A Siamese CNN Architecture for Learning Chinese Sentence Similarity

Haoxiang Shi
Waseda University
Tokyo, Japan
hollis.shi@toki.waseda.jp

Cen Wang
KDDI Research, Inc.
Saitama-ken, Japan
ce-wang@kddi-research.jp

Tetsuya Sakai
Waseda University
Tokyo, Japan
tetsuyasakai@acm.org

Abstract

This paper presents a deep neural architecture which applies the Siamese Convolutional Neural Network sharing model parameters for learning a semantic similarity metric between two sentences. In addition, two different similarity metrics (i.e., the Cosine Similarity and Manhattan similarity) are compared based on this architecture. Our experiments in binary similarity classification for Chinese sentence pairs show that the proposed Siamese convolutional architecture with Manhattan similarity outperforms the baselines (i.e., the Siamese Long Short-Term Memory architecture and the Siamese Bidirectional Long Short-Term Memory architecture) by 8.7 points in accuracy.

1 Introduction

Measuing the similarity between words, sentences, paragraphs and documents is an important component in various tasks such as information retrieval, document clustering, word-sense disambiguation, automatic essay scoring, short answer grading, machine translation and text summarization. Traditional sentence similarity measurement is based on the edit distance, Jaccard index, and the bag-of-words models such as TF-IDF. These methods of learning sentence similarity are in fact based on the word level, which may not be sufficient. For example, there are two Chinese sentences as shown in Figure 1. The corresponding English translations are “How to buy LCD TVs.” and “What kind of LCD TVs is good?”. From the word level (i.e., character level in Chinese), the two sentences look the same, but they have totally different meaning at the sentence level. That is, we need sentence-level methods to capture the semantics of the sentences for sentence similarity measurement.

With the rapid development of machine learning, using neural network to learn representations of sentence-level meanings has been widely verified to be effective. The beginning of using neural network to learn sentence-level representations may be the Word2Vec from Google (Mikolov et al., 2013), which used a shallow structure to learn the vector-based representations of sentence level. However, using one neural architecture to learn two sentences in two steps may cause inconsistent representations. Hence, Siamese structures, which can learn two sentences at a time, are attractive alternatives. The Siamese architecture that can achieve state-of-the-art accuracy results in learning English sentence similarity is a Bidirectional Long Short-Term Memory (Bi-LSTM) based Siamese recurrent architecture (Neculoiu et al., 2016).

In our preliminary study, we tested the effectiveness of a Siamese recurrent architecture for learning Chinese sentence similarities. However, this did not perform as well as what is reported in (Neculoiu et al., 2016). Therefore, we borrowed a Siamese convolutional architecure from the image processing field (Koch et al., 2015) to implement Chinese sentence similarity learning. The results in binary similarity classification for Chinese sentence pairs show that Siamese convolutional architecture outperforms the Siamese recurrent architecture in learning accuracy. In addition, we consider two similarity metrics in the Siamese convolutional architecture, namely, the Manhattan similarity and the Cosine similarity. The results show that the Siamese convolutional architecture plus the Manhattan similarity performs better than other baselines for learning the similarity between two Chinese sentences. Our contributions are as follows: (1) we verified that Siamese convolutional architecture is effective in learning Chinese sentence semantic similarity; (2) we verified that the Manhattan similarity can achieve better performance than other similarity metrics regardless of the learning architectures.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 24–29
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

