Joint Embeddings of Chinese Words, Characters, and Fine-grained Subcharacter Components

Jinxing Yu Xun Jian Hao Xin Yangqiu Song
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{jyuat, xjian, hxinaa, yqsong}@cse.ust.hk

Abstract

Word embeddings have attracted much attention recently. Different from alphabetic writing systems, Chinese characters are often composed of subcharacter components which are also semantically informative. In this work, we propose an approach to jointly embed Chinese words as well as their characters and fine-grained subcharacter components. We use three likelihoods to evaluate whether the context words, characters, and components can predict the current target word, and collected 13,253 subcharacter components to demonstrate the existing approaches of decomposing Chinese characters are not enough. Evaluation on both word similarity and word analogy tasks demonstrates the superior performance of our model.

1 Introduction

Distributed word representation represents a word as a vector in a continuous vector space and can better uncover both the semantic and syntactic information over traditional one-hot representations. It has been successfully applied to many downstream natural language processing (NLP) tasks as input features, such as named entity recognition (Collobert et al., 2011), text classification (Joulin et al., 2016), sentiment analysis (Tang et al., 2014), and question answering (Zhou et al., 2015). Among many embedding methods (Bengio et al., 2003; Mnih and Hinton, 2009), CBOW and Skip-Gram models are very popular due to their simplicity and efficiency, making it feasible to learn good embeddings of words from large scale training corpora (Mikolov et al., 2013b,a).

Despite the success and popularity of word embeddings, most of the existing methods treat each word as the minimum unit, which ignores the morphological information of words. Rare words cannot be well represented when optimizing a cost function related to a rare word and its contexts. To address this issue, some recent studies (Luong et al., 2013; Qiu et al., 2014; Sun et al., 2016a; Wieting et al., 2016) have investigated how to exploit morphemes or character n-grams to learn better embeddings of English words.

Different from other alphabetic writing systems such as English, written Chinese is logosyllabic, i.e., a Chinese character can be a word on its own or part of a polysyllabic word[1]. The characters themselves are often composed of subcharacter components which are also semantically informative. The subword items of Chinese words, including characters and subcharacter components, contain rich semantic information. The characters composing a word can indicate the semantic meaning of the word and the subcharacter components, such as radicals and components themselves being a character, composing a character can indicate the semantic meaning of the character. The components of characters can be roughly divided into two types: semantic component and phonetic component. The semantic component indicates the meaning of a character while the phonetic component indicates the sound of a character. For example, 氵 (water) is the semantic component of characters 湖 (lake) and 海 (sea), 马 (horse) is the phonetic component of characters 妈 (mother) and 骂 (scold) where both 妈 and 骂 are pronounced similar to 马.

Leveraging the subword information such as characters and subcharacter components can enhance Chinese word embeddings with internal morphological semantics. Some methods have been proposed to incorporate the subword infor-

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

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 286–291
Copenhagen, Denmark, September 7–11, 2017.

mation for Chinese word embeddings. Sun et al. (2014) and Li et al. (2015) proposed methods to enhance Chinese character embeddings with radicals based on C&W model (Collobert and Weston, 2008) and word2vec models (Mikolov et al., 2013a,b) respectively. Chen et al. (2015) used Chinese characters to improve Chinese word embeddings and proposed the CWE model to jointly learn *Chinese word and character* embeddings. Xu et al. (2016) extended the CWE model by exploiting the internal semantic similarity between a word and its characters in a cross-lingual manner. To combine both the *radical-character* and *character-word* compositions, Yin et al. (2016) proposed a multi-granularity embedding (MGE) model based on the CWE model, which represents the context as a combination of surrounding words, surrounding characters, and the radicals of the target word. Particularly, they developed a dictionary of 20,847 characters and 296 radicals.

