Enhancing Chinese Word Embeddings from Relevant Derivative Meanings of Main-Components in Characters

Xinyu Su, Wei Yang*, and Junyi Wang

School of Computer Science and Technology,
University of Science and Technology of China, Hefei, 230027, China
{sa517303,sa517352}@mail.ustc.edu.cn

Abstract. Word embeddings have a significant impact on natural language processing. In morpheme writing systems, most Chinese word embeddings take a word as the basic unit, or directly use the internal structure of words. However, these models still neglect the rich relevant derivative meanings in the internal structure of Chinese characters. Based on our observations, the relevant derivative meanings of the main-components in Chinese characters are very helpful for improving Chinese word embeddings learning. In this paper, we focus on employing the relevant derivative meanings of the main-components in the Chinese characters to train and enhance the Chinese word embeddings. To this end, we propose two main-component enhanced word embedding models named MCWE-SA and MCWE-HA respectively, which incorporate the relevant derivative meanings of the main-components during the training process based on the attention mechanism. Our models can fine-grained enhance the precision of word embeddings without generating additional vectors. Experiments on word similarity and syntactic analogy tasks are conducted to validate the feasibility of our models. Furthermore, the results show that our models have a certain improvement in the similarity task over most baselines, and have nearly 3% improvement in Chinese analogical reasoning dataset compared with the state-of-the-art model.

Keywords: Relevant derivative meaning · Component level · Enhanced word embedding.

1 Introduction

Data representation is the basic work in natural language processing (NLP), and the quality of data representation directly affects the performance of the entire system. Word embedding, which is also called distributed word representation, has been an important foundation in the field of NLP. It encodes the semantic meaning of a word into a real-valued low-dimensional vector, which performs better in many tasks such as text classification [1,2], machine translation [3,4],

* Corresponding author: qubit@ustc.edu.cn

sentiment analysis [5,6] and question answering [7,8] over traditional one-hot representations. Among many word embedding models, Continuous Bag-of-Word (CBOW) [9], Skip-gram [9] and Global Vectors(GloVe) [12] are popular because of their simplicity and efficiency. The idea of those algorithms is mainly based on the distributed hypothesis which means words that are used and occur in the same contexts tend to purport similar meanings [13]. However, these models only focus on word level information, and do not pay attention to the fine-grained morphological information inside the words or the characters, such as components of Chinese characters or English morphemes.

Different from English words, Chinese characters are glyphs whose components may depict objects or represent abstract notions. Usually a character consists of two or more components which may have meanings related to the character, using a variety of different principles[1]. That means Chinese words themselves are often composed of Chinese characters and subcharacter components, including abundant semantic information.

Previous researchers have done some work by using the abundant information inside Chinese for word embeddings enhancement with internal morphological semantics. Li et al. [14] used the radicals to enhance the Chinese character embeddings. Chen et al. [15] proposed the CWE model to improve the quality of Chinese word embeddings by exploiting character level information. For a more fine-grained combination of Chinese character and radical, Yin et al. [16] proposed methods to enhance Chinese character embeddings based on CWE model. Jian et al. [17] used external language to calculate the similarity between Chinese words and characters to enhance quality of word vectors based on the rich internal structure information of Chinese words. Huang et al. [19] proposed the GWE model, a pixel-based model that learns character features from font images to enhance the representation of words. Yu et al. [18] used Chinese characters and subcharacter components to improve Chinese word embeddings and proposed the JWE model to jointly learn Chinese word and character embeddings. Cao et al. [20] proposed cw2vec model, which exploits stroke-level information to improve the learning of Chinese word embeddings.

