Radical-Enhanced Chinese Character Embedding

Yaming Sun[†], **Lei Lin**[†] , **Duyu Tang**[†], **Nan Yang**[‡], **Zhenzhou Ji**[†], **Xiaolong Wang**[†]
[†]Harbin Institute of Technology, China
[‡]University of Science and Technology of China, Hefei, China

Abstract

We present a method to leverage radical for learning Chinese character embedding. **Radical** is a semantic and phonetic component of Chinese character. It plays an important role as characters with the same radical usually have similar semantic meaning and grammatical usage. However, existing Chinese processing algorithms typically regard word or character as the basic unit but ignore the crucial radical information. In this paper, we fill this gap by leveraging radical for learning continuous representation of Chinese character. We develop a dedicated neural architecture to effectively learn character embedding and apply it on Chinese character similarity judgement and Chinese word segmentation. Experiment results show that our radical-enhanced method outperforms existing embedding learning algorithms on both tasks.

1 Introduction

Chinese "**radical** (部首)" is a graphical component of Chinese character, which serves as an indexing component in the Chinese dictionary[1]. In general, a Chinese character is phono-semantic, with a radical as its semantic and phonetic component suggesting part of its meaning. For example, "氵(water)" is the radical of "河 (river)", and "足 (foot)" is the radical of "跑 (run)".

Radical is important for the computational processing of Chinese language. The reason lies in that characters with the same radical typically have similar semantic meanings and play similar grammatical roles. For example, verbs "打(hit)" and "拍(pat)" share the same radical "扌(hand)" and usually act as the subject-verb in sentences.

[1]http://en.wikipedia.org/wiki/Radical_(Chinese_character)

To our best knowledge, existing studies in Chinese NLP tasks, such as word segmentation, typically treat word (Zhang and Clark, 2010) or character (Zhang et al., 2013) as the basic unit, while ignore the radical information. In this paper, we leverage the radical information of character for the computational processing of Chinese. Specifically, we exploit the radical of character for learning Chinese character embedding. Most existing embedding learning algorithms (Bengio et al., 2003; Morin and Bengio, 2005; Mikolov et al., 2010; Huang et al., 2012; Luong et al., 2013; Mikolov et al., 2013b) model the representation for a word with its context information. We extend an existing embedding learning algorithm (Collobert and Weston, 2008; Collobert et al., 2011) and propose a tailored neural architecture to leverage radical for learning the continuous representation of Chinese character. Our neural model integrates the radical information by predicting the radical of each character through a $softmax$ layer. Our loss function is the linear combination of the loss of C&W model (Collobert et al., 2011) and the cross-entropy error of $softmax$. We apply the radical-enhanced character embedding on two tasks, Chinese character similarity judgement and Chinese word segmentation. Experiment results on both tasks show that, our method outperforms existing embedding learning algorithms which do not utilize the radical information. The major contributions of this paper are summarized as follows.

- To our best knowledge, this is the first work that leverages radical for learning Chinese character embedding.
- We learn Chinese character embedding by exploiting the radical information of character and verify its effectiveness on two tasks.

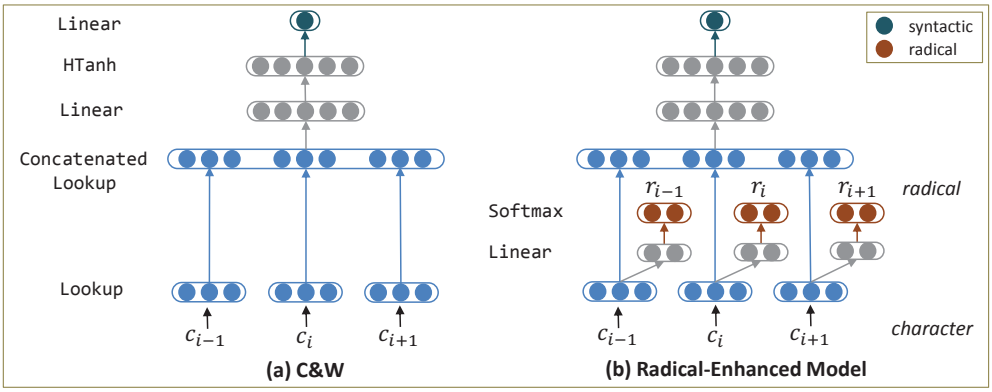

Figure 1: Architecture of C&W (a) and our radical-enhanced character embedding learning model (b).

