



# Paraphrase Identification Based on Weighted URAE, Unit Similarity and Context Correlation Feature

Jie Zhou<sup>1</sup> , Gongshen Liu<sup>1</sup> , and Huanrong Sun<sup>2</sup>

<sup>1</sup> School of Electronic Information and Electrical Engineering,  
Shanghai Jiao Tong University, Shanghai, China

{sanny02, lgshen}@sjtu.edu.cn

<sup>2</sup> SJTU-Shanghai Songcheng Information Content Analysis Joint Lab.,  
Shanghai, China  
sunhuanrong@021.com

**Abstract.** A deep learning model adaptive to both sentence-level and article-level paraphrase identification is proposed in this paper. It consists of pairwise unit similarity feature and semantic context correlation feature. In this model, sentences are represented by word and phrase embedding while articles are represented by sentence embedding. Those phrase and sentence embedding are learned from parse trees through Weighted Unfolding Recursive Autoencoders (WURAE), an unsupervised learning algorithm. Then, unit similarity matrix is calculated by matching the pairwise lists of embedding. It is used to extract the pairwise unit similarity feature through CNN and k-max pooling layers. In addition, semantic context correlation feature is taken into account, which is captured by the combination of CNN and LSTM. CNN layers learn collocation information between adjacent units while LSTM extracts the long-term dependency feature of the text based on the output of CNN. This model is experimented on a famous English sentence paraphrase corpus, MSRPC, and a Chinese article paraphrase corpus. The results show that the deep semantic feature of text could be extracted based on WURAE, unit similarity and context correlation feature. We release our code of WURAE, deep learning model for paraphrase identification and pre-trained phrase end sentence embedding data for use by the community.

**Keywords:** Paraphrase identification · Recursive Autoencoders  
Phrase embedding · Sentence embedding · Deep learning · Semantic feature

## 1 Introduction

In general, paraphrase means expressing the same meaning in different words. With the development of NLP and paraphrase generation, there is a phenomenon that AI machine writers paraphrase similar news or stories on different websites and social medias. Paraphrase identification is useful in news event detection and first story detection. It is also helpful to other NLP applications, including question answering, information retrieval, plagiarism detection, machine translation evaluation and so on.

Paraphrase identification is a subtask in natural language processing (NLP), which aims at recognizing if the given pair of text convey same meaning. That pair of text might have different length and be expressed in different way. If a pairwise text have equivalent semantic, it would be labelled as paraphrase. In another way, a pairwise text is non-paraphrase if they have different meaning.

In this paper, a deep learning model is proposed for paraphrase identification, based on pairwise unit similarity feature and semantic context correlation feature. The pairwise unit similarity feature is extracted from given pairs of text through a convolutional neural model. Moreover, the work of [1, 9, 10] are extended to get the semantic context correlation feature based on CNN and LSTM. Also, for the purpose of learning phrase and sentence embedding, the work of [18] is extended to Weighted Unfolding Recursive Autoencoders (WURAE).

The model is adaptive to both sentence-level and article-level paraphrase identification (PI) task. The sentence-level PI task is experimented in an English sentence corpus, Microsoft Research Paraphrase Corpus (MSRPC), and compared with the state-of-art models. In our work, an extension to existing problem is made by introducing article-level paraphrase detection, detecting whether the given pair of articles talk about the same matter. The article-level PI task is experimented in a Chinese article paraphrase dataset, which is generated from sports and entertainment news.

In the rest of paper, we first review related works in Sect. 2. In Sect. 3, our methodology is introduced in detail. Experimental setup and results are discussed in Sect. 4. Finally, conclusion and future work plans are exposed in Sect. 5.

## 2 Related Work

The coverage of existing literature is about paraphrase identification (PI) and sentence embedding. The part of PI is divided into lexical similarity, semantic feature, syntactic feature and traditional features. The issue of sentence embedding is mainly about unsupervised learning method and certain-task-supervised learning method.