However, the subcharacter components of a character contain a lot of extra noise information, so we explore a new direction for better quality of word embeddings through integrating several word components into main-component. Chinese characters are composed of components and radicals, and a component of the complex subword item may be a simple Chinese character. In our models, the subcharacter components of characters can be roughly divided into two types: main-component and radical. The main-component which consists of several components indicates the basic meaning of a character while the radical indicates some attribute meanings of a character. For example, as Fig. 1 shows, “慧” (intelligent) is divided into the subcharacter components “丰” (abundant), “彐” (snow) and “心” (heart) where “丰” (abundant) and “彐” (snow) are the components and “心” (heart) is the radical. All these subcharacter components mentioned above may be not relevant to the semantics of the

[1] https://en.wikipedia.org/wiki/Written_Chinese

character “慧” (intelligent). However, the main-component “彗” (clever) consisting of several small components is closely related to the meaning of the character.

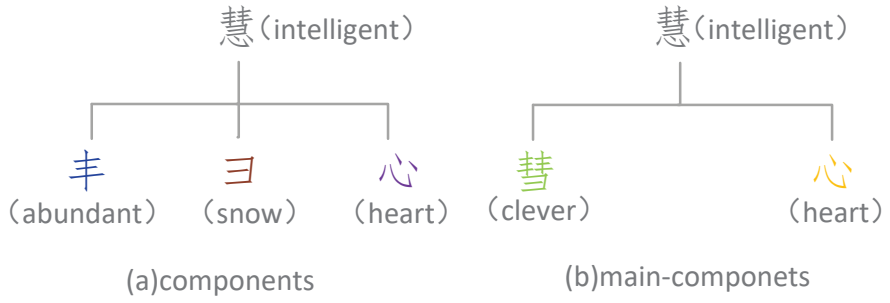

Fig. 1. An illustrative example to show the derivative meanings of the main-components are more relevant to the target word than the components.

The attention mechanism [21] originates from the way of imitating human thinking. It allocates limited information processing resources to the important part, that is, assigns different weights to different parts of the input. Soft attention concentrates on each input value and assigns a weight between 0 and 1. On the other hand, hard attention only focuses on the components that need the most attention, selectively deletes some low-associative values, and assigns weights to 0 or 1.

In this paper, we modify the CBOW model in the word2vec source code[2] and introduce the information of the relevant derivative meanings of the Chinese main-components, and propose two efficient models called MCWE-SA and MCWE-HA. The learned Chinese word embeddings not only contain more morphological information, but also have a higher similarity to the synonym. Our models directly modify embeddings of the target words, without generating and training extra embeddings for relevant derivative meanings. In addition, we create a derivative meaning table to describe the relationship between Chinese words and the relevant derivative meanings of their main-components. Through our models, the rich implicit information in Chinese is fully utilized and the similarity of related words is improved. Our contributions of this paper can be summarized as follows:

- Rather than directly leverage the components of the word itself, we provide a method to use the relevant derivative meanings of the main-components to train the word embeddings. In order to verify the feasibility of our method, two models named MCEW-SA and MCWE-HA, are proposed to incorporate the extra meanings.

[2] https://code.google.com/p/word2vec

- We put forward a method to assign the weights of relevant derivative meanings at input layer based on the attention scheme. Through the attention mechanism, the derivative meanings negatively related to the target word will be filtered in order to improve the accuracy of word embedding.

- We evaluate the quality of word embeddings learned by our models and the state-of-the-art models, using a medium-sized corpus through word similarity tasks and word analog tasks. The results show that all of our models have performance improvements and outperform all baselines on the word analog task.

2 Main-Component Word Embedding Models

In this section, we introduce Main-Component Word Embedding models, named MCWE-Soft Attention (MCWE-SA) and MCWE-Hard Attention (MCWE-HA), which is based on CBOW model [9]. It should be noted that our models use the internal meanings of the Chinese characters, rather than directly using the characters or components of the word themselves. In particular, some radicals are also main-components, so we treat these radicals as the main-components and have the same contribution in our models. Most of the main-components are frequently-used Chinese words which may contain some ambiguous information. To address this concern, we propose the MCWE-SA which bases on the soft attention scheme. The MCWE-SA assumes that the relevant derivative meanings of main-components have their own weights, and assigns higher weights to the meanings closely related to the target words, so that they have greater impact on the word embeddings. We treat soft attention as a filter to remove the negative correlation and add positive contributions to the target word vector. What's more, MCWE-HA which bases on the hard attention scheme only focuses on the relevant derivative meaning of the main-component with the highest similarity to the target word. When backpropagating the target word vector parameters, we introduce the vector update method for related derivative meanings. In what follows, we will introduce each of our models in detail.