However, all the above approaches still missed a lot of fine-grained components in Chinese characters. Formally and historically, radicals are character components used to index Chinese characters in dictionaries. Although many of the radicals are also semantic components, a character has only one radical, which cannot fully uncover the semantics and structure of the character. Besides over 200 radicals, there are more than 10,000 components which are also semantically meaningful or phonetically useful. For example, Chinese character 照 (illuminate, reflect, mirror, picture) has one radical 灬 (the corresponding traditional Chinese radical is 火, meaning fire) and three other components, i.e., 日 (sun), 刀 (knife), and 口 (mouth). Shi et al. (2015) proposed using WUBI input method to decompose the Chinese characters into components. However, WUBI input method uses rules to group Chinese characters into meaningless clusters which can fit the alphabet based keyboard. The semantics of the components are not straightforwardly meaningful.

In this work, we present a model to jointly learn the embeddings of Chinese words, characters, and subcharacter components. The learned Chinese word embeddings can leverage the external context co-occurrence information and incorporate rich internal subword semantic information. Experiments on both word similarity and word analogy tasks demonstrate the effectiveness of our model over previous works. The code and data are available at https://github.com/HKUST-KnowComp/JWE.

2 Joint Learning Word Embedding

In this section, we introduce our joint learning word embedding model (JWE), which combines words, characters, and subcharacter components information. Our model is based on CBOW model (Mikolov et al., 2013a). JWE uses the average of context word vectors, the average of context character vectors, and the average of context subcharacter vectors to predict the target word, and uses the sum of these three prediction losses as the objective function.

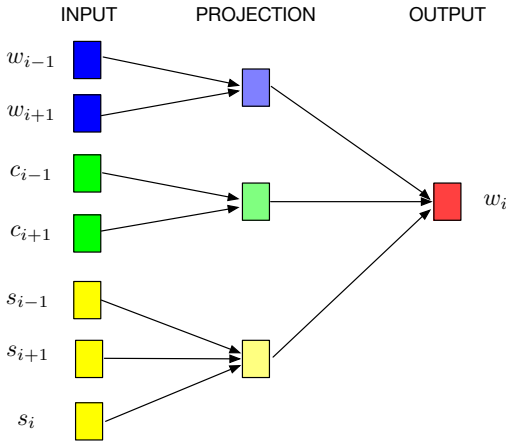

Figure 1: Illustration of JWE. w_i is the target word. w_{i-1} and w_{i+1} are the left word and right word of w_i respectively. c_{i-1} and c_{i+1} represent the characters in the context. s_{i-1} and s_{i+1} represent the subcharacters in the context, s_i represents the subcharacters of the target word w_i.

We denote D as the training corpus, $W = (w_1, w_2, \cdots, w_N)$ as the vocabulary of words, $C = (c_1, c_2, \cdots, c_M)$ as the vocabulary of characters, $S = (s_1, s_2, \cdots, s_K)$ as the vocabulary of subcharacters, and T as the context window size respectively. As illustrated in Figure 1, JWE aims to maximize the sum of log-likelihoods of three predictive conditional probabilities for a target word w_i:

$$L(w_i) = \sum_{k=1}^{3} \log P(w_i | h_{i_k}), \qquad (1)$$

where $h_{i_1}, h_{i_2}, h_{i_3}$ are the composition of context words, context characters, context subcharacters respectively. Let $\boldsymbol{v}_{w_i}, \boldsymbol{v}_{c_i}, \boldsymbol{v}_{s_i}$ be the "input" vectors of word w_i, character c_i, and subcharacter s_i

respectively, $\hat{\boldsymbol{v}}_{w_i}$ be the "output" vectors of word w_i. The conditional probability is defined by the softmax function as follows:

$$p(w_i|h_{i_k}) = \frac{\exp(\boldsymbol{h}_{i_k}^T \hat{\boldsymbol{v}}_{w_i})}{\sum_{j=1}^{N} \exp(\boldsymbol{h}_{i_k}^T \hat{\boldsymbol{v}}_{w_j})}, \quad \text{k = 1, 2, 3,} \tag{2}$$

where $\boldsymbol{h}_{i_1}$ is the average of the "input" vectors of words in the context, i.e.:

$$\boldsymbol{h}_{i_1} = \frac{1}{2T} \sum_{-T \le j \le T, j \ne 0} \boldsymbol{v}_{w_{i+j}}. \tag{3}$$

Similarly, $\boldsymbol{h}_{i_2}$ is the average of characters' "input" vectors in the context, $\boldsymbol{h}_{i_3}$ is the average of subcharacters' "input" vectors in the context or in the target word or all of them. Given a corpus D, JWE maximizes the overall log likelihood:

$$L(D) = \sum_{w_i \in D} L(w_i), \tag{4}$$

where the optimization follows the implementation of negative sampling used in CBOW model (Mikolov et al., 2013a).

