Which Embedding Level is Better for Semantic Representation? An Empirical Research on Chinese Phrases

Kunyuan Pang, Jintao Tang(✉), and Ting Wang

College of Computer, National University of Defense Technology Changsha, Hunan 410073, People's Republic of China
{pangkunyuan,tangjintao,tingwang}@nudt.edu.cn

Abstract. Word embeddings have been used as popular features in various Natural Language Processing(NLP) tasks. To overcome the coverage problem of statistics, compositional model is proposed, which embeds basic units of a language, and compose structures of higher hierarchy, like idiom, phrase, and named entity. In that case, selecting the right level of basic-unit embedding to represent semantics of higher hierarchy unit is crucial. This paper investigates this problem by Chinese phrase representation task, in which language characters and words are viewed as basic units. We define a phrase representation evaluation tasks by utilizing Wikipedia. We propose four intuitionistic composing methods from basic embedding to higher level representation, and investigate the performance of the two basic units. Empirical results show that with all composing methods, word embedding out performs character embedding on both tasks, which indicates that word level is more suitable for composing semantic representation.

Keywords: Word embedding · Phrase representation
Composing model

1 Introduction

Word embeddings have been working as popular features in nearly every NLP task like named entity recognition [15], similarity measurement [8,10], machine translation [3,14], etc. Popular embeddings methods such as skip-gram, CBOW [8], and Glove [11] adhere to the *Distributional Hypothesis* [5]. They first generate a token list from a specific vocabulary of a language, and then calculate embeddings for each token with token cooccurrence information.

However, limited by the vocabulary and resource, embeddings do not cover all language phenomenon, like idioms, named entities and phrases.

The research is supported by the National Key Research and Development Program of China (2018YFB1004502) and the National Natural Science Foundation of China (61472436, 61532001).

© Springer Nature Switzerland AG 2018
M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 54–66, 2018.
https://doi.org/10.1007/978-3-319-99501-4_5

Table 1. Semantics of a Chinese phrase 大学城北站

Segmentation	Semantics Candidate	Evaluation
大学城北站	Higher Education Mega Center North Station	precise, but hard to include in vocabulary
大学城 | 北 || 站	advanced education district| north part/north of|| station	high word semantic selection, but less generative / potential segmentation error
大学 || 城 || 北站	university|| city|| north (key) station	segmentation error
大 | 学 | 城 | 北 | 站	big/advanced| mimic/knowledge/school| city/town/center| north part/north of/defeat| station/stand/stay	flexible in generation, but complex combinations to choose from

[8] generate embeddings for phrases in a statistical way. Frequent bigrams in corpus are viewed as *idiomatic phrases*. These frequent bigrams are recorded as new tokens and participate in embeddings with other words. This method partially breaks from vocabulary restrictions, but is still restricted to the corpus. Named entities, for example that come into existence after the corpus are not embedded.

Hierarchical structure of language makes it possible to form composed entities and phrases with basic units. Based on this idea, compositional models [18] embed basic units and compose into higher hierarchy structures. This overcomes the coverage problem in both vocabulary and corpus.

For languages like Chinese, token definition is also a problem in embeddings, in which vocabulary can be built on characters or words. Embeddings of characters and words are semantically different, and this difference affects the semantic representing ability of compositional model. Generally, a character embedding is an average of more senses while word embeddings are more specific. Whether with characters or with words is an important question in composing models. As is shown in Table 1, a location entity is composed with different units. The quality of compositional semantic representation is largely decided by composing unit selection. Consequently, which embedding level is better for semantic representing becomes an important research question.

To be specific, We evaluate the quality of semantic representation with (1) measuring the distance between composed embedding and trained embedding for Wikipedia titles, and (2) comparing semantic similarity of phrase embeddings against Wikipedia redirection. Under this evalutaion measure, we use three intuitionistic composing methods, component average(CA), neighbor average(NA) and neighbor cluster average(NCA) and investigate performance of these models on word embeddings and character embeddings.

Results of experiments on this task show that with each composing model, word embeddings outperform character embeddings, which suggests in semantic analyzation tasks, word embeddings might be more suitable.

This paper is organized as follows. Section 2 summarizes related works, especially on how to generate word embeddings, how to use them and how to ensemble component embedding into greater parts embedding. Section 3 illustrates our methods, from embedding short words and characters to calculating long words and phrases. Section 4 shows experiments on embedding error and on Wikipedia redirection prediction tasks. Section 5 summarizes our model and discusses remaining research points.