2.1 MCWE-SA

Through observation, we discover that most Chinese words have more than one relevant derivative meaning with their main-components, but some relevant derivative meanings have low correlation with the corresponding word. For example, main-component "知*(know)*" means "学识*(knowledge)*" and "了解*(understand)*". As Fig. 2 shows, for the item "智慧*(intelligence)*" $\mapsto$ {学识*(knowledge)*, 了解*(understand)*,太阳*(sun)*, 时间*(time)*,白天*(day)*,聪慧*(clever)*, 扫把*(broom)*,思想*(thought)*, 心脏*(heart)*,感情*(feeling)*}, each relevant derivative meaning has a bias on the word "智慧*(intelligence)*". Therefore, we assign different weights to each relevant derivative meaning based on the idea of soft attention model. We measure the weights of relevant derivative meanings by

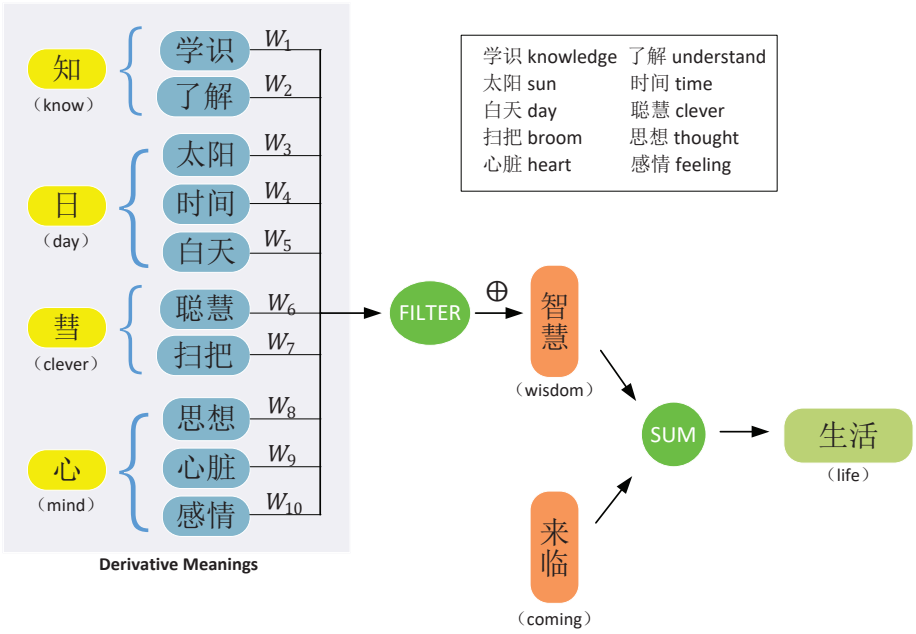

Fig. 2. An example of MCWE-SA. We take the sentence "智慧生活来临*(wisdom life coming)*" as an example. When we select "智慧*(wisdom)*" as the input word and calculate its word vector, we pick out the relevant derivative meanings of the main-components of "智慧" from the derivative meaning table, and add all the word vectors of the relevant derivative meanings which have a positive correlation to the vector of "智慧" according to the correlation weights.

calculating the cosine similarity between the corresponding relevant derivative meanings and the target word where cosine similarity is usually used to measure the similarity between word embeddings.