### 2.1 Paraphrase Identification (PI)

To compare the meaning of given pairwise text, a traditional method is based on their lexical similarity. The basic method includes Longest Common Subsequence (LCS) [2], similarity of name entity, calculating the cosine distance of word embedding, obtaining statistics feature by Vector Space Model (VSM), n-gram overlap and so on. [16] used corpus-based and knowledge-based measures of similarity with WordNet. A set of words in different order may differ in meaning. Thus, meaning of phrases should be taken into account. Considering continuous and discontinuous linguistic phrases, [8] extended TF-IDF by discriminative weights of words and phrases. With the development of neural networks and word embedding, deep learning algorithm is widely used in NLP. [6] proposed a pairwise semantic and lexical similarity measurement based on CNN. [9] figured out a method of using wide one-dimension convolution to get n-gram feature, which [1] have used in paraphrase detection. [1] also

combined it with LSTM to get semantic representation of sentences. [10] used multiple filter widths, getting various n-gram feature maps, in sentence classification task.

Syntactic feature is helpful for deep semantic comprehension. Structured alignment in syntactic feature based on dependency trees is explained in [15]. [18] used dynamic pooling layer to construct a fixed-sized similarity matrix from phrase embedding.

Some other features can be added to improve the accuracy of identification. Number feature was applied in [18]. The use of machine translation (MT) evaluation in paraphrase identification was explained by [14] which made use of 8 different MT metrics.

## 2.2 Sentence Embedding

The distributed representation of nature language makes the computer process natural language more convenient. Recently, many studies have proposed various methods for distributed representation of phrase, sentence or even paragraph. [9, 10] explained a way of modelling a sentence by CNN while they are both concerned on one certain topic, training sentence embedding with the labelled data. [13] advised a self-attention mechanism and a special regularization term. [12] proposed Paragraph Vector, an unsupervised learning algorithm, which learns fixed-length representation from variable-length pieces of sentences, paragraph or documents. [18] proposed Unfolding Recursive Autoencoders, an unsupervised learning method to calculate phrase or sentence embedding based on parse tree. [11] used continuity of text from books, training an encoder-decoder model that tries to reconstruct surrounding sentences of an encoded passage.

## 3 Methodology

The inputs of our deep learning models are the distributed representation of words. Then, the phrases and sentences embedding are learned from WURAE, an unsupervised learning algorithm trained by a large scale of both English and Chinese sentence corpus. In the sentence-level PI task, word and phrase embedding are regarded as the units of sentence like the nodes in parse tree. By analogy, sentence embedding is considered as the units of article in the article-level PI task. With the distributed representation of text, pairwise unit similarity feature is extracted from the unit similarity matrix through CNN and k-max pooling layers. In addition, semantic context correlation feature is learned from the combination of CNN and LSTM. Some other features are also added to the model. The probability of being paraphrased is predicted by the combination of features. The overall architecture of sentence-level paraphrase identification would be described in Sect. 3.5, including lexical, syntactic and semantic feature. And the entire architecture of article-level one would be explained in Sect. 3.6.

### 3.1 Distributed Representation of Words

Distributed representation of data is a must for applying deep learning method into NLP. Word embedding can convert one word in natural language into a node of vector

space, which helps computer process NLP tasks more convenient. With the implementation of word embedding, a sentence could be represented with a list of fixed-dimensional vectors. If a sentence is composed of  $n$  words and the dimension of word embedding is  $m$ , the sentence could be expressed as  $(w_1, w_2, \dots, w_i, \dots, w_n)$  where  $w_i$  equals  $(x_1, x_2, \dots, x_i, \dots, x_m)$ . In the work of learning phrase or sentence embedding, word embedding is the data of every leaf node in parse tree. A pre-trained word embedding with the dimension of 300, Google News vectors<sup>1</sup>, is used in the experiment of sentence-level PI task. Since the article-level paraphrase dataset is constructed from Chinese sports and entertainment news, we trained 300-dimensional vectors through Word2Vec algorithm based on the corpus of Chinese Wiki data<sup>2</sup> and Sogou News data<sup>3</sup>.