- We publish the radical-enhanced Chinese character embedding, which can be easily applied on other Chinese NLP tasks. We also introduce a dataset on Chinese character similarity judgement.

This paper is organized as follows. We propose the radical-enhanced character representation learning model in Section 3. In section 4, we introduce the Chinese word segmentation task and the neural Conditional Random Field (CRF) model for utilizing character embedding as features. Then we present the experimental studies in Section 5, and finally conclude the paper in Section 6.

2 Related Work

In this section, we review the previous studies from two directions, namely learning word embedding and applying word embedding for NLP applications.

2.1 Learning Word Embedding

It is well-accepted that the representation of word is the basis of the field of natural language processing (Turney et al., 2010; Turian et al., 2010). In the early studies, a word is represented as a one-hot vector, whose length is the size of vocabulary, and only one dimension is 1, others are 0. The main drawback of the one-hot representation is that it can not reflect the grammatical and semantic relations between words. To overcome this shortcoming, some studies have been done to learn the latent factors of words, such as Latent Semantic Analysis (Deerwester et al., 1990) and Latent Dirichlet Allocation (Blei et al., 2003). With the revival of deep learning (Bengio, 2013), many researchers focus on the continuous representation of words (a.k.a word embedding). Existing embedding learning algorithms can be divided into two directions based on the use of unstructured raw texts (Collobert et al., 2011) or structured knowledge base (Bordes et al., 2011). Due to the lack of large-scale Chinese knowledge base (KB), this paper focuses on learning character embedding from unstructured corpus and leaves the KB-based method to the future work. From the perspective of learning embedding from raw corpus, most existing algorithms model the representation for a word with its context information. Bengio et al. (2003) propose a feed-forward neural probabilistic language model to predict the next word based on its previous contextual words. Based on their work, some methods are presented to reduce the training time of neural language model. Morin and Bengio (2005) and Mnih and Hinton (2008) propose hierarchical language models, which encode the vocabulary-sized $softmax$ layer into a tree structure. Collobert and Weston (2008) propose a feed-forward neural network (C&W) which learns word embedding with a ranking-type cost. Mikolov et al. introduce the Recurrent neural network language models (RNNLMs) (Mikolov et al., 2010), Continuous Bag-of-Word (CBOW) and skip-gram model (Mikolov et al., 2013a) to learn embedding for words and phrases. Huang et al. (2012) propose a neural model to utilize the global context in addition to the local information. Besides utilizing neural networks to learn word embedding, some recent studies try the PCA-based algorithm to simplify the computation process (Lebret et al., 2013). The representation of words heavily relies on the characteristic of language.

The linguistic feature of English has been studied and used in the word embedding learning procedure. Specifically, Luong et al. (2013) utilize the morphological property of English word and incorporate the morphology into word embedding. In this paper, we focus on learning Chinese character embedding by exploiting the radical information of Chinese character, which is tailored for Chinese language. Unlike Luong et al. (2013) that initialize their model with the pre-trained embedding, we learn Chinese character embedding from scratch.

2.2 Word Embedding for NLP Tasks

Word embedding is able to capture the syntactic and semantic meanings of a word from massive corpora, which can reflect the discriminative features of data. Recently, word embedding has been successfully applied to a variety of NLP tasks, such as chunking, named entity recognition (Turian et al., 2010), POS tagging, semantic role labeling (Collobert et al., 2011), sentiment analysis (Socher et al., 2013b), paraphrase detection (Socher et al., 2011), parsing (Socher et al., 2013a) and Chinese word segmentation (Mansur et al., 2013; Zheng et al., 2013). For the task of Chinese word segmentation, Mansur et al. (2013) propose a feature-based neural language model for learning feature embedding. They develop a deep neural architecture which takes the embedding as input and tag the sequence. Zheng et al. (2013) present a neural architecture which combines embedding learning and sequence tagging in a unified model. The two studies on Chinese word segmentation utilize character embedding, yet they do not take the radical nature of Chinese language into consideration. Unlike previous studies, we leverage the radical information into the embedding learning process.