We denote $W = (w_1; w_2; ....; w_i)$ as the vocabulary of words, $M_i = (m_1; m_2; ...; m_k)$ as the relevant derivative meanings in the derivative meaning table for each word w_i. The item for w_i in the derivative meaning table is $w_i \mapsto M_i$ where M_i is a collection of the relevant derivative meanings of w_i's main-components. We denote sim$(\cdot)$ as a method to measure the similarity between Chinese words and their relevant derivative meanings. Furthermore, we remove the negatively correlated derivative meanings. Hence, at the input layer, the modified embedding of w_i can be expressed as

$$\hat{v}_{w_i} = \frac{1}{2}\{v_{w_i} + \frac{1}{N_i}\sum_{k=1}^{N_i} sim(w_i, m_k) \cdot m_k\}, \qquad (1)$$
$$sim(w_i, m_k) = cos(v_{w_i}, v_{m_k}), \quad s.t.\ \cos(v_{w_i}, v_{m_k}) > 0$$

where v_{w_i} is the original word embedding of w_i, $\hat{v}_{w_i}$ is the modified word embedding of w_i and v_{m_k} indicates the vector of relevant derivative meaning m_k that positively contributes to w_i. N_i denotes the number of v_{m_k} whose cosine similarity between v_{w_i} is greater than 0.

2.2 MCWE-HA

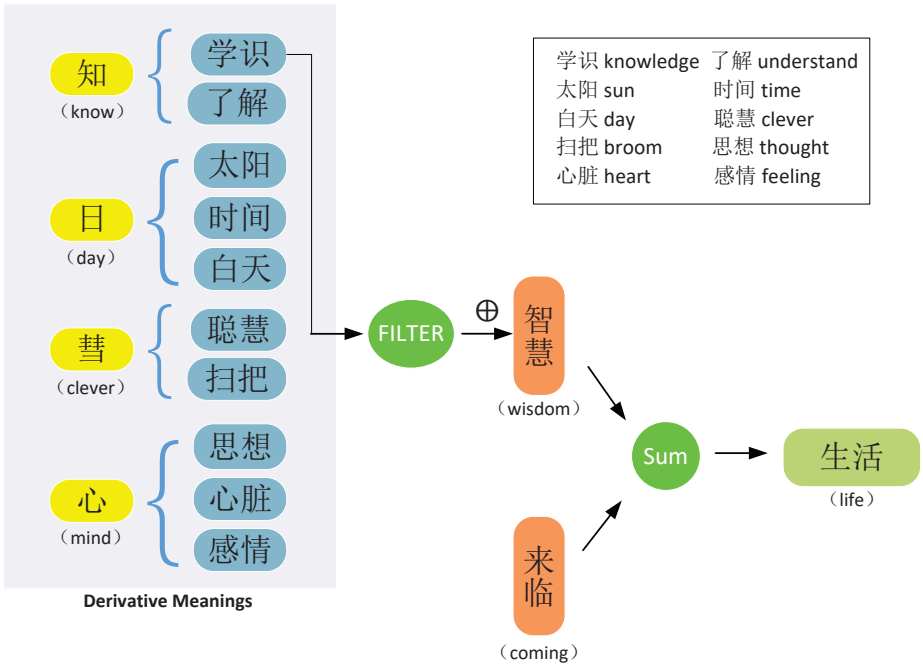

Fig. 3. An example of MCWE-HA. When we choose "智慧*(wisdom)*" as the input word and calculate its word vector, we select the relevant derivative meaning of the main-component with the highest similarity to "智慧*(wisdom)*" from the derivative meaning table, here we choose the vector of "学识*(knowledge)*". If the angle between the vector of "学识" and "智慧" is not greater than approximately 25°, we add the two vectors and get the average of their sum.

In order to further reduce the impact of low-relevant derivative meanings to the word, we propose MCWE-HA which is based on the idea of hard attention model. We only choose the derivative meaning which has the greatest cosine similarity to the target word. According to experimental experience, we retain the derivative meanings of similarity with token w_i greater than 0.9. We denote M_i as a collection of relevant derivative meanings of the word w_i, finally MCWE-

HA is mathematically defined as

$$\begin{aligned} \hat{v}_{w_i} &= \frac{1}{2}(v_{w_i} + v_{m_{max}}), \\ m_{max} &= \underset{m_k}{argmax}\, cos(v_{w_i}, v_{m_k}), m_k \in M_i \\ s.t.\ &\cos(v_{w_i}, v_{m_k}) > 0.9 \end{aligned} \tag{2}$$

where m_{max} denotes the vector of derivative meaning which has the greatest cosine similarity to w_i. We denote a paradigm of Fig. 3 to better illustrate the MCWE-HA model.