### 3.2 Distributed Representation of Phrase and Sentence

Owing to the diversity and complexity of natural language, although word embedding could represent sentences as lists of vectors, it is still difficult to get the accurate semantic feature. The phrases composed of ordered words are more important than separate words while understanding meaning of sentences. For the purpose of extracting deep semantic feature, there is a need to train on phrase or sentence embedding, capturing syntactic and semantic feature besides lexical one. In this research, the work of [18] is extended to Weighted Recursive Autoencoders (WURAE). Here we will introduce their previous work briefly and then propose our improvement on it.

**Unfolding Recursive Autoencoders (URAE).** Based on parse tree of sentence, we can obtain a binary tree structure representing the sentence. The leaf nodes of the tree are word embedding of words in the sentence. The internal nodes representing phrases and root node representing sentence are computed from their children, which is called as encoding part. The child node could be a leaf node or an internal node. For the given  $n$ -length sentence  $S$ , represent it with a list of  $m$ -dimensional vectors as  $S = (w_1, w_2, \dots, w_i, \dots, w_n)$  where  $w_i = (x_1, x_2, \dots, x_i, \dots, x_m)$ . In the encoding part, the parent node  $p$  is calculated from its children  $(c_1, c_2)$  by a standard neural network layer:

$$p = f(W_e[c_1; c_2] + b_e) \quad (1)$$

where  $[c_1; c_2]$  means the concatenation of its children,  $f$  is an element-wise activation function such as  $\tanh$ ,  $b_e \in R^m$  is the encoding bias vector and  $W_e \in R^{m \times 2m}$  is the encoding matrix to learn.

---

<sup>1</sup> <https://drive.google.com/file/d/0B7XkCwpI5KDYNINUTTlSS21pQmM/edit?usp=sharing>.

<sup>2</sup> <https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2>.

<sup>3</sup> <http://www.sogou.com/labs/resource>.

To optimize the training and improve the representation of phrase or sentence, the reconstruction is calculated during the decoding part. The decoding calculation of one parent node  $p$  reconstructs its children as  $(c'_1, c'_2)$ :

$$\begin{bmatrix} c'_1; c'_2 \end{bmatrix} = f(W_d p + b_d) \quad (2)$$

where  $f$  is an element-wise activation function,  $W_d \in R^{2m \times m}$  is the decoding matrix and  $b_d \in R^{2m}$  is the decoding bias vector. In the URAE, decoding part of node  $p_i$  reconstructs the entire subtree underneath  $p_i$ . With all the reconstructed leaf nodes underneath  $p_i$ , we could get the reconstruction error by computing Euclidean distance between the concatenation of original inputs and its reconstructions:

$$E_{rec}(y_{(i,j)}) = \left\| \begin{bmatrix} w_i; \dots; w_j \end{bmatrix} - \begin{bmatrix} w'_i; \dots; w'_j \end{bmatrix} \right\|^2 \quad (3)$$

where node  $y_{(i,j)}$  is encoded from leaf nodes  $(w_i, \dots, w_j)$ .

**Weighted Unfolding Recursive Autoencoders (WURAE).** As is mentioned above, URAE could calculate the distributed representation of phrases and sentence. Its method of reconstruction error ensures the increased importance of the child which has larger subtree. However, the method also causes that the more a word occurs in the corpus, the more times it would be reconstructed, the more contributions it would make to the reconstruction error. Because URAE optimizes weights by minimizing the reconstruction error, the more time a word occurs, the more effect it would have on the model weights, and it would be reconstructed better. It would happen that stopwords like ‘the’, ‘a’ and etc. have the same effect or even more effect than the others while representing phrases. But different words affect semantic meaning of phrases in different degrees. So, we propose that reconstruction error of every leaf nodes should be weighted by the reciprocal of its frequency:

$$E_{rec}(y_{(i,j)}) = \sum_{k=i}^j \frac{1}{count(w_k)} \cdot \|w_k - w'_k\|^2 \quad (4)$$

where  $count(w_k)$  means the count of the word in the corpus.