In this paper, we propose a neural network architecture tailored for Chinese character representation learning utilizing the radical information which is an typical characteristic of Chinese. We apply the learned embedding into a neural-CRF based Chinese word segmentation framework (Zheng et al., 2013) to verify its effectiveness. Neural-CRF is a sequential labeling framework that incorporates the representations of word (or character) into the CRF with a feed-forward neural network (detailed in Section 4). In the neural-CRF model, the word (or character) embeddings are treated as input features and the performance of further application highly depends on the quality of word (or character) representation.

3 Radical-Enhanced Model for Chinese Character Representation Learning

In this section, we describe the details of leveraging the radical information for learning Chinese character embedding. Based on C&W model (Collobert et al., 2011), we present a radical-enhanced model, which utilizes both radical and context information of characters. In the following subsections, we first briefly introduce the C&W model, and then present the details of our radical-enhanced neural architecture.

3.1 C&W Model

C&W model (Collobert et al., 2011) is proposed to learn the continuous representation of a word from its context words. Its training objective is to assign a higher score to the reasonable ngram than the corrupted ngram. The loss function of C&W is a ranking-type cost:

$$loss_c(s, s^w) = max(0, 1 - score(s) + score(s^w)) \quad (1)$$

where s is the reasonable ngram, s^w is the corrupted one with the middle word replaced by word w, and $score(.)$ represents the reasonability scalar of a given ngram, which can be calculated by its neural model.

C&W is a feed-forward neural network consisted of four layers, as illustrated in Figure 1(a). The input of C&W is a ngram composed of n words, and the output is a score which evaluates the reasonability of the ngram. Each word is encoded as a column vector in the embedding matrix $\mathbf{W}_e \in \mathbb{R}^{d \times |V|}$, where d is the dimension of the vector, and V is the vocabulary. The $lookup$ layer has a fixed window size n, and it maps each word of the input ngram into its embedding representation. The output $score(s)$ is computed as follows:

$$score(s) = \mathbf{W}_2\mathbf{a} + b_2 \quad (2)$$

$$\mathbf{a} = HTanh(\mathbf{W}_1[\mathbf{x}_1 ... \mathbf{x}_n] + \mathbf{b}_1) \quad (3)$$

where $[\mathbf{x}_1 ... \mathbf{x}_n]$ is the concatenation of the embedding vectors of words $x_1, ... x_n$, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, b_2$ are the weights and biases of the two linear layers, and function $HTanh(.)$ is the $HardTanh$ function. The parameters can be learned by minimizing the loss through stochastic gradient descent algorithm.

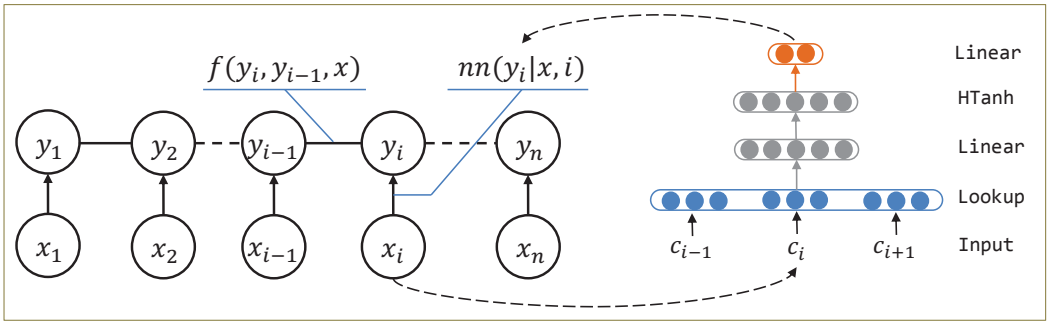

Figure 2: The neural CRF for Chinese word segmentation. Each input character x_i is denoted with its embedding vector, and $window(x_i)$ is the input of the neural network.

3.2 Radical-Enhanced Model

In this part, we present the radical-enhanced model for learning Chinese character embedding. Our model captures the radical information as well as the context information of characters.

