	81.00
BIT [4]	82.91	80.85
DT TEAM [18]	83.00	79.20
HCTI [5]	83.32	78.42
Our model	85.41	79.38

4 Related Work

In recent years, neural-network based methods have achieved excellent results in many different tasks. Some previous work has used neural network methods in the field of semantic text similarity [1–7, 9–13]. The word embedding [14, 15] generated by the neural network is a basic research work for calculating the similarity of texts for neural-network based methods. Word embedding is low-dimensional real number vector unsupervised trained from large-scale texts, aiming to represent tokens. This representation can better reflect the contextual semantics of the word. It can better solve the problem brought by traditional bag of the words such as high-dimensional sparseness and lack of sequences semantics. Some widely used word embeddings are Word2Vec [16] and Glove [17].

Many studies used word embeddings as the basis features to calculate text similarities through various formulas or models. Kusner et al. [18] thought that the minimum movement distance needed to move all the words from one text to the corresponding word in the word vector space is the similarity of the two texts. This method only considered the semantics of words ignoring the semantics of the entire sentence or the entire document which can cause deviations when computing similarities. Pagliardini et al. [19] proposed a sen2vec model based on the word vector. They used n-grams to combine the word vectors into sentence vectors and extended the word context to the sentence context. The experimental results showed that sentence vectors can express semantic information more effectively. Tai et al. [20] used the long short-term memory network (LSTM) [21] and bidirectional long short-term memory network (BiLSTM) to calculate semantic text similarity. At the same time, the syntactic analysis was introduced in the long short-term memory network which further mined the semantic information through the semantic dependency tree. Although this method can effectively extract the global semantic information of sentences, it is computationally intensive and does not work well for the long texts, and it did not pay attention to local information in sentences which is important for sentence textual similarity.

Different from previous works which used many traditional NLP handcrafted features to evaluate the semantic similarity of sentences [3, 4], Shao et al. [5] used a convolutional neural network to calculate the similarity of semantic texts, but their model only used a convolution kernel of size 1 which only can focus on the local information of the unigram without considering global information of sentences.

In this paper, we build a text similarity calculation model based on the convolutional neural network without any traditional NLP features. Also, different from previous methods, we represent the sentences with the local and global semantic information. We use convolution kernels of size 1, 2 and 3 to capture unigrams, bigrams and trigrams information as local information vectors of the sentence and use the last hidden output of recurrent neural network with GRU units to capture the global semantic information of the sentence. Combining vectors with local and global information into one vector which can be regarded as a final semantic representation of the sentences. We perform absolute difference and multiplication operations on the source and target sentence and then input them into a fully connected layer respectively, and we can get a final representation of the semantic similarity of the sentence

pair by adding operation. Several experiments showed that our model performs well on SemEval-2017 task1 and STS benchmark datasets.

5 Conclusion

In this paper, we build a model to evaluate cross-lingual sentence semantic textual similarity based on the neural networks without any traditional manual NLP features. Different from previous models, we represent the sentence pairs with joint of the local and global semantic information captured from the sentence. We use three kinds convolution kernels of window size 1, 2 and 3 to capture unigrams, bigrams and trigrams information as local semantic information vectors of the sentence. And then, we use recurrent neural network with GRU units and take the last hidden states as the global semantic information of the sentence. We combine these vectors with local and global semantic information into a vector to be a final representation of the semantic of the sentences. In order to get the semantic similarity of the sentences, we perform absolute difference and multiplication operations on the sentence pair and then input them into a fully connected layer respectively, getting a final representation of the semantic similarity of the sentence pair. We test our model on SemEval-2017 Task1 and STS benchmark datasets. Despite the simplicity of our model, the results of our several experiments in two datasets show that our model has a good performance.

There is still some work to do with our model in the future. First, the words order of a sentence may change when it is translated from one language into English, but the relationship between each two words anywhere in the sentence is relatively constant. Therefore, in the future, we will consider incorporating the self-semantic relationship of the words of a translated sentence into the semantic representation of the sentence in order to improve the performance of the model. In addition, we also see that the ECNU system [3] still outperforms our neural network system due to the use of rich traditional features. In the future, we will consider incorporating traditional features into neural network to enhance learning and improve the performance of the model.

Acknowledgement. This work is supported by the National Science Foundation of China (61402119) and Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation. ("Climbing Program" Special Funds.)

References

1. Agirre, E., et al.: Semeval-2016 Task 1: semantic textual similarity, monolingual and cross-lingual evaluation. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 497–511 (2016)
2. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: SemEval-2017 Task 1: semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055 (2017)

3. Tian, J., Zhou, Z., Lan, M., Wu, Y.: ECNU at SemEval-2017 Task 1: leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 191–197 (2017)
4. Wu, H., Huang, H.Y., Jian, P., et al.: BIT at SemEval-2017 Task 1: using semantic information space to evaluate semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 77–84 (2017)
5. Shao, Y.: HCTI at SemEval-2017 Task 1: use convolutional neural network to evaluate semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 130–133 (2017)
6. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014)
9. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Towards universal paraphrastic sentence embeddings. arXiv preprint arXiv:1511.08198 (2015)
10. Agirre, E., Diab, M., Cer, D., Gonzalez-Agirre, A.: Semeval-2012 Task 6: a pilot on semantic textual similarity. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pp. 385–393 (2012)
11. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: * SEM 2013 shared task: semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, vol. 1, pp. 32–43 (2013)
12. Agirre, E., et al.: Semeval-2014 Task 10: multilingual semantic textual similarity. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 81–91 (2014)
13. Agirre, E., et al.: Semeval-2015 Task 2: semantic textual similarity, English, Spanish and pilot on interpretability. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 252–263 (2015)
14. Hinton, G.E.: Learning distributed representations of concepts. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, vol. 1, p. 12, (1986)
15. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(6), 1137–1155 (2003)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
17. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
18. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning, pp. 957–966 (2015)
19. Pagliardini, M., Gupta, P., Jaggi, M.: Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv preprint arXiv:1703.02507 (2017)
20. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075 (2015)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)