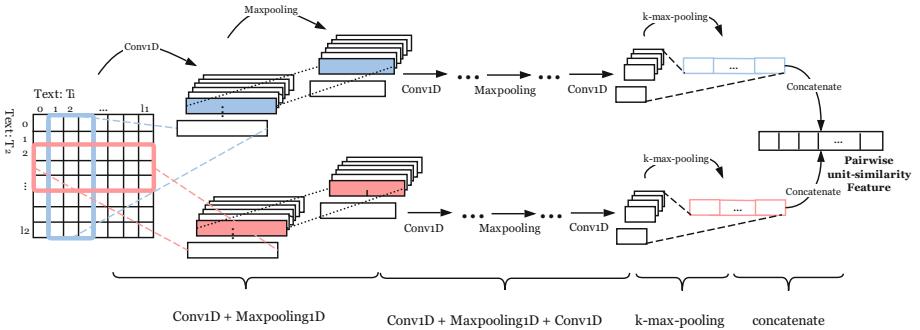
**WURAE Training.** A large set of sentences is used to train this unsupervised learning algorithm. The model minimizes the sum of all node’s reconstruction errors in a mini-batch. It uses backward propagation through structure [5] to compute the gradient and optimizes with L-BFGS in the mini-batch training.

After learning phrase and sentence embedding, we can get the sentence represented as  $(w_1, w_2, \dots, w_n, w_{n+1}, \dots, w_{2n-1})$ , where  $(w_1, \dots, w_n)$  are word vectors,  $(w_{n+1}, \dots, w_{2n-2})$  are phrase vectors and  $w_{2n-1}$  is the sentence vector.

### 3.3 Pairwise Unit Similarity from CNN

In order to find out whether the given pair of text convey same meaning, we take the similarity of basic units into account. Words and phrases are regarded as the units of sentence. By analogy of sentence and article, sentences are considered as the basic units of article. For the given pair of  $l_1$ -length text  $T_1$  and  $l_2$ -length text  $T_2$ ,  $T_1$  and  $T_2$  are represented by the unit embedding lists, like  $S$  mentioned above. To extract basic unit similarity feature, we firstly compute similarity matrix via the unit embedding lists of the pairwise text. The similar or same pair of units, matched from  $T_1$  and  $T_2$ , might appear in different positions of the two lists. Thus, we need to compare every unit vector in one text with all the unit vectors in another one. A similarity matrix with the size of  $l_1 \times l_2$  is constructed by calculating cosine distance between the matched-pairs of units.

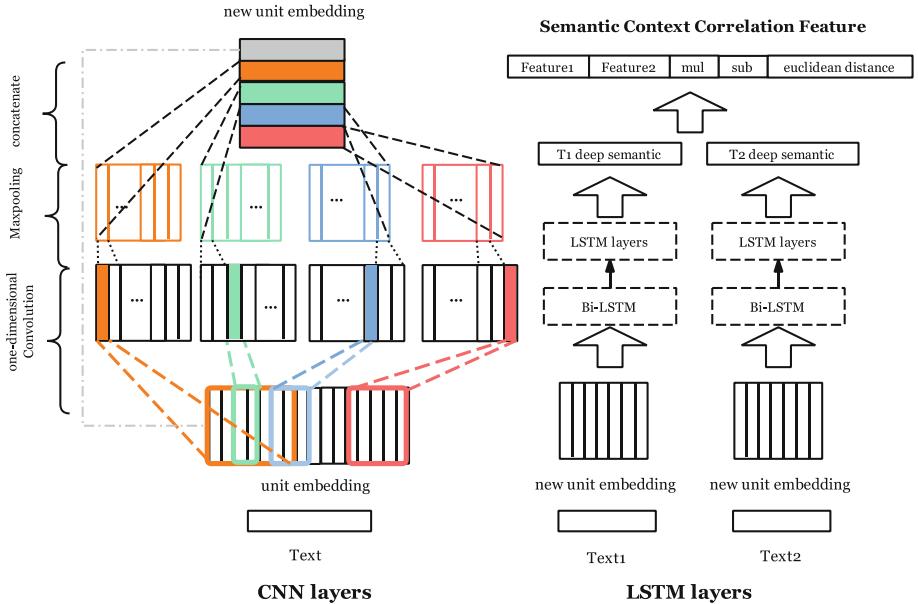
Convolution neural network is used to learn the patterns of pairwise semantic resemblance. The architecture of pairwise unit similarity measurement is as described in Fig. 1. The model consists of 3 convolution layers and the former two convolution layers are both followed by a max-pooling layer. The output of the third convolution layer is fed into a k-max-pooling layer, which extracts the top  $k$  most important features and gets the result of a flattened feature  $F_{\text{unit\_similarity}}$ . Due to the dissymmetry of two text, the pairwise unit resemblance is calculated through two directions of the input similarity matrix. Then the pairwise unit similarity feature could be obtained by concatenating the two features,  $F_{\text{unit\_similarity}_1}$  and  $F_{\text{unit\_similarity}_2}$ .