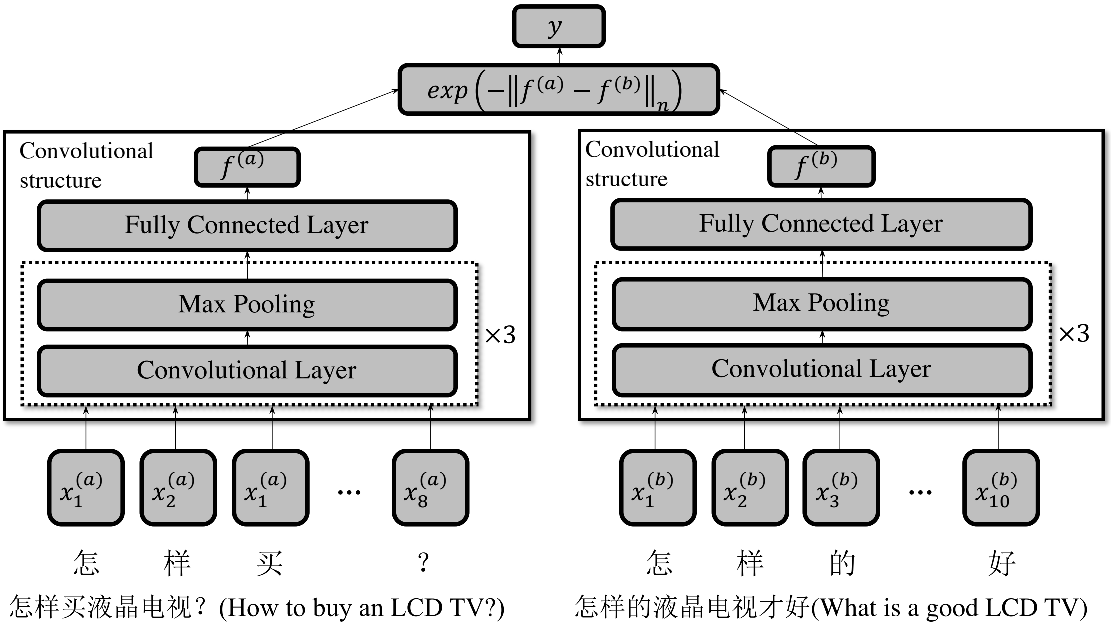


Figure 1: Siamese convolutional architecture.

2 Related Work

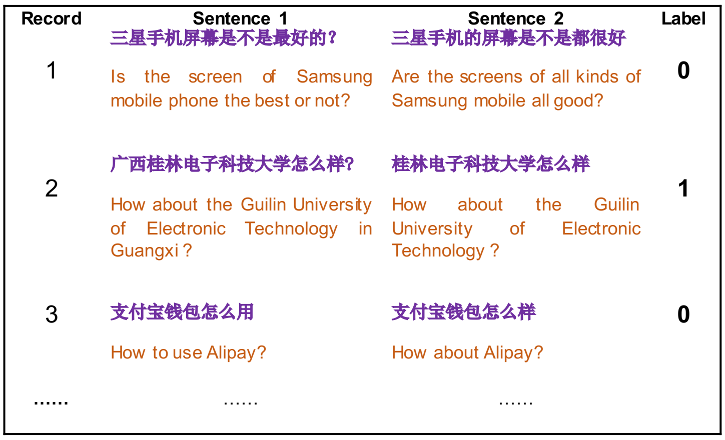
The Siamese network (Bromley et al., 1993) is firstly proposed for non-linear metric learning with similarity information. It naturally learns representations that embody the invariance and selectivity desiderata through explicit information about similarity between pairs of objects. The Siamese architecture has since been widely used in vision applications. Specifically, the Siamese convolutional networks were used to learn complex similarity metrics for face verification (Chopra et al., 2005) and dimensionality reduction on image features (Hadsell et al., 2006). While in the natural language processing (NLP) field, the Convolutional Neural Network (CNN) has attracted more attentions since the successes in using CNN to do the traditional NLP tasks (Collobert et al., 1993), and the availability of high-quality semantic word representations has been verified when using the CNN (Mikolov et al., 2013).

Recently, CNNs have been applied to matching sentences (Hu et al., 2014). Although the work (Hu et al., 2014) has used the CNN to learn representations of two sentences, this is not a Siamese CNN architecture. Following this, the Siamese Long Short-Term Memory (LSTM) architecture was proposed for sentence similarity task using token level embedding (Mueller and Thyagarajan, 2016). Subsequently, a Siamese Bi-LSTM structure was proposed in order to improve the result of sentence similarity (Neculoiu et al., 2016). A Siamese CNN combines Bi-LSTM structure has been proposed for learning sentence similarity (Pontes et al., 2018). However, this architecture achieves lower accuracy than the independent Bi-LSTM structure. Also, the paper (Pontes et al., 2018) did not give any comparisons between Siamese CNN architecture and Siamese Bi-LSTM architecture. Later, Siamese LSTM and Siamese Bi-LSTM were compared based on an English dataset (Ranasinghe et al., 2019).