2 Related Works

Word embedding can be categorized into 3 classes, which are language model based, task based, and direct generation.

Both language model based and direct generation models follow *Distributional Hypothesis* [5], which states that *words with same contexts tend to have similar meanings.* In different models, this hypothesis is implemented with different optimization objectives. In NNLM [1], a 3-layer neural network is used to estimate $P(w_i|w_{i-(n-1)}, ..., w_{i-1})$, where embeddings of $(w_{i-(n-1)}, ..., w_i)$ serve and get trained as the first layer.

CBOW and skip-gram [8] try to generate word embedding with a simple task. The task is to predict the word with context words or reversely. CBOW predicts by dot production current word and the average of embeddings in context window. Skip-gram is similar. GloVe [11] records co-occurrence of words and suppose words co-occurrence frequency ratio as similarity ratio. Ideally GloVe and skip-gram converge to the same embedding if an optimum embedding exists.

Task based embedding solve supervised tasks with neural network. The embedding layer can be generalized to other tasks and serve as word embedding. Fasttext [6], for example, use a shallow neural network for text classification. The first layer is taken as word embedding.

All word embeddings mentioned above claim to represent syntactic and semantic embedding. These trained embeddings are released for use in other tasks.

Models are put forward to improve word embedding or solve problems in application with hierarchical structure of language, especially in dealing with OOV(out of vocabulary) words. [12] model words with a convolutional network over its characters. Character patterns in English is believed to be strong in syntactic and the combination of characters can make up any word. [?] used Byte-Pair Encoding(BPE) to control vocabulary size in machine translation. Words are first encoded with character pairs before fed into neural machine translator. [17] built a recursive neural network on its syntactic tree to encode a sentence. When encountering an OOV word, the recursive network is formed in a primitive left-to-right way on its sub-word parts. These sub-word parts are found by BPE and serve as leaf embedding nodes in the syntactic tree.

In Chinese, most researchers use word embedding as is parallel in English. Still, character embedding is also used, especially in word segmentation [20] and text classification [19]. Characters can also enrich word embeddings or even be

divided into sub-character parts. CWE [2] modifies CBOW, Skip-gram and GloVe by adding character embedding average into context vector, and improve the quality of target word embedding. [13] breaks Chinese characters into radicals. They use CBOW for radical embedding, and feed radical embedding to different neural networks for short text classification, Chinese word segmentation and web search ranking.

[9] work on composing phrase with its components on early vector based models of word meaning. They designed additive model and multiplication model. Both models are enlightening for composing on word embedding.

3 Methodology

This section illustrates the framework of word embedding generation and phrase embedding composing.

3.1 Problem Definition

With a given segmented corpus $\mathbf{D}$ and an embedding method, sets of chars $\mathbf{S_C}$, words $\mathbf{S_W}$ and phrases $\mathbf{S_P}$ are defined. To avoid repetitive description, we denote these language units **tokens**, $\mathbf{S_T} = S_C \cup S_W \cup S_P$. Embedding methods give every token t a vector representation $\boldsymbol{e}(t)$. We research on long tokens $t = (c_1 c_2 ... c_n)$ where $c_1, c_2, ..., c_n \in S_T$. These substrings $c_1, ..., c_n$ are called **components**. The aim of this research is to compose estimation of $\boldsymbol{e}(t)$ with $\boldsymbol{e}(c_1), ..., \boldsymbol{e}(c_n)$, and make the composed estimation $\boldsymbol{e}_{\mathbf{comp}}(\mathbf{t})$ as close to trained $\boldsymbol{e}(t)$ as possible if $t \in S_T$.

3.2 Modified CBOW

CBOW is used to generate word embedding as a basis of phrase embedding composing. In order to capture composability from Chinese characters to words, a modification is made.

The original version of CBOW works on Chinese words as follows. First, sentences in the corpus are segmented into words with segmentation algorithm. Secondly, words with frequency above the threshold are selected as tokens and form token list. Finally CBOW iterates on the corpus several times, on each word in token list with objective described in Eq. 1. $\boldsymbol{e}(word)$ and $\boldsymbol{e}'(word)$ are two sets of embeddings, and $\boldsymbol{e}_{context}$ is the average vecter of context words.

$$Cost = \sum_{(word,context)\in D} \frac{\exp(\boldsymbol{e}'(word)\boldsymbol{e}_{context})}{\sum_{word'\in V} \exp(\boldsymbol{e}'(word')\boldsymbol{e}_{context})} \tag{1}$$

In order to compare estimating precision of different composition methods, embedding of characters, words and phrases are needed. CBOW can produce phrase embedding by including them in user segmentation dictionary and segmenting by large-grain. Thus, those phrases in the corpus are included in the token list and get an embedding.