**Fig. 1.** The architecture of pairwise unit similarity measurement

### 3.4 Semantic Context Correlation from CNN and LSTM

Different arrangements of words and phrases express various semantics in the sentences. Also, various sequences of sentences make the meaning of articles different. So, besides pairwise unit similarity feature, we also regard semantic correlation among the units as an important feature. In this part, a combination of CNN and LSTM is used to get semantic context correlation feature as depicted in the Fig. 2. The input of this model is a list of basic unit embedding which represents the text.



**Fig. 2.** The architecture of semantic context correlation similarity measurement

Firstly, CNN is utilized to get the collocation information between adjacent units in context, like n-gram feature in sentences. The model uses 4 one-dimensional convolutional layers with window sizes of 2, 3, 5 and 7 to get features from embedding list, resembling 2-gram, 3-gram, 5-gram and 7-gram feature. The input of embedding list is fed into the four different convolutional layers respectively. New unit embedding is constructed by concatenating these 4 results and the original unit embedding list. For the given input of m-dimensional embedding, the new unit embedding would have the dimension of  $5 \times m$ .

The new unit embedding is fed into LSTM so as to learn long-term dependencies from sequential units of text. Here a bidirectional LSTM performs both forward pass and backward pass on the new unit embedding matrix. Then, other two LSTM layers learn more from its output. The last hidden state is taken as deep semantic feature of the input text.

For the given pair of texts, each one is fed into the model separately to get its own deep semantic feature  $F_{semantic}$ . And then the pairwise deep semantic features ( $F_{semantic1}, F_{semantic2}$ ) generate the semantic context correlation feature:

$$F_{sub\_sem} = F_{semantic1} - F_{semantic2} \quad (5)$$

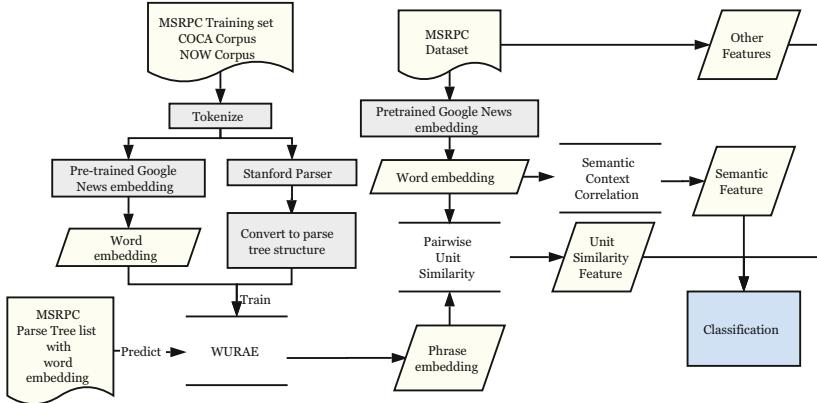
$$F_{mul\_sem} = F_{semantic1} \cdot * F_{semantic2} \quad (6)$$

$$F_{euclidean\_sem} = F_{sub\_sem} \cdot * F_{sub\_sem} \quad (7)$$

where semantic context correlation feature equals to the concatenate of those features,  $[F_{semantic1}; F_{semantic2}; F_{mul\_sem}; F_{sub\_sem}; F_{euclidean\_sem}]$ .