3 Siamese Convolutional Architecture

The proposed Siamese convolutional architecture is depicted in Figure 1. In the architecture, there are two exactly alike convolutional structures that are used. The inputs of each convolutional structure are the character-level embeddings of a sentence, and the outputs of each convolutional structure are the sentence level representations. Then, a similarity metric is used to compare the outputs of the two convolutional structures. The calculated similarity is the final output of the Siamese convolutional architecture.

Within each convolutional architecture, there are one fully connected layer after three repeated convolutional layers and max pooling layers. We have also tested the six repeated structure, but the accuracy did not show a significant improvement. The kernel size of each convolutional layer is different. A higher convolutional layer is equipped



Record	Sentence 1	Sentence 2	Label
1	三星手机屏幕是不是最好的？ Is the screen of Samsung mobile phone the best or not?	三星手机的屏幕是不是都很好 Are the screens of all kinds of Samsung mobile all good?	0
2	广西桂林电子科技大学怎么样? How about the Guilin University of Electronic Technology in Guangxi ?	桂林电子科技大学怎么样 How about the Guilin University of Electronic Technology ?	1
3	支付宝钱包怎么用 How to use Alipay?	支付宝钱包怎么样 How about Alipay?	0
……	……	……	

Figure 2: The format of the LCQMC.

with a larger kernel size. The fully connected layer then reduces the dimension of the learned representations from pooling layer. The learned output vector from the fully connected layer will be used to calculate the similarity then.

The similarity depicted in Figure 1 is the exponential negative norm of two learned representation vectors, which is defined as:

$$sim_{Man} = \exp\left(-\left\|f^{(a)} - f^{(b)}\right\|_n\right) \qquad (1)$$

where in equation (1) $f^{(a)}$ and $f^{(b)}$ are the representations of the two sentences from the two convolutional structures. If $n = 1$, the similarity is the Manhattan distance-based similarity or the Manhattan similarity for short. If $n = 2$, the similarity is then the Euclidean distance-based similarity or the Euclidean similarity for short. We have also tested the performance of the Siamese network with Euclidean similarity. The accuracy is around 50%, which means the Euclidean similarity does not work well with the Siamese architecture. Therefore, this result is not shown in Section 4. The similarity can also be replaced by the Cosine similarity.

$$sim_{Cos} = \frac{(f^{(a)} \cdot f^{(b)})}{\left\|f^{(a)}\right\| \cdot \left\|f^{(b)}\right\|} \qquad (2)$$

After calculating the similarity, we then use the mean-square error (MSE) of the similarity and the label as the loss function. The gradients of the loss will be fed back to both convolutional structures. In this way, the two convolutional structures will share the same parameters, and then they can learn the representations of the two sentences with the same distribution. Based on a threshold of the similarity, we can then evaluate the accuracy after learning.

4 Experiments

Our experiments are the binary similarity classification tasks for Chinese sentence pairs. Although obtaining a Chinese sentence similarity dataset is difficult, we found a dataset named LCQMC with even distribution of the labels (i.e., similar sentence pairs and dissimilarity sentence pairs occupy 50% and 50% of all dataset respectively) from Baidu. The format of the dataset is shown in Figure 2. Punctuations of some sentences are omitted in the original data. This dataset consists 283,000 data records. We have chosen 250,000 data records as the training data, and 12,500 data records as the test data. A data record is like <sentence 1, sentence 2, similarity> (i.e., 1 represents that the two sentences are similar and 0 represents that two sentences are dissimilar). We also used the English dataset PAWS-X (Yang et al., 2019) to train and test the different saimese architecture as the comparisons. In the PAWS-X, the data format is the same with LCQMC, and the task is also to learn the semantic similarity between two sentences. We used 49,401 data records in PAWS-X to train models and 2,000 to test.