This objective function is different from that of MGE (Yin et al., 2016). For a target word w_i, the objective function of MGE is almost equivalent to maximizing $P(w_i|\boldsymbol{h}_{i_1} + \boldsymbol{h}_{i_2} + \boldsymbol{h}_{i_3})$. During the backpropagation, the gradients of $\boldsymbol{h}_{i_1}, \boldsymbol{h}_{i_2}, \boldsymbol{h}_{i_3}$ can be different in our model while they are always same in MGE, so the gradients of the embeddings of words, characters, subcharacter components can be different in our model while they are same in MGE. Thus, the representations of words, characters, and subcharacter components are decoupled and can be better trained in our model. A similar decoupled objective function is used in (Sun et al., 2016a) to learn English word embeddings and phrase embeddings. Our model differs from theirs in that we combine the subwords of both the context words and target word to predict the target word while they use the morphemes of the target English word to predict it.

3 Experiments

We quantitatively evaluate the quality of word embeddings learned by our model on word similarity evaluation and word analogy tasks.

3.1 Experimental Settings

Training Corpus. We adopt the Chinese Wikipedia Dump[2] as our training corpus. In preprocessing, pure digits and non Chinese characters are removed. We use THULAC[3] (Sun et al., 2016b) for Chinese word segmentation and POS tagging. We identify all entity names for CWE (Chen et al., 2015) and MGE (Yin et al., 2016) as they do not use the characters information for non-compositional words. Our model (JWE) does not use such a non-compositional word list. We obtained a 1GB training corpus with 153,071,899 tokens and 3,158,225 unique words.

Model	Wordsim-240	Wordsim-295
CBOW	0.5009	0.5985
CWE	0.5133	0.5805
MGE	0.5128	0.5425
JWE+c+p1	0.5437	0.6549
JWE+c+p2	0.5476	0.6676
JWE+c+p3	0.5554	0.6533
JWE+r+p1	0.5478	0.6434
JWE+r+p2	**0.5619**	0.6621
JWE+r+p3	0.5273	0.6461
JWE-n	0.5476	**0.6710**

Table 1: Results on word similarity evaluation. For our JWE model, +c represents the components feature and +r represents the radicals feature; +p indicates which subcharacters are used to predict the target word; +p1 indicates using the surrounding words' subcharacter features; +p2 indicates using the target word's subcharacter features; +p3 indicates using the subcharacter features of both the surrounding words and the target word; -n indicates only using characters without either components or radicals.

Subcharacter Components. We crawled the components and radicals information of Chinese characters from HTTPCN[4]. We obtained 20,879 characters, 13,253 components and 218 radicals, of which 7,744 characters have more than one components, and 214 characters are equal to their radicals.

Parameter Settings. We compare our method with CBOW (Mikolov et al., 2013b)[5] , CWE (Chen et al., 2015)[6], and MGE (Yin et al., 2016)[7].

[2] http://download.wikipedia.com/zhwiki

[3] http://thulac.thunlp.org/

[4] http://tool.httpcn.com/zi/

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

[6] https://github.com/Leonard-Xu/CWE

[7] We used the source code provided by the author. Our experimental results of baselines are different from that in MGE paper because we used a 1GB corpus while they used a 500MB corpus and we fixed the training iteration while they tried the training iteration in range [5, 200] and chose the best.

For all models, we used the same parameter settings. We fixed the word vector dimension to be 200, the window size to be 5, the training iteration to be 100, the initial learning rate to be 0.025, and the subsampling parameter to be 10^{-4}. Words with frequency less than 5 were ignored during training. We used 10-word negative sampling for optimization.