A large majority of characters are also included in the token list, but they are not character embeddings. Single characters appear in word-level token list for two reasons. It is a single-character word, or it is left because of segmentation error. A Chinese character has a lot more senses than words. Senses of a character as an independent word are usually different from those composing other words. Embedding of single-character words in CBOW is thus infeasible in character to word/phrase composition.

An option is to train CBOW on character level separately and generate character embedding with composing senses. However, the alignment between character embedding space and word embedding space takes extra effort, and existing research of alignment is not satisfying in Chinese character to word alignment task. CWE produce character embedding and word embedding at the same time, too, but that character embedding is an additive part in context, which is not in word embedding space either.

We modify CBOW by randomly replacing a word as a character composing it in each iteration. With enough iteration, the character embedding is a combination of single-character word sense and word composing sense. Experiments show that this modification dose not harm CBOW in its ability to embed semantic information of words. Character sense is improved since when looking for similar words of a character, more words that it composes are recalled.

3.3 Trivia Combination Model

In the discussion to follow, we discuss methods to calculate representation of tokens via components of lower level linguistic units.

CA (Component Average) is a trivia model of estimation is component average. Let T be the component sepuence of a token t.

$$\boldsymbol{e}_{comp}(t) = \frac{1}{|T|} \sum_{c \in T} \boldsymbol{e}(c) \tag{2}$$

NA (Neighbor Average) explores more information of the components by finding m most similar words in the embedding space and then calculates average of these neighbors. Let N be the set of neighbours, i.e. $N = \{q | rank(q, c_i) < m, c_i \in T\}$

$$\boldsymbol{e}_{comp}(t) = \frac{1}{|N|} \sum_{n \in N} \boldsymbol{e}(n) \tag{3}$$

3.4 Neighbor Cluster Average

NCA is based on the following hypothesis in composing tokens from components:
Sense Activation Hypothesis: A component as words or characters has several senses. When composing high level structures, i.e. tokens or sentences, one sense of the component is activated. Activated senses of components compose semantic meaning of tokens and sentences.

In the embedding point of view, this hypothesis means that component embedding is an average of its sense embeddings. This is in accordance with *Distributional Hypothesis* and the process of CBOW. Each sense of a component can be represented by a distribution of context words. Training on the corpus by CBOW is training the embedding of a component by a superposition of its sense context distributions and results in an average of senses.

It is observed from word similarity task that similar words, or neighbors of a word requires interpretation from different senses of a word, concluding that more information of different senses can be recovered from word neighors.

NCA model aims to discover combination of senses by clustering. Neighbors of each component are retrieved with a large window, ensuring that as many senses represented by neighbors are included as possible. Since the embedding model views cooccurance as similarity. If two senses of two components are likely to be acitvated in the same token, their corresponding neighbour clusters should be close in the embedding space and forms one cluster when clusting all neighbors of all components. Selecting the largest cluster is thus selecting the most likely average of combined senses. Let C_{max} be the largest cluster over all $n \in N$ as defined in 3.3

$$\boldsymbol{e}_{comp}(t) = \frac{1}{|C_{max}|} \sum_{n \in C_{max}} \boldsymbol{e}(n) \tag{4}$$

We use k-means cluster algorithm. k is set to size of components. We take the centroid of the largest cluster as the representation of the token.

3.5 Self Attention Model

Self attention model(denoted as **ATTN**) follows attention machenism preopsed in [16].

$$\boldsymbol{e}_{comp}(t) = \frac{1}{|T|} \sum_{c_i \in T} \alpha(c_i) \boldsymbol{e}(c_i) \tag{5}$$

$$\alpha(c_i) = tanh(w * cos(\boldsymbol{e}_{context}, \boldsymbol{e}(c_i)) + b_p) \tag{6}$$

$$\boldsymbol{e}_{context} = \frac{1}{|T| - 1} \sum_{c_j \in T, j \neq i} \boldsymbol{e}(c_j) \tag{7}$$

The importance or weight of a component c_i is estimated as coherence with other components in the token. w is used for controlling weight ratio of high relavance to low relavance. $b_p \in \{begin, middle, end\}$ is the position bias of weight encoding positional information of a component. The cost to train w and b_p is: every combosed estimation embedding is close to its trained embedding.

$$Cost = -\sum_{t \in S_T} cos(\boldsymbol{e}_{comp}(t), \boldsymbol{e}(t)) \tag{8}$$

4 Experiments

To find the right level of representing phrase semantics, we compare segmenting phrase into words and characters. We also experiment on composing words with characters to show composability of characters. Absolute Embedding Error compares the precision of composing compared to standard embedding trained by CBOW.