The training objective of our radical-enhanced model contains two parts: 1) for a ngram, discriminate the correct middle character from the randomly replaced character; 2) for each character within a ngram, predict its radical. To this end, we develop a tailored neural architecture composed of two parts, context-based part and radical-based part, as given in Figure 1(b). The context-based part captures the context information, and the radical-based part utilizes the radical information. The final loss function of our model is shown as follows:

$$\begin{aligned} Loss(s, s^w) &= \alpha \cdot loss_c(s, s^w) + \\ &(1-\alpha) \cdot (\sum_{c \in s} loss_r(c) + \sum_{c \in s^w} loss_r(c)) \end{aligned} \quad (4)$$

where s is the correct ngram, s^w is the corrupted ngram, $loss_c(.)$ is the loss of the context-based part, $loss_r(.)$ is the loss of the radical-based part, and α linearly weights the two parts.

Specifically, the context-based part takes a ngram as input and outputs a score, as described in Equation 1. The radical-based part is a list of feed-forward neural networks with shared parameters, each of which is composed of three layers, namely $lookup \rightarrow linear \rightarrow softmax$ (from bottom to top). The unit number of each $softmax$ layer is equal to the number of radicals. Softmax layer is suitable for this scenario as its output can be interpreted as conditional probabilities. The cross-entropy loss of each softmax layer is defined as follows:

$$loss_r(c) = -\sum_{i=0}^{N} \boldsymbol{p}_i^g(c) \times log(\boldsymbol{p}_i(c)) \quad (5)$$

where N is the number of radicals; $\boldsymbol{p}^g(c)$ is the gold radical distribution of character c, with $\sum_i \boldsymbol{p}_i^g(c) = 1$; $\boldsymbol{p}(c)$ is the predicted radical distribution.

Model Training Our model is trained by minimizing the loss given in Equation 4 over the training set. The parameters are embedding matrix of Chinese characters, weights and biases of each linear layer. All the parameters are initialized with random values, and updated via stochastic gradient descent. Hyper-parameter α is tuned on the development set.

4 Neural CRF for Chinese Word Segmentation

It is widely accepted that Chinese Word Segmentation can be resolved as a character based tagging problem (Xue and others, 2003). In this paper, we treat word segmentation as a sequence tagging task, and assign characters with four possible boundary tags: "B" for a character at the beginning of a word, "I" for the characters inside a word, "E" for that at the end of a word, and "S" for the character which is a word itself (**?**).

4.1 Traditional CRF

Linear chain conditional random field (CRF) (Lafferty et al., 2001) is a widely used algorithm for Chinese word segmentation. Given an observation sequence $\vec{\mathbf{x}}$ and its gold tag sequence $\vec{\mathbf{y}}$, CRF models a conditional probability distribution

as follows,

$$P(\vec{\mathbf{y}}|\vec{\mathbf{x}}) = \frac{1}{Z}\prod_{C}\Psi_c(Y_c) = \frac{exp\ \phi(\vec{\mathbf{y}},\vec{\mathbf{x}})}{\sum_{\vec{\mathbf{y}}'} exp\ \phi(\vec{\mathbf{y}}',\vec{\mathbf{x}})} \tag{6}$$

where C is a maximum clique, $\Psi_C(Y_C)$ is the potential function which is defined as an exponential function, $exp\phi(\vec{\mathbf{y}},\vec{\mathbf{x}})$ is the product of potential function on all the maximum cliques, and Z is the normalization factor. Function $\phi(\vec{\mathbf{y}},\vec{\mathbf{x}})$ is defined as follows:

$$\phi(\vec{\mathbf{y}},\vec{\mathbf{x}}) = \sum_{i,k}\lambda_k t_k(y_{i-1},y_i,\vec{\mathbf{x}},i) + \sum_{i,l}\mu_l s_l(y_i,\vec{\mathbf{x}},i) \tag{7}$$

where t_k and s_l are feature functions, λ_k and μ_l are the corresponding weights.