3.2 Word Similarity

This task evaluates the embedding's ability of uncovering the semantic relatedness of word pairs. We select two different Chinese word similarity datasets, wordsim-240 and wordsim-296 provided by (Chen et al., 2015) for evaluation. There are 240 pairs of Chinese words in wordsim-240 and 296 pairs of Chinese words in wordsim-296. Both datasets contain human-labeled similarity scores for each word pair. There is a word in wordsim-296 that did not appear in the training corpus, so we removed this from the gold-standard to produce wordsim-295. All words in wordsim-240 appeared in the training corpus. The similarity score for a word pair is computed as the cosine similarity of their embeddings generated by the learning model. We compute the Spearman correlation (Myers et al., 2010) between the human-labeled scores and similarity scores computed by embeddings. The evaluation results of our model and baseline methods on wordsim-240 and wordsim-295 are shown in Table 1.

From the results, we can see that JWE substantially outperforms CBOW, CWE, and MGE on the two word similarity datasets. JWE can better leverage the rich morphological information in Chinese words than CWE and MGE. It shows the benefits of decoupling the representation of words, characters, and subcharacter components as opposed to employing concatenation, sum, or average on all of them as the context.

We also observe that JWE with only characters can get competitive results on the word similarity task compared to JWE with characters and subcharacters. The reason may be that characters are enough to provide additional semantic information for computing the similarities of many word pairs in the two datasets. For example, the similarity of 法律 (law, statute) and 律师 (lawyer) in wordsim-295 can be directly inferred from the shared character 律 (law, rule).

3.3 Word Analogy

This task examines the quality of word embedding by its capacity of discovering linguistic regularities between pairs of words. For example, for a tuple like "罗马 (Rome): 意大利 (Italy):: 柏林 (Berlin): 德国 (Germany)", the model can answer correctly if the nearest vector representation to vec(意大利) - vec(罗马) + vec(柏林) is vec(德国) among all words except from 罗马, 意大利, and 柏林. More generally, given an analogy tuple "$a : b :: c : d$," the model answers the analogy question "$a : b :: c :?$" by finding x in the vocabulary such that

$$\arg \max_{x \neq a, x \neq b, x \neq c} cos(\vec{\boldsymbol{b}} - \vec{\boldsymbol{a}} + \vec{\boldsymbol{c}}, \vec{\boldsymbol{x}}).$$

We use accuracy as the evaluation metric. In this task, we use the Chinese word analogy dataset introduced by (Chen et al., 2015), which consists of 1,124 tuples of words and each tuple contains 4 words, coming from three different categories: "Capital" (677 tuples), "State" (175 tuples), and "Family" (272 tuples). Our training corpus covers all the testing words.

Model	Total	Capital	State	Family
CBOW	0.7954	0.8493	0.8857	0.6029
CWE	0.7553	0.8420	0.8743	0.4632
MGE	0.7696	0.8907	0.8857	0.3934
JWE+c+p1	0.7562	0.8272	0.8286	0.5331
JWE+c+p2	0.8407	0.8848	**0.9486**	**0.6618**
JWE+c+p3	**0.8505**	**0.9188**	0.9371	0.6250
JWE+r+p1	0.7553	0.8198	0.8171	0.5551
JWE+r+p2	0.8185	0.8656	0.9143	0.6397
JWE+r+p3	0.8416	0.9010	0.9200	0.6434
JWE-n	0.8229	0.8803	0.9028	0.6286

Table 2: Results on word analogy reasoning. The configurations are the same of the ones used in Table 1.

The results in Table 2 show that JWE outperforms the baselines on all categories' word analogy tasks. Different from the results on the word similarity task, JWE with components consistently performs better than JWE with radicals and JWE without either radicals or components. It demonstrates the necessary of delving deeper into fine-grained components for complex semantic reasoning tasks.

3.4 Case Studies

In addition to evaluating the benefits of incorporating subword information for Chinese word em-

beddings, it would be interesting to see the relationships of the embeddings of words, characters, and subcharacter components as they are embedded into a same continuous vector space.