### 3.5 Paraphrase Identification on Pairwise Sentence

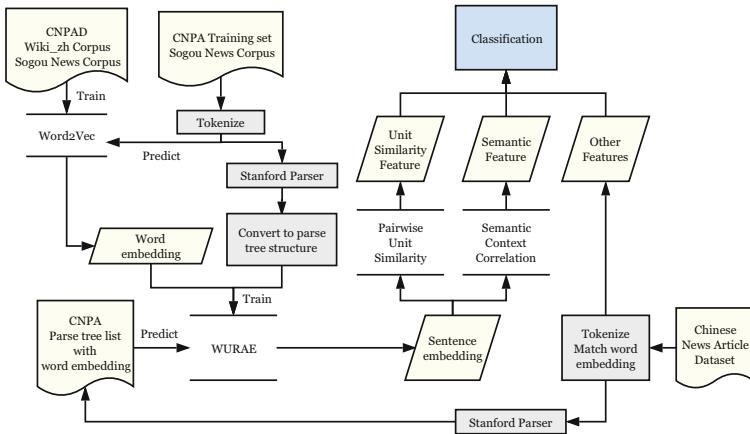
MSRPC, an English paraphrase sentences corpus, is used in the sentence-level PI task. The entire architecture is shown in Fig. 3. Firstly, WURAE is trained in a large scale of English news sentences and then it calculates the phrase embedding of sentence. The sentence is represented by pre-trained word embedding and phrase embedding. For the purpose of classification, we extract its pairwise word & phrase similarity feature and semantic context correlation feature from models proposed in Sects. 3.3 and 3.4. Moreover, other features are added to the model, including number feature, BLEU score, ratio of Longest Common Subsequence (LCS), ratio of edit distance and similarity based on TF-IDF.



**Fig. 3.** The overall architecture of sentence-level paraphrase identification

### 3.6 Paraphrase Identification on Pairwise Article

The overall methodology of article-level PI task is depicted in Fig. 4. A dataset of Chinese news paraphrase article (CNPA) is used in this task. Chinese word embedding is trained by Word2Vec. Then, WURAE is trained in a large scale of Chinese sports & entertainment news sentences. Sentence embedding, calculated by WURAE, represents the articles. For classification, pairwise sentence similarity feature and semantic context correlation feature are extracted from the given pair of articles through the models mentioned above. To get better performance, number feature is also added to this method. The probability of being paraphrased is predicted by the combination of features.



**Fig. 4.** The overall architecture of article-level paraphrase identification

## 4 Experiments

### 4.1 Datasets and Settings

**MSRPC.** In our sentence-level PI task, we use the benchmark Microsoft Research Paraphrase Corpus. The length of sentences in this corpus ranges from 7 to 35 and 67% of the pairs are paraphrased. The origin train set has 4,076 pairs and we split it into train set and validation set with the ratio of 9 to 1. And the origin test set has 1,725 pairs of sentences. Owing to the asymmetry of two sentences, we expand the dataset by exchanging position of two sentences in one pair. As is mentioned above, a 300-dimensional English word embedding of Google News vectors is applied in this task.

**Chinese News Paraphrase Article Dataset (CNPA).** An article paraphrase corpus of Chinese sports & entertainment news is used in our article-level PI task. Non-paraphrase pairs are constructed in this corpus by randomly matching articles from different paraphrase pairs. We further introduce comparison on length and TF-IDF to prevent negative pairs from differing too much. The length of articles, which means the number of its sentences, varies from 10 and 55. We split the dataset into train set, validation set and test set, as shown in Table 1. The Chinese word embedding is trained from Word2Vec with Chinese Wiki data and Sogou News data in the dimension of 300.