From the aforementioned related works, (Neculoiu et al., 2016) and (Ranasinghe et al., 2019), we have chosen two baselines: the Siamese Bi-LSTM architecture and the Siamese LSTM architecture. Moreover, we also evaluated the two baslines and the Siamese convolutional architecture with two different loss functions (i.e., the Manhattan simi-

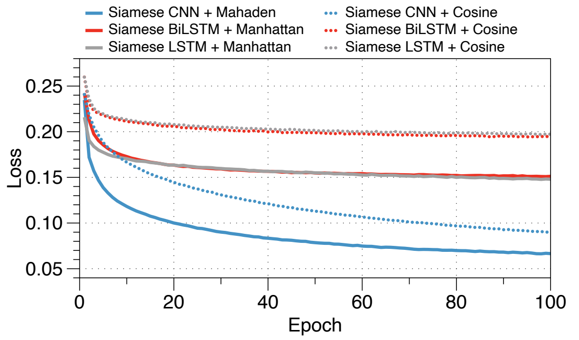


Figure 3: The convergences and the losses of the Siamese convolutional architectures and the baselines.

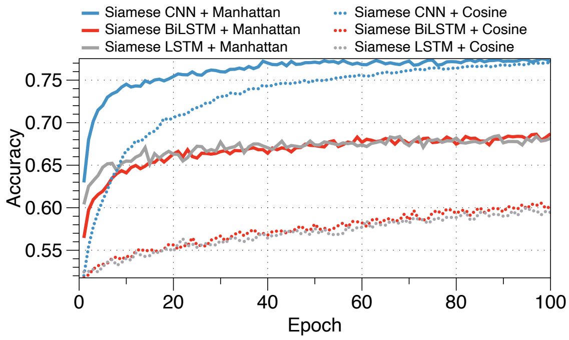


Figure 4: The accuracies of the Siamese convolutional architectures and the baselines.

larity based MSE and the Cosine distance based MSE).

As for the Siamese convolutional architecture, the kernel sizes of the three repeated convolutional layers are set as 3, 4 and 5. We ran a total of 100 epochs and the batch size of each epoch is 128. The Adam optimizer is used. During the optimization, we set the learning rate to be 0.001. Following previous work (Neculoiu et al., 2016), we used accuracy as the evaluation metric. We then set the similarity threshold as 0.5. That is to say, if the calculated similarity is more than 0.5, the prediction is that the two sentences are similar. Conversely, the similarity less than 0.5 is decided as dissimilar.If the similarity is exactly 0.5, the result is excluded for calculating accuracy.

In Figure 3, we compared the convergence speeds and losses of all combinations of the Siamese architectures and the two loss functions. The lines in different colors represent different Siamese architectures. The full lines are the losses using the Manhattan similarity, and the dotted lines are the losses using the Cosine similarity. It can be observed that no matter what kind of the Siamese architecture is used, the Manhattan similarity based Siamese architectures converge fast. As for the loss, the Siamese convolutional architectures always achieve lower losses than the baselines. In the Siamese convolutional architectures, the Manhattan similarity based Siamese architecture always gets a lower loss. As a result, the Siamese convolutional architecture with the Manhattan similarity metric achieves the lowest loss. Regardless of the choice the similarity metric, the losses of the Siamese LSTM architecture and the Siamese Bi-LSTM architecture are similar.

Next, we evaluated the accuracy of all the combinations of the Siamese architectures and the two loss functions. The representation formats of different combinations are the same with Figure 3. As shown in Figure 4, it can be seen that the Siamese convolutional architectures always achieve higher accuracy. In the Siamese convolutional architectures, the one with the Manhattan similarity metric always achieves higher accuracy. In summary, the Siamese convolutional architecture with the Manhattan similarity metric can obtain the highest accuracy. The performances of the two baselines are not substantially different regardless of the similarity metric. The Siamese Bi-LSTM architecture shows a slight improvement of the accuracy comparing to the Siamese LSTM architecture.