照 (photograph)	照片 (photo) 相片 (photo) 拍照 (photograph) 护照 (passport) 照相 (photography)
河 (river)	黄河 (the Yellow River) 河流 (river) 河道 (watercourse) 运河 (canal) 河南 (Henan province)

Table 3: Closest words of characters 照 (photograph) and 河 (river).

Component	疒 (illness)
Closest characters	疗 (cure) 症 (symptom) 痛 (pain) 疮 (sore) 患 (suffer) 痒 (itch) 疳 (infantile malnutrition) 病 (disease) 肿 (swelling)
Closest words	治疗 (cure) 病症 (symptom) 复发 (recurrence) 疼痛 (pain) 症状 (symptom) 腹绞痛 (abdominal pain) 患者 (patients) 癫痫 (epilepsy) 疾病 (disease) 疗法 (therapy)

Table 4: Closest characters and closest words of the component 疒 (illness).

We evaluate the embeddings' abilities of uncovering the semantic relatedness of words, characters, and subcharacter components through case studies. The similarities between them are computed by the cosine similarities of their embeddings. Take two Chinese character 照 (photograph) and 河 (river) as examples, we list their closest words in Table 3. We can see that most of the closest words are semantically related to the corresponding character.

We further take the component 疒 (illness) as an example and list its closest characters and words in Table 4. All of the closest characters and words are semantically related to the component 疒 (illness). Most of them have the component 疒 (illness). 患 (suffer), 肿 (swelling), and 患者 (patients) do not have the component 疒 (illness), but they are also semantically related to 疒 (illness). It shows that JWE does not overuse the component information but leverages both the external context co-occurrence information and internal subword morphological information well.

4 Conclusion and Future Work

In this paper, we propose a model to jointly learn the embeddings of Chinese words, characters, and subcharacter components. Our approach makes full use of subword information to enhance Chinese word embeddings. Experiments show that our model substantially outperforms the baseline methods on Chinese word similarity computation and Chinese word analogy reasoning, and demonstrate the benefits of incorporating fine-grained components compared to just using characters.

There could be several directions to be explored for future work. First, we use the average operation to integrate the subcharacter components as the context to predict the target word. The structure of Chinese characters and the positions of components in the character may be considered to fully leverage the component information of Chinese characters. Second, for any target word, we simply use word context, character context, and subcharacter context to predict it and do not distinguish compositional words and non-compositional words. To solve this problem, attention models may be used to adaptively assign weights to word context, character context, and subcharacter context.

Acknowledgments

This paper was supported by HKUST initiation grant IGN16EG01, the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 26206717), China 973 Fundamental R&D Program (No. 2014CB340304), and the LORELEI Contract HR0011-15-2-0025 with the US Defense Advanced Research Projects Agency (DARPA). Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We also thank the anonymous reviewers for their valuable comments and suggestions that help improve the quality of this manuscript.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In *Proceedings of IJCAI*, pages 1236–1242.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. 2015. Component-enhanced chinese character embeddings. In *Proceedings of EMNLP*, pages 829–834.

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of CoNLL*, pages 104–113.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Proceedings of NIPS*, pages 1081–1088.

Jerome L Myers, Arnold Well, and Robert Frederick Lorch. 2010. *Research design and statistical analysis*. Routledge.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *Proceedings of COLING*, pages 141–150.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical embedding: Delving deeper to chinese radicals. In *Proceedings of ACL*, pages 594–598.

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016a. Inside out: Two jointly predictive models for word representations and phrase representations. In *Proceedings of AAAI*, pages 2821–2827.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016b. Thulac: An efficient lexical analyzer for chinese. *Technical Report*.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. In *International Conference on Neural Information Processing*, pages 279–286. Springer.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of ACL*, pages 1555–1565.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of EMNLP*, pages 1504–1515.

Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. 2016. Improve chinese word embeddings by exploiting internal structure. In *Proceedings of NAACL-HLT*, pages 1041–1050.

Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. 2016. Multi-granularity chinese word embedding. In *Proceedings of EMNLP*, pages 981–986.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of ACL*, pages 250–259.