4.2 Neural CRF

In this section, we apply the radical-enhanced character embedding for Chinese word segmentation. Instead of hand-crafting feature, we leverage the learned character embedding as features for Chinese word segmentation with Neural CRF (Turian et al., 2010; Zheng et al., 2013). The illustration of neural CRF is shown in Figure 2. Given an observation sequence $\vec{\mathbf{x}}$ and its gold tag sequence $\vec{\mathbf{y}}$, neural CRF models their conditional probability as follows,

$$P(\vec{\mathbf{y}}|\vec{\mathbf{x}}) = \frac{exp\ \phi(\vec{\mathbf{y}},\vec{\mathbf{x}})}{\sum_{\vec{\mathbf{y}}'} exp\ \phi(\vec{\mathbf{y}}',\vec{\mathbf{x}})} \tag{8}$$

where $\phi(\vec{\mathbf{y}},\vec{\mathbf{x}})$ is the potential function which is computed as follows,

$$\phi(\vec{\mathbf{y}},\vec{\mathbf{x}}) = \sum_{i}[f(y_i,y_{i-1},\vec{\mathbf{x}})\vec{\mathbf{w}}_1 + f(y_i,\vec{\mathbf{x}})\vec{\mathbf{w}}_2] \tag{9}$$

where $f(y_i,y_{i-1},\vec{\mathbf{x}})$ is a binary-valued indicator function reflecting the transitions between y_{i-1} and y_i, and $\vec{\mathbf{w}}_1$ is its associated weight. $f(y_i,\vec{\mathbf{x}})\vec{\mathbf{w}}_2$ reflects the correlation of the input $\vec{\mathbf{x}}$ and the i-th label y_i, which is calculated by a four-layer neural network as given in Figure 2. The neural network takes a ngram as input, and outputs a distribution over all possible tags, such as "B/I/E/S". The unit number of the top $linear$ layer is equal to the number of tags, and the output is computed as follows,

$$\mathbf{output} = \mathbf{W}_2\mathbf{a} + \mathbf{b}_2 \tag{10}$$

$$\mathbf{a} = HTanh(\mathbf{W}_1\mathbf{window}(c_i) + \mathbf{b}_1) \tag{11}$$

$$\mathbf{window}(c_i) = [\mathbf{c}_{i-m}...\mathbf{c}_i...\mathbf{c}_{i+m}] \tag{12}$$

where c_i is the current character, m is the window size, $\mathbf{window}(c_i)$ is the concatenation of the embeddings of c_i's context characters, $\mathbf{W}_1,\mathbf{W}_2,\mathbf{b}_1,\mathbf{b}_2$ are the weights and biases of the linear layers, $HTanh$ is the $HardTanh$ function.

The neural CRF is trained via maximizing the likelihood of $P(\vec{\mathbf{y}}|\vec{\mathbf{x}})$ over all the sentences in the training set. We use Viterbi algorithm (Forney Jr, 1973) in the decoding procedure.

5 Experiments

In this section, we evaluate the radical-enhanced character embedding on two tasks, Chinese character similarity judgement and Chinese word segmentation. We compare our model with C&W (Collobert et al., 2011) and word2vec[2] (Mikolov et al., 2013a), and learn Chinese character embedding with the same settings. To effectively train character embeddings, we randomly select one million sentences from the Sougou corpus[3]. We extract a radical mapping dictionary from an online Chinese dictionary[4], which contains 265 radicals and 20,552 Chinese characters. Each character listed in the radical dictionary is attached with its radical, such as ⟨吃(eat), 口(mouth)⟩. We empirically set the embedding size as 30, window size as 5, learning rate as 0.1, and the length of hidden layer as 30.

5.1 Chinese Character Similarity

In this part, we evaluate the effectiveness of character embedding through Chinese character similarity judgement in the embedding space. Due to the lack of public dataset in Chinese, we build an evaluation dataset manually.

In view of polysemy, we divide characters into different clusters according to their most frequently-used meanings. The dataset totally contains 26 categories and 988 characters. The evaluation metric is the accuracy of semantic consistency between each character and its top K nearest neighbors. The accuracy is calculated as follows,

$$Accuracy = \frac{1}{|S|}\sum_{c_i\in S}\frac{1}{K}\sum_{t_j\in top(c_i)}\delta(c_i,t_j) \tag{13}$$

[2]Available at https://code.google.com/p/word2vec/. We utilize Skip-Gram as baseline.