2.3 Update Rules for Relevant Derivative Meanings

Considering that the relevant derivative meanings are highly correlated with the target word embeddings and this relationship does not change in the current training round, when we update the embedding of the target word, we also choose to update the implied words which have great similarity with the corresponding word. According to the experimental experience, the cosine similarity between the derivative meanings and the target word is set to be greater than 0.9, which means the derivative meanings' word vector whose the angle between the two vectors not exceeding approximately 25° is updated. Our models aim to maximize the log-likelihoods function as follows:

$$\begin{aligned} L(v_{m_k}) &= \sum_{m_k \in M} \log P(v_{m_k} | context(v_{w_i})) \\ s.t.\ &\cos(v_{w_i}, v_{m_k}) > 0.9 \end{aligned} \tag{3}$$

where $context(v_{w_i})$ represents the context window of the word v_{w_i}, which is the composition of context words, and M represents the set of all relevant derivative meanings in derivative meaning table. $P(v_{m_k}|context(v_{w_i})$ is conditional probability which is defined by the softmax function. When updating v_{w_i} in backpropagation, we also update v_{m_k} with the same weight.

3 Experiments and Analysis

In this section, we evaluate the performance of our models in generating high-quality word embeddings on word similarity evaluation and word analogy task.

3.1 Experimental Settings

Training Corpus: We adopt a medium-sized Chinese corpus which is downloaded from Chinese Wikipedia Dump[3] to train all word embedding models. We

[3] https://download.wikipedia.com/zhwiki

utilize a script named WikiExtractor[4] to convert data from XML into text format. Moreover, we use THULAC[5] for Chinese word segmentation and normalize all characters as simplified Chinese. In pre-processing, we filter all digits, punctuation masks and non-Chinese characters in order to improve the efficiency of model training. To get better quality of the word embeddings, we remove the Chinese common stop words[6] in the corpus. Finally, we obtained a training corpus of approximately 2GB in size, containing 354,707,204 tokens and 1,090,983 unique words.

Derivative Meaning Table: The Modern Chinese Word List[7] is divided into two parts: 2500 common characters and 1000 secondary common characters, and the coverage of those words in most corpora reached 99.48% which means mastering the Chinese common and secondary words has reached the basic requirements for using Chinese[8]. Hence, we use crawler script to obtain the main-components and radicals' information of those Chinese characters which is a total of 3500 from HTTPCN[9]. In this step, we finally obtained 3491 main-components.

To create the derivative meaning table, we need to obtain the relevant derivative meanings of the main-components of each character. Although the main-components are part of the characters, they are also simple Chinese characters with their own meanings. Therefore, we crawl the Chinese interpretations of 3491 main-components from HTTPCN and obtain the relevant derivative meanings. By manually labeling, we simplify the interpretation of each main-component into a core phrase which may appear in the corpus. Although this process costs manpower and time, it could be done once and for all for each language because it has the same knowledge base. We traverse the entire corpus, pick out all the different words and put their main-components' derivative meanings in the table, so as to improve efficiency by looking up the table during model training. When we choose a Chinese word during training, its main-components will be determined and can be further replaced by derivative meanings as an intermediate variable by looking up the table.