We listed all the experimental results in the Table 1, including using LCQMC dataset and PAWS-X dataset. It can be observed that when using LCQMC dataset, both Siamese convolutional architecture with the Manhattan similarity metric and with the Cosine similarity metric outperform the Siamese Bi-LSTM architecture and the Siamese LSTM architecture. Specifically, our Siamese convolutional architecture with Manhattan similarity metric outperforms the Siamese Bi-LSTM architecture with Manhattan similarity by 8.67 points and the Siamese LSTM architecture by 8.68 points respectively. In addition, our Siamese convolutional architecture with Cosine similarity metric also outperforms the Siamese Bi-LSTM architecture with the same similarity metric by 16.50 points and the Siamese LSTM architecture with the same similarity metric by 17.03 points respectively. However, when using PAWS-X, the English dataset, the Siamese LSTM architecture and Siamese Bi-LSTM architecture outperform Siamese convolutional architecture. In the experiments of learning English dataset, to improve the performance of the Siamese LSTM and

Dataset	Architecture	Manhattan Similarity	Cosine Similarity
LCQMC	Siamese convolutional architecture	**77.31**	**77.05**
	Siamese Bi-LSTM architecture	68.64	60.55
	Siamese LSTM architecture	68.63	60.02
PAWS-X	Siamese convolutional architecture	57.80	56.41
	Siamese Bi-LSTM architecture	67.75	**69.20**
	Siamese LSTM architecture	**68.14**	67.45

Table 1: Accuracy comparison for different architectures with Manhattan and Cosine similarities.

Bi-LSTM architectures, introducing Glove (Pennington et al., 2014) may be effective. The performance discrepancy of the Siamese convolutional architecture between Chinese and English may because that part of the CNN can do character-level encoding for Chinese. This is also why in some Chinese language tasks such as (Dai and Cai, 2017) and (Su and Lee, 2017), a CNN-based character-level encoder is added before the word-level or sentence-level encoding

5 Conclusion and Future Work

We proposed a Siamese convolutional architecture for Chinese sentence similarity learning. The experimental results have verified that the Siamese convolutional architecture outperforms the Siamese Bi-LSTM architecture and the Siamese LSTM architecture in terms of accuracy. Moreover, with the proposed architecture, we learned that for Chinese sentence similarity task, Manhattan similarity metric can help to achieve faster convergence and higher accuracy than any other similarity metric. Our results also suggest that the Siamese architectures which are effective in English NLP tasks may not necessarily work well in Chinese NLP tasks.

In the future, we will try to build and conduct experiments on Siamese Transformer (Vaswani et al., 2017) architecture. In addition, we will use BERT (Devlin et al., 2018) to obtain word embeddings as the inputs of the Siamese architecture to test performances.

References

J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. 1993. Signature verification using a siamese time delay neural network. *Advances in Neural Information Processing Systems*, pages 737–744.

S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 539–546.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 1993. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

F. Dai and C. Zheng Cai. 2017. Glyph-aware embedding of chinese characters. arXiv:1709.00028, 2017.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742.

B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional neural network architectures for matching natural language sentences. *Advances in Neural Information Processing Systems*, pages 2042–2050.

G. Koch, R. Zemel, and R. Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, pages 148–157.

T. Mikolov, K. Chen I. Sutskever, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119.

J. Mueller and A. Thyagarajan. 2016. Dimensionality reduction by learning an invariant mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*.

P. Neculoiu, M. Versteegh, and M. Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

E. L. Pontes, S. Huet, A. C. Linhares, and J. M. Torres-Moreno. 2018. Predicting the semantic textual similarity with siamese cnn and lstm. arXiv:1810.10641.

T. Ranasinghe, C. Orasan, and R. Mitkov. 2019. Semantic textual similarity with siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011.

T. Su and H. Lee. 2017. Learning chinese word representations from glyphs of characters. arXiv:1708.04755.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, pages 5998–6008.

Y. Yang, Y. Zhang, C. Tar, and J. Baldridge. 2019. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. arXiv:1908.11828.