**Table 1.** Statistics of Chinese news paraphrase article dataset

Set	Article pairs	Paraphrase	Non-paraphrase
Train	10191	5721	4470
Val	2909	1633	1276
Test	1455	817	638

**Settings.** The hyperparameters are tuned on the validation set of MSRPC. The settings of English sentence-level PI experiment are chosen as Adadelta optimizer, learning rate of 0.175, dropout rate of 0.1 and mini-batch size of 50. We adjusted the mini-batch size of Chinese article-level PI experiment to 64. The size of k-max pooling in pairwise unit similarity measurement is separately 15 for word-level or sentence-level unit and 17 for phrase-level unit.

## 4.2 Distributed Representation of Phrase and Sentence

WURAE is trained in the mini-batch of the sentences from a large scale of English and Chinese corpus. The English corpus is constructed by COCA (Corpus of Contemporary American English), NOW (News on the Web)<sup>4</sup> and MSRPC train set, which have 80,697 sentences. The Chinese corpus is composed of Sogo News data and sentences in train set, which has 421,293 sentences. To get the parse tree, we preprocessed the corpus by Stanford Parser. Based on WURAE, the phrase and sentence embedding is learned from the parse tree with the initial word embedding.

**Table 2.** Performance of different features

Model	Accuracy	F1-score	Model	Accuracy	F1-score
WE + Pairwise Similarity	71.94%	79.13%	With Other Features	75.01%	82.23%
WE + Semantic Context Correlation	71.42%	80.73%		73.80%	81.58%
WE + PE + Pairwise Similarity	72.70%	81.35%		75.48%	82.38%
WE + PE + Pairwise Sim + Context Corr	73.91%	81.99%		76.70%	83.44%

## 4.3 Results

**MSRPC.** Firstly, we test performance of separate and combined features, shown in Table 2. We can find that phrase embedding improve the performance of pairwise similarity by 0.76% on accuracy and 2.22% on F1-score. A performance of 71.42% is obtained from semantic context correlation. Our entire sentence-level paraphrase identification model gained the accuracy of 76.70% and F1-score of 83.44%.

We also compare our methodology with lots of state-of-art methods. The comparison is shown in Table 3. Our method achieves a competitive result compared with the

---

<sup>4</sup> <https://corpus.byu.edu/>.

**Table 3.** Experimental results of english sentence-level paraphrase detection

Method		Open resources	Acc	F1-score
All paraphrase (Baseline)			66.5%	79.9%
Hu et al. [7]	Convolutional Matching Model	Project homepage	69.9%	80.91%
Socher et al. [18]	URAE with Dynamic Pooling	Pre-trained phrase vector data, PI code	76.8%	83.6%
Madnani et al. [14]	8 Machine Translation Metrics	Error analysis data	77.4%	84.1%
Pang et al. [17]	Text Matching via CNN		75.94%	83.01%
El-Sayed et al. [3]	Similarity & Abductive Network		73.91%	81.25%
Eyecioglu et al. [4]	Character-Based Features		74.2%	82.7%
Our Work	WURAE with K-Max, CNN and LSTM	WURAE, PI code, pre-trained data <sup>5</sup>	76.70%	83.44%

existing methods. It shows that deep semantic features of sentences could be extracted by the combination of WURAE, pairwise similarity and context correlation method.

**CNPAD.** This dataset is experimented on both separate and combined features, shown in Table 4, where SE means sentence embedding. The pairwise sentence similarity gets the accuracy of 96.15% and semantic context correlation gets the accuracy of 96.91%. Through combination of those two methods, we could get an improvement of 0.55% on accuracy. And the overall architecture obtains the accuracy of 99.31% and F1-score of 99.39%. We can see that the combination of sentence embedding, pairwise similarity and semantic context correlation do capture the deep semantic feature of articles.

**Table 4.** Experimental result of Chinese article-level paraphrase detection

Method	Accuracy	F1-score
All paraphrase (Baseline)	56.15%	71.92%
SE + Pairwise Sentence Similarity	96.15%	96.57%
SE + Semantic Context Correlation	96.91%	97.23%
SE + Pairwise Similarity + Context Correlation	97.46%	97.72%
SE + Pairwise Similarity + Context Correlation + Number feature	99.31%	99.39%

<sup>5</sup> [https://github.com/SannyZhou/WURAE\\_Paraphrase\\_Identification\\_CNN\\_LSTM](https://github.com/SannyZhou/WURAE_Paraphrase_Identification_CNN_LSTM).