Baselines: For comparison, we choose two classic models including CBOW [9] and GloVe [12] and two component-level state-of-the-art character embedding models including CWE [15] and GWE [19]. In addition, we introduce a model named Latent Meaning Model-Average(LMM-A) [22], which uses the latent meanings of English morphemes to enhance the word embeddings. LMM-A employs the morpheme embeddings to adjust the word embeddings of the target

[4] https://github.com/attardi/wikiextractor
[5] http://thulac.thunlp.org/
[6] https://github.com/goto456/stopwords
[7] https://en.wikipedia.org/wiki/Simplified_Chinese_characters
[8] https://en.wikipedia.org/wiki/List_of_Commonly_Used_Characters_in_Modern_Chinese
[9] http://tool.httpcn.com/zi

word during training, and assumes that all latent meanings have the same contribution to the target word. We modified the source code of the LMM-A model to match our experiments. In order to better verify the word analog performance of our models, we also selected two additional models named cw2vec [20] and JWE [18] respectively for word analogy experiment.

Parameter Settings: For the sake of fairness, we use the same hyperparameter settings for all models. In order to speed up the training process, we adopt negative sampling techniques for CBOW, CWE, GWE, LMM-A and our models. What's more, we set the word vector dimension as 200, the window size as 5, the negative samples as 10, the training iteration as 15, the learning rate as 0.025 and the subsampling parameter as 1e-4.

3.2 Word Similarity

This experiment is used to evaluate the semantic relevance of generated word embeddings in word pairs. For Chinese word similarity task, we employ two different manually-annotated datasets named wordsim-240 and wordsim-296 provided by Chen et al. [15]. These datasets are composed of word pairs and manually labeled with the scores to measure the similarity of word pairs. We utilize the cosine similarity to measure the similarity of each word pair, and the Spearman' s rank correlation coefficient (ρ) is employed to evaluate our calculation results and human scores. More details of results are shown in Table. 1.

The performance of our models on the Wordsim-240 dataset exceeds the original CBOW model, which indicates that adding implicit information to the input layer during the model training can indeed improve the quality of the word vector. The performance of the MCWE-SA model exceeds all baselines on the Wordsim-296, indicating that adding the derivative meanings of positive elements can effectively enhance the similarity between words. On the other hand, the LMM-A model does not perform well on both datasets, indicating that the method of averaging all the latent meanings' weights is not desirable because it contains a large amount of negative correlation information. Our strategy uses the attention mechanism to select information with positive meanings and filter out the negative correlation vector, which finally improves the quality of Chinese word embeddings. Although our models have little improvement in the word similarity task compared to the CBOW model, we verify the method that adding derivative meanings is a feasible direction, and we can continue to improve performance by optimizing the derivative meaning table.

3.3 Syntactic Analogy

This task examines the quality of word embeddings by discovering the semantic inferential capability between pairs of words. The core task of syntactic analogy is to answer the questions like "雅典 (Athens) is to 希腊 (Greece) as 东京 (Tokyo) is to 日本 (Japan)" where 日本(Japan) is the answer we hope to get. This means

Table 1. Spearman's Coefficients of word similarity on wordsim-240 and wordsim-296($\rho\times100$). The higher the values, the better the performance.

Model	Wordsim-240	Wordsim-296
CBOW	51.17	57.46
GloVe	50.15	42.07
CWE	**53.40**	57.06
GWE	52.07	56.98
LMM-A	39.45	43.01
MCWE-SA	52.23	**57.95**
MCWE-HA	51.21	57.23

that the model gets the answer correctly if the similarity of "vector (希腊) - vector (雅典) + vector (东京)" and "vector (日本)" is the largest among all words. We utilize the Chinese analogical reasoning dataset created by Chen et al. [15], which contains 3 analogy types: some capitals and their countries (677 groups), some cities and their states (175 groups) and family relationships (272 groups). Each strategy contains four phrases, and we use the information from the first three phrases to predict the fourth and calculate the accuracy of the final result. The results in Table. 2 show that our models outperform the comparative baselines in all classifications. The JWE model uses subcharacter components for word embedding enhancements, which has a good effect compared to most baselines but ignores the implicit information inside the characters, so the final performance is still not as good as our models. The CBOW model is stable and exhibits high performance, but still weaker than our models. Since our models make the spatial distance of words with the same main-components closer and the words with the same main-components have similar derivative meanings, our word embeddings achieve high performance in semantic analogy.