4.1 Experiment Settings

Corpus and Dictionary. We use modified CBOW on cleaned Chinese Wikipedia corpus. We extracted 1GB pure text from dumped wiki-pages. Jieba[1] is used to segment the pure text. Wikipedia title list is added as user dictionary to ensure that we retrieve enough phrases and train their embedding for comparison.

Embedding Algorithm and Parameters. We run our modified CBOW on the text. Replace ratio is 0.1, and iteration is set to 20 times, larger than usual to ensure replacement balance. The embedding dimension is 60 and minimum occurrence of a token is set to 3.

Character, Word and Phrase Selection. The identification of characters, words and phrases is by length. We take into consideration only tokens purely consist of Chinese characters.

We select tokens with a length of 1 as characters, 2–3 as words and longer than 5 as phrases. This selection is based on reasons that follows. First of all, it is hard to separate characters and single-character words. Thanks to our modification over CBOW, embedding of tokens with the surface form of single characters always contain semantic information as characters.

According to [7], Chinese linguists listed the most frequently used words. Among 56008 of them, only 162 are of 5 characters and above and most of them have an inner structure of shorter words. We are confident that these long tokens are phrases.

2–3 characters long tokens are words without doubt. Words with 4 characters are a mixture with independent words, short phrases, and a lot of Chinese traditional idioms of weak composability. As a result, we end up with 10386 characters, 118348 words and 49878 phrases for composing experiments.

Compose Levels. Table 2 shows all 4 composing levels that we test on. Composing from characters to words is also included, in case the number of components affects composing quality.

Though our separation and composing method is abrupt and simple. It separates composing from words and composing from characters well. Any other

[1] http://www.oss.io/p/fxsjy/jieba.

Table 2. Illustration of different composing levels

Level	Component Selection	example(广州市大学城北站)
p_w	non-overlapping words found in the phrase with forward maximum matching	广州市 | 大学城北
p_c	all characters in the phrase	广 | 州 | 市 | 大 | 学 |·····
p_l	words in p_w and characters **not** in p_w	广州市 | 大学城北 | 站
w_c	all characters in a **word**	广 | 州 | 市

given composing divides into word-to-phrase and character-to-phrase composing patterns.

4.2 Relative Composing Precision

Relative Composing Precision experiment compares the composed phrase embedding with trained phrase embedding. Formally, this precision is defined as Eq. 9. $\boldsymbol{e}(n)$ is the embedding of the most similar token in S_T.

$$RCP = cos(\boldsymbol{e}_{comp}(t), \boldsymbol{e}(t)) - cos(\boldsymbol{e}(n), \boldsymbol{e}(t)) \tag{9}$$

Note that $\boldsymbol{e}_{comp}(t)$ is the only variable for a given sample token. The reason not using bit-wise L2-loss is, in CBOW similiarity is valued by cosine. The norm of token vecters is not 1. $|\boldsymbol{e}_{comp}(t) - \boldsymbol{e}(t)|^2$ can still be large even if we get the exact meaning. It is acceptable that our composed embedding is synonym of the original phrase. The reason for adding reference score $cos(\boldsymbol{e}(n), \boldsymbol{e}(t))$ is to align samples at different composing difficulties. For tokens that lies in dense parts of embedding space, the error is penalised by likely higher reference score.