[3]http://www.sogou.com/labs/dl/c.html

[4]http://xh.5156edu.com/

where S is the dataset, c_i is a character, $top(c_i)$ is the top K nearest neighbors of c_i in the embedding space using cosine similarity. $\delta(c_i, t_j)$ is an indicator function which is equal to 1 if c_i and t_j have the same semantic category, and equal to 0 on the contrary. We set K=10 in the following experiment.

Results and Analysis Figure 3 shows the accuracy of our radical-enhanced model and baseline embedding learning algorithms on character similarity judgement. The $alpha$ on the x-axis is the weight of the context-based component in our radical-enhanced model. Our model with $alpha$=1.0 represents the C&W model. Results show that our radical-enhanced model outperforms C&W and word2vec consistently when $alpha$ is lower than 0.8. The reason is that our model can effectively leverage rich semantic information from radicals, which are not explicitly captured in the baseline embedding learning algorithms. We also find that the accuracy of our model decreases with the increase of $alpha$ because the impact of radical is larger with smaller $alpha$. The trend further verifies the effectiveness of radical information.

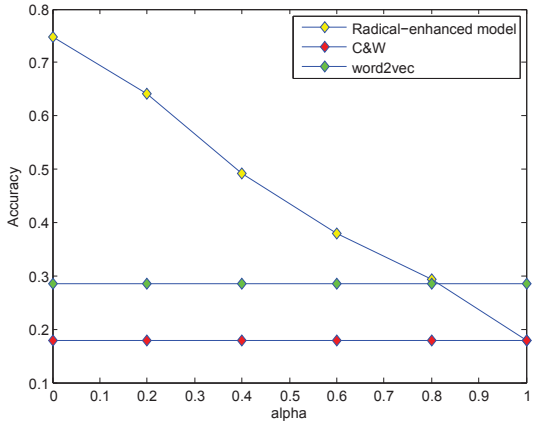

Figure 3: Accuracy of C&W, word2vec and our radical-enhanced model with K=10.

5.2 Chinese Word Segmentation

In this part, we apply character embedding as features for Chinese word segmentation using neural CRF. We conduct experiments on the widely-used Penn Chinese Treebanks 5 (CTB5) and CTB7. CTB5 is split according to (Jiang et al., 2008). CTB7 is split according to (Wang et al., 2011). The details of the datasets are given in Table 1.

	Training	Development	Test
CTB5	18,085	350	348
CTB7	31,088	10,036	10,291

Table 1: Statistics of the datasets of CTB5 and CTB7 for Chinese word segmentation.

The parameters of the neural CRF are empirically set as follows, the window size is 3, the hidden layer is set with 300 units, and the learning rate is set to 0.1. The evaluation criterion is Precision (P), Recall (R) and F1-score (F_1).

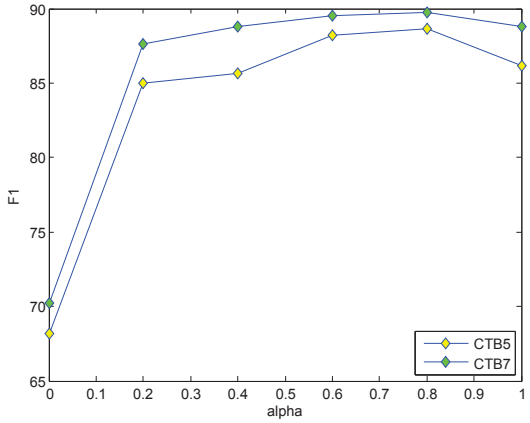

Figure 4: F_1 on the development set of CTB5 and CTB7. $alpha$ is the weight of the context-based loss in our model.

Results and Analysis Figure 4 gives the experiment results of our radical-enhanced model on the development set. The trends of our model are consistent on CTB5 and CTB7. Both performances increase sharply at $alpha = 0.2$ because context information is crucial for this sequential labeling task yet not utilized in the purely radical-driven model ($alpha = 0$). The best performances are achieved with $alpha$ in the range of [0.6,0.8]. Figure 3 and Figure 4 have different trends because of the different characteristics of the two tasks. For character similarity judgement, radical is dominant because it reflects the character category information. But for Chinese word segmentation, contexts also play an important role.