Table 2. Evaluation accuracies (%) on word analogy reasoning. The higher the values, the better the performance.

Model	Total	Capital	State	Family
CBOW	79.92	86.91	85.14	60.66
GloVe	50.84	55.08	68.00	30.14
CWE	59.00	66.72	65.71	37.13
GWE	64.54	71.24	77.14	41.17
LMM-A	33.87	31.01	28.00	44.12
JWE	73.07	78.83	82.28	54.04
cw2vec-subword	39.11	37.64	71.42	21.69
MCWE-SA	**81.43**	**89.18**	88.00	59.56
MCWE-HA	81.33	87.40	**89.72**	**62.13**

3.4 The Impacts of Corpus Size

The parameter settings in word embedding training have a great impact on final result. The larger corpus size, the more semantic information it contains, which can improve the quality of the word vector. We take a task to investigate the impact of corpus size for word embeddings. In the analysis of corpus size, we set the same hyperparameters as before. We select the Chinese analogical reasoning dataset as the evaluation standard of syntactic analogy task. The entire corpus previously mentioned is divided into $\frac{1}{4}$, $\frac{2}{4}$, $\frac{3}{4}$, and $\frac{4}{4}$ respectively as our new corpus for the task. As shown in Fig. 4, the performance of the MCWE-SA model is better than the CBOW model in each corpus. Although the MCWE-HA model has weaker performance than CBOW at the beginning, with the increment of the corpus, the performance exceeds that of CBOW from the point that the corpus's size is about 800MB.

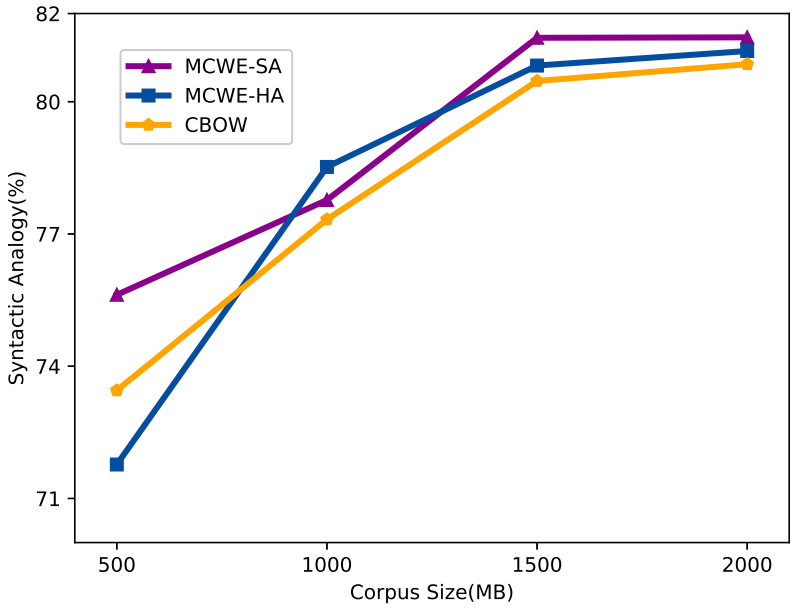

Fig. 4. Results on Chinese analogical reasoning dataset with different corpus size.

4 Conclusion

In this paper, we explored a new direction of using the relevant derivative meanings of Chinese internal components instead of themselves to enhance the Chinese

word embeddings. We proposed two models named MCWE-SA and MCWE-HA, which make full use of subword information. The attention model was used to dynamically adjust the weights of derivative meanings of the main-components in Chinese characters. Experimental results showed that our models have obvious advantages in syntactic analogy compared to all baselines.

Acknowledgments

The authors are grateful to the reviewers for constructive feedback. We would like to thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Natural Science Foundation of China (No. 61572456) and the Anhui Initiative in Quantum Information Technologies (No. AHY150300).