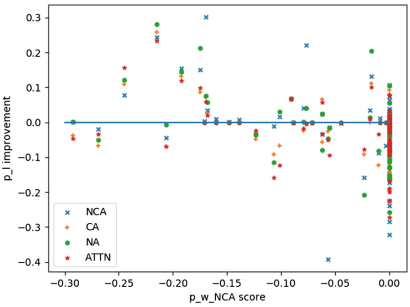
Table 3. Relative composing precision on different levels and methods

	CA			ATTN			NA			NCA		
	p_w	p_c	p_l	p_w	p_c	p_l	p_w	p_c	p_l	p_w	p_c	p_l
Mean RCP	−0.28	−0.56	−0.30	−0.28	−0.56	−0.29	−0.21	−0.46	−0.22	**−0.20**	−0.43	−0.21
Best sample	0.01	−0.13	0.01	0.02	−0.12	0.02	**0.06**	−0.08	**0.06**	**0.06**	−0.05	0.04
RCP @75%	−0.14	−0.47	−0.17	−0.13	−0.45	−0.15	**−0.08**	−0.35	−0.09	−0.09	−0.31	−0.10
RCP @50%	−0.26	−0.56	−0.28	−0.24	−0.55	−0.26	−0.17	−0.44	−0.20	**−0.16**	−0.40	−0.19
RCP @25%	−0.36	−0.65	−0.40	−0.39	−0.67	−0.42	−0.30	−0.58	−0.30	**−0.28**	−0.57	−0.29
Worst sample	−0.75	0.93	−0.80	−0.75	−1.00	−0.80	−0.68	−1.00	−0.71	−0.70	−0.95	**−0.64**

We compare composing methods in Table 3. On each composing level, NCA always achieves the best performance. On p_w, NCA advantage over NA exists

in the low-quality cases. Attention model improves at most 0.05 points over CA model. The improvement of introducing neighbor for more information is significant, as on each level, NA and NCA are a lot better than CA.

Comparing between p_w level and p_c level. Even the worst model for p_w is better than p_c. This comparison shows the importance of level selection. Pure character level is not suitable for semantic composing task.



	CA	NA	NCA	ATTN
average affect	-0.02	-0.01	-0.02	-0.02
p_w leads by	**0.03**	**0.03**	**0.04**	**0.03**
p_l leads by	0.01	0.01	0.02	0.01

Fig. 1. Comparison between p_w and p_l

Improvements of information from left characters is not significant. Table 3 shows the result of introducing leftover characters for more information(level p_l). We also scatter sample points in Fig. 1 to show p_l improvments over p_w results when the original p_w scores differently. Overall, p_l result is better when p_w score is small, but becomes noise when p_w socre is large.

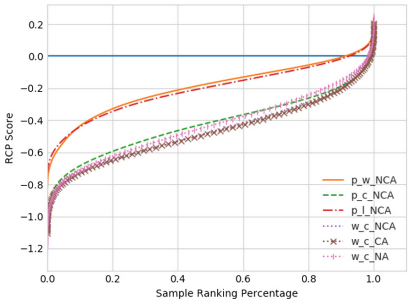


Fig. 2. w_c Results

Figure 2 shows the performance of composing words with characters. A potential reason why p_c performs badly is segments the phrase into too many components for the model to process. w_c has 2–3 components and is similar to p_w segmentation. If character embedding were also a good level of composing, w_c should achieve similar performance as p_w. However, as is shown in the figure,

w_c performs similar to p_c level. The reason lies in characters or character embedding, but not in component number.

These experiments show that word alone is the only level that compose the embedding of phrase with low difference. Character level is not only unsuitable itself, but also bring noise when integrating with words.

4.3 Wikipedia Redirections Prediction

The most direct semantic task is word similarity test like word-sim 393 in English and wordsim245 and wordsim297 in Chinese. However, wordsim245 and wordsim297 contains very few phrases and we have to compose our own semantic similarity task.

We compose phrase-word semantic similarity task by utilizing Wikipedia redirections. Redirections in Wikipedia are paraphrases of the same thing or closely related things noted by Wikipedia editors. A pair of redirections are thus semantically identical or very close.

We construct positive set by finding all redirections with at least one embedded phrase. Negative set is constructed by sampling a pair of words and phrases from positive set, making sure that the pair is not in positive set. The size of positive and negative set each is 426.

We use word similarity directly for this classification task and adopt AUC [4] to examine precision of similarity without setting threshold manually.