## 5 Conclusion

In this paper, we proposed a method of sentence-level paraphrase identification and introduced an article-level paraphrase identification method by analogy. Also, Weighted Unfolding RAE, an unsupervised learning algorithm, is proposed for learning phrase and sentence embedding. In the sentence-level PI task, words and phrases embedding represents the sentences while the articles are represented by sentence embedding in the article-level PI task. Pairwise unit similarity feature is captured from unit similarity matrix through CNN and k-max pooling layers. After getting region information from sequences by multiple CNN layers with different window sizes, the model implements LSTM to learn the long-term dependency of text. The experimental results prove that our methodology could capture deep semantic feature and perform well in paraphrase identification. It also shows that we can get better semantic feature with the distributed representation of phrases and sentences based on WURAE. In the future, we could build an open domain Chinese paraphrase corpus. Also, we would adjust our paraphrase identification method and our algorithm of phrase & sentence embedding in different NLP applications, such as question answering, information retrieval, text classification, etc.

**Acknowledgements.** This research work has been funded by the National Natural Science Foundation of China (Grant No. 61772337, U1736207 and 61472248), the SJTU-Shanghai Songcheng Content Analysis Joint Lab, and program of Shanghai Technology Research Leader (Grant No. 16XD1424400).

## References

1. Agarwal, B., Ramampiaro, H., Langseth, H., Ruocco, M.: A deep network model for paraphrase detection in short text messages. arXiv preprint [arXiv:1712.02820](https://arxiv.org/abs/1712.02820) (2017)
2. Chitra, A., Kumar, S.: Paraphrase identification using machine learning techniques. In: Proceedings of the 12th International Conference on Networking, VLSI and Signal Processing, pp. 245–249 (2010)
3. El-Alfy, E.S.M., Abdel-Aal, R.E., Al-Khatib, W.G., Alvi, F.: Boosting paraphrase detection through textual similarity metrics with abductive networks. *Appl. Soft Comput.* **26**, 444–453 (2015)
4. Eyecioglu, A., Keller, B.: Knowledge-lean paraphrase identification using character-based features. In: Filchenkov, A., Pivovarova, L., Žižka, J. (eds.) AINL 2017. CCIS, vol. 789, pp. 257–276. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-71746-3\\_21](https://doi.org/10.1007/978-3-319-71746-3_21)
5. Goller, C., Kuchler, A.: Learning task-dependent distributed representations by backpropagation through structure. In: IEEE International Conference on Neural Networks, vol. 1, pp. 347–352. IEEE (1996)
6. He, H., Lin, J.: Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 937–948 (2016)

7. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural Information Processing Systems, pp. 2042–2050 (2014)
8. Ji, Y., Eisenstein, J.: Discriminative improvements to distributional sentence similarity. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 891–896 (2013)
9. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint [arXiv:1404.2188](https://arxiv.org/abs/1404.2188) (2014)
10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint [arXiv: 1408.5882](https://arxiv.org/abs/1408.5882) (2014)
11. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in Neural Information Processing Systems, pp. 3294–3302 (2015)
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
13. Lin, Z., et al.: A structured self-attentive sentence embedding. arXiv preprint [arXiv:1703.03130](https://arxiv.org/abs/1703.03130) (2017)
14. Madnani, N., Tetreault, J., Chodorow, M.: Re-examining machine translation metrics for paraphrase identification. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 182–190. Association for Computational Linguistics (2012)
15. Mahajan, R.S., Zaveri, M.A.: Machine learning based paraphrase identification system using lexical syntactic features. In: 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–5. IEEE (2016)
16. Mihalcea, R., Corley, C., Strapparava, C., et al.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, vol. 6, pp. 775–780 (2006)
17. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: AAAI, pp. 2793–2799 (2016)
18. Socher, R., Huang, E.H., Pennin, J., Manning, C.D., Ng, A.Y.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Advances in Neural Information Processing Systems, pp. 801–809 (2011)