We compare our radical-enhanced model ($alpha$=0.8 tuned on the development set) with C&W model and Word2Vec in the framework of Neural CRF. Table 2 shows that, our model obtains better P, R and F_1 than C&W and word2vec on both CTB5 and CTB7. One reason is that the radial-enhanced model is capable

Method	CTB5			CTB7		
	P	R	F_1	P	R	F_1
NeuralCRF(C&W)	0.9215	0.9306	0.9260	0.8956	0.8974	0.8965
NeuralCRF(word2vec)	0.9132	0.9257	0.9194	0.8910	0.8896	0.8903
NeuralCRF(Our model)	**0.9308**	**0.9451**	**0.9379**	**0.9047**	**0.9022**	**0.9034**
CRF(character)	0.9099	0.9141	0.9120	0.8805	0.8769	0.8787
CRF(character + radical)	0.9117	0.9153	0.9135	0.8816	0.8778	0.8797

Table 2: Comparison of F_1 on the test set of CTB5 and CTB7.

to capture the semantic connections between characters with the same radical, which usually have similar semantic meaning and grammatical usage yet not explicitly modeled in C&W and word2vec. Another reason is that, the embeddings of lower-frequent characters are typically not well estimated by C&W and word2vec due to the lack of syntactic contexts. In the radical-enhanced model, their radicals bring important semantic information thus we obtain better embedding results. We also compare with two CRF-based baseline methods. $CRF(character)$ is the use of linear-chain CRF with character as its feature. In $CRF(character + radical)$, we utilize the radical information and the character as features with linear-chain CRF. Results of $CRF(character)$ and $CRF(character+radical)$ show that simply using radical as feature does not obtain significant improvement. Our radical-enhanced method outperforms two CRF-based baselines on both datasets, which further verifies the effectiveness of our method.

6 Conclusion

In this paper, we propose to leverage radical for learning the continuous representation of Chinese characters. To our best knowledge, this is the first work on utilizing the radical information of character for Chinese computational processing. A dedicated neural architecture with a hybrid loss function is introduced to effectively integrate radical information for learning character embedding. Our radical-enhanced model is capable to capture the semantic connections between characters from both syntactic contexts and the radical information. The effectiveness of our method has been verified on Chinese character similarity judgement and Chinese word segmentation. Experiment results on both tasks show that, our method outperforms two widely-accepted embedding learning algorithms, which do not utilize the radical information.

References

[Bengio et al.2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

[Bengio2013] Yoshua Bengio. 2013. Deep learning of representations: Looking forward. *arXiv preprint arXiv:1305.0445*.

[Blei et al.2003] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Bordes et al.2011] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

[Collobert and Weston2008] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

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

[Deerwester et al.1990] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

[Forney Jr1973] G David Forney Jr. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

[Huang et al.2012] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

[Jiang et al.2008] Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Citeseer.

[Lafferty et al.2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[Lebret et al.2013] Rémi Lebret, Joël Legrand, and Ronan Collobert. 2013. Is deep learning really necessary for word embeddings? *NIPS*.

[Luong et al.2013] Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. *CoNLL-2013*, page 104.

[Mansur et al.2013] Mairgup Mansur, Wenzhe Pei, and Baobao Chang. 2013. Feature-based neural language model and chinese word segmentation. *IJCNLP*.

[Mikolov et al.2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ICLR*.

[Mikolov et al.2013b] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

[Mnih and Hinton2008] Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.

[Morin and Bengio2005] Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252.

[Socher et al.2011] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

[Socher et al.2013a] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013a. Parsing with compositional vector grammars. In *ACL*.

[Socher et al.2013b] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. EMNLP.

[Turian et al.2010] J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. *ACL*.

[Turney et al.2010] Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

[Wang et al.2011] Yiou Wang, Yoshimasa Tsuruoka Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In *IJCNLP*, pages 309–317.

[Xue and others2003] Nianwen Xue et al. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

[Zhang and Clark2010] Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In *Proceedings of the EMNLP*, pages 843–852, Cambridge, MA, October.

[Zhang et al.2013] Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2013. Chinese parsing exploiting characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 125–134, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Zheng et al.2013] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. *EMNLP*.