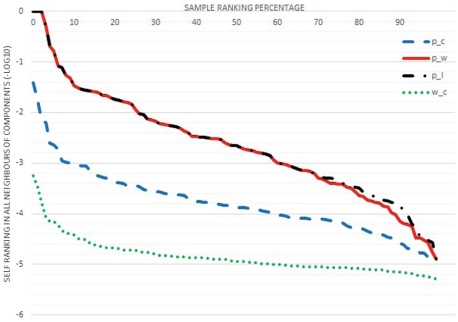
Table 4. AUC of Wikipedia redirection prediction

	Composing level	
	p_w	p_c
CA	0.9485	0.7845
NA	0.9336	0.6841
NCA	0.9295	0.7037

Trained value of words and phrases are used as standard reference value. Composed embedding of phrases are used for each test case and The AUC values are shown in Table 4. High AUC of standard reference show that our embedding and cosine similarity is a good feature for the task. With each method, p_w is a lot better than p_c. This again proves word is the only right level to compose semantic embedding of phrases.

4.4 Case Study: Difference in Component Quality

Opposite from common sense that if we understand all characters in a word well, we will understand the word. Composing words and phrases from characters is impossible. We explain this phenomenon with descriptive experiments.



	nearest neighbor		
	p_w	p_c	p_l
mean RCP	-0.0364	-0.0894	-0.0312
best sample	0.0000	-0.0000	0.0000
RCP @75%	0.0000	-0.0163	0.0000
RCP @50%	0.0000	-0.0824	0.0000
RCP @25%	-0.0468	-0.1304	-0.0284
worst sample	-0.2924	0.3693	-0.2924

Fig. 3. Difference in component quality

Figure 3 left illustrates the most 'precise' neighbor we retrieved in NA and NCA at different levels.[2] This precise neighbor is useless in models because we need the standard answer to identify its precision, and the composing model is mostly sorting this neighbor by all information. Still, it helps to illustrate quality of our components. It is shown that half p_w components include synonym of the target token in neighbor set, while character levels finds only close words.

We also try to retrieve the target phrase as most similar word of its components. As Fig. 3 shows, to achieve a 50% recall rate, Character need to expand similarity words window to 4,000 tokens. In contrast, 1,000 tokens window retrieves 63.8% phrases with word level components.

We conclude that the failure with character level composing lies in the character embeddings being too far away from words and phrases that it forms.

5 Conclusion and Future Work

We investigate the token definition problem of embedding for semantic representation by phrase composition task in Chinese. Evaluated on different composing methods, composing precision and Wikipedia redirection prediction both show that each method with word embedding outperforms the same method with character embedding. This indicates word embedding might be better in semantic representation then character embedding.

Our future work includes 2 directions. (1) We plan to conduct more experiments on semantic analyze tasks and evaluate on semantic representativeness of word embedding and character embedding from more perspectives. (2) We plan to create more complex and precise phrase semantic composing models and try to compose phrase, entities and out of vocabulary tokens better.

[2] We have excluded token themselfs in neighbors in all composing experiments to avoid bias.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
2. Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H.: Joint learning of character and word embeddings. In: International Conference on Artificial Intelligence, pp. 1236–1242 (2015)
3. Cho, K., Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP, pp. 16–37 (2014)
4. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**(8), 861–874 (2006)
5. Harris, Z.S.: Distributional Structure. Springer, Netherlands (1981). https://doi.org/10.1007/978-94-009-8467-7_1
6. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431 (2017)
7. Li, X., Wang, T.: Lexicon of Common Words in Contemporary Chinese. The Commercial Press, Beijing (2008)
8. 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)
9. Mitchell, J., Lapata, M.: Vector-based models of semantic composition. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, pp. 236–244 (2008)
10. Parikh, A., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2249–2255 (2016)
11. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543 (2014)
12. Shen, Y., He, X., Gao, J., Deng, L.: Learning semantic representations using convolutional neural networks for web search. In: International Conference on World Wide Web, pp. 373–374 (2014)
13. Shi, X., Zhai, J., Yang, X., Xie, Z., Liu, C.: Radical embedding: delving deeper to chinese radicals. In: 2010 European Signal Processing Conference, pp. 572–575 (2015)
14. Sundermeyer, M., Alkhouli, T., Wuebker, J., Ney, H.: Translation modeling with bidirectional recurrent neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 14–25 (2014)
15. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
17. Yang, B., Wong, D.F., Xiao, T., Chao, L.S., Zhu, J.: Towards bidirectional hierarchical representations for attention-based neural machine translation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1432–1441 (2017)

18. Yu, M., Dredze, M.: Learning composition models for phrase embeddings. Trans. Assoc. Comput. Linguist. **3**, 227–242 (2015)
19. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
20. Zheng, X., Chen, H., Xu, T.: Deep learning for chinese word segmentation and pos tagging. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 647–657 (2013)