References

1. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In: ACL, pp 1555–1565 (2014)
2. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for natural language processing. arXiv preprint arXiv:1606.01781. (2016)
3. Jean, S., Cho, K., Memisevic, R., Bengio, Y.: On using very large target vocabulary for neural machine translation. arXiv preprint arXiv:1412.2007. (2015)
4. Kyunghyun, C., Bart, M., Caglar, G., Dzmitry, B., Fethi, B., Holger, S., Yoshua, B.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv: 1406.1078. (2014)
5. Bonggun, S., Timothy, L., Jinho, D.: Lexicon integrated cnn models with attention for sentiment analysis. arXiv preprint arXiv:1610.06272. (2016)
6. Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., Zhou, M.:Sentiment Embeddings with Applications to Sentiment Analysis. In: IEEE Transactions on Knowledge and Data Engineering, pp 496-509 (2016)
7. Guangyou, Z., Tingting, H., Jun, Z, Po, H.: Learning continuous word embedding with metadata for question retrieval in community question answering. In: ACL, pp 250–259 (2015)
8. Antoine, B., Sumit, C., Jason, W.: Question Answering with Subgraph Embeddings. arXiv preprint arXiv:1406.3676. (2014)
9. Tomas, M., Kai, C., Greg, C., Jeffrey, D.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013a)
10. Tomas, M., Ilya, S., Kai, C., Greg, S., Jeff, D.: Distributed representations of words and phrases and their compositionality.In: Advances in neural information processing systems, pp. 3111–3119 (2013b)
11. Tomas, M., Wen-tau, Y., Geoffrey, Z.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 746–751 (2013c)
12. Jeffrey, P., Richard, S., Christopher, M.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)

13. Harris, Z.: "Distributional structure".In: Word.10(23), pp. 146–162 (1954) https://doi.org/10.1080/00437956.1954.11659520.
14. Li, Y., Li, W., Sun, F., Li, S.: Component-enhanced chinese character embeddings.In: Proceedings of EMNLP, pp. 829–834 (2015)
15. Chen, X.; Xu, L.; Liu, Z.; Sun, M.; Luan, H.: Joint learning of character and word embeddings. In: IJCAI, pp. 1236–1242 (2015).
16. Rongchao, Y., Quan, W., Peng, L., Rui, L., Bin, W.: Multi-granularity Chinese word embedding.In: Proceedings of EMNLP, pp. 981–986 (2016)
17. Xu, J., Liu, J., Zhang, L., Li, Z., Chen, H.: Improve Chinese Word Embeddings by Exploiting Internal Structure.In: NAACL, pp. 1041-1050 (2016)
18. Yu, J., Jian. X., Xin, H., Song, Y.: Joint embeddings of Chinese words, characters, and fine-grained subcharacter components. In: EMNLP, pp. 286-291 (2017)
19. Tzu-Ray, S., Hung-Yi, L.: Learning Chinese word representations from glyphs of characters. In: EMNLP (2017)
20. Cao, S., Lu, W., Zhou, J., Li, X.:cw2vec: Learning Chinese Word Embeddings with Stroke n-gram Information. In: AAAI (2018)
21. Kelvin, X., Jimmy B., Ryan K., Aaron C., Ruslan S., Richard Z., Yoshua B.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: ICML (2015)
22. Xu, Y., Liu, J., Yang, W.,Huang, L.: Incorporating Latent Meanings of Morphological Compositions to EnhanceWord Embeddings. In: ACL, pp. 1232-1242 (2018)
23. Lai, S., Liu, K., Xu, L., Zhao, J.:How to Generate a Good Word Embedding?. arXiv preprint arXiv:1507.05523. (2015)
24. Piotr, B., Edouard, G., Armand, J., Tomas, M.: Enriching Word Vectors with Subword Information. In: ACL, pp. 135-146 (2017)
25. Chen, Z., Hu, K.: Radical Enhanced Chinese Word Embedding. In: CCL(The Seventeenth China National Conference on Computational Linguistics) (2018)