Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference

# Radical-Based Hierarchical Embeddings for Chinese Sentiment Analysis at Sentence Level

**Haiyun Peng, Erik Cambria**
School of Computer Science and Engineering
Nanyang Technological University
{peng0065,cambria}@ntu.edu.sg

**Xiaomei Zou**
College of Computer Science and Technology
Harbin Engineering University
zouxiaomei@hrbeu.edu.cn

**Abstract**

Text representation in Chinese sentiment analysis is usually working at word or character level. In this paper, we prove that radical-level processing could greatly improve sentiment classification performance. In particular, we propose two types of Chinese radical-based hierarchical embeddings. The embeddings incorporate not only semantics at radical and character level, but also sentiment information. In the evaluation of our embeddings, we conduct Chinese sentiment analysis at sentence level on four different datasets. Experimental results validate our assumption that radical-level semantics and sentiments can contribute to sentence-level sentiment classification and demonstrate the superiority of our embeddings over classic textual features and popular word and character embeddings.

## Introduction

For every natural language processing (NLP) task, text representation is always the first step. In English, words are segmented by spaces and they are naturally taken as basic morphemes in text representation. Then, word embeddings were born based on distributed hypothesis.

Unlike English, whose fundamental morpheme is a combination of characters, such as prefixes, words etc., the fundamental morpheme of Chinese is radical, which is a (graphic) component of Chinese characters. Each Chinese character can contain up to five radicals. The radicals within character have various relative positions. For instance, it could be left-right (‘ 蛤 (toad) ’,‘ 秒 (second) ’), up-down (‘ 岗 (hill) ’, ‘ 孬 (not good) ’), inside-out (‘ 国 (country) ’, ‘ 问 (ask) ’) etc.

The point of their existence is not only decorative but also functional. Radicals have two main functions: pronunciation and meaning. As the aim of this work is sentiment prediction, we are more interested in the latter function. For example, the radical ‘ 疒 ’ carries the meaning of disease. Any Chinese character containing this radical is related with disease and, hence, tends to express negative sentiment, such as ‘ 病 (illness) ’, ‘ 疯 (madness) ’, ‘ 瘫 (paralyzed) ’, etc. In order to utilize this semantic and sentiment information among radicals, we decide to map radicals to embeddings (numeric representation at lower dimension).



The reason why we chose embeddings rather than classic textual feature like ngram, POS, etc. is because the embedding method is based on the distributed hypothesis, which greatly explores the semantics and relies on token sequences. Correspondingly, radicals alone may not carry enough semantic and sentiment information. It is only when they are placed in a certain order that their connection with sentiment begins to reveal (Poria et al. 2017).

To the best of our knowledge, no sentiment-specific radical embeddings have ever been proposed before this work. We firstly train a pure radical embedding named $R_{semantic}$, hoping to capture semantics between radicals. Then, we train a sentiment-specific radical embedding and integrate it with the $R_{semantic}$ to form a radical embedding termed $R_{sentic}$, which encodes both semantic and sentiment information (Cambria et al. 2016; Poria et al. 2016). With the above, we integrate the two obtained radical embeddings with Chinese character embedding to form the radical-based hierarchical embedding, termed $H_{semantic}$ and $H_{sentic}$, respectively.

The rest of the paper is organized as follows: the first section illustrates general word embedding methods and Chinese radical embedding; the second section presents a detailed analysis of Chinese characters and radicals via decomposition; the third section introduces our hierarchical embedding models; the fourth section demonstrates experimental evaluations of the proposed methods; finally, the fifth section concludes the paper and suggests a few future improvements.

## Related Works

### General Embedding Methods

One-hot representation is the initial numeric word representation method in NLP. However, it usually leads to a problem of high dimensionality and sparsity. To solve this problem, distributed representation or word embedding was proposed (Turian, Ratinov, and Bengio 2010). Word embedding is a representation which maps words into low dimensional vectors of real numbers by using neural networks. The key idea is based on distributional hypothesis so as to model how to represent context words and the relation between context words and target word. Thus, language model is a natural solution. Bengio et al. (Bengio et al. 2003) introduced neural network language model (NNLM) in 2001.

Instead of using counts to model ngram language model, they built a neural network. Word embeddings are the byproducts of building the language model. I n 2007, Mnih and Hinton proposed a log-bilinear language model (LBL) (Mnih and Hinton 2007) which is built upon NNLM and later upgraded to hierarchical LBL (HLBL) (Mnih and Hinton 2009) and inverse vector LBL (ivLBL) (Mnih and Kavukcuoglu 2013). Instead of modeling ngram model like the above, Mikolov et al. (Mikolov et al. 2010) proposed a model based on recurrent neural networks to directly estimate the probability of target words given contexts.

Since the introduction of the C&W model (Collobert and Weston 2008) in 2008, people started to design models whose objectives are no longer the language model but the word embedding itself. C&W places the target word in the input layer, and output only one node which denotes the likelihood of the input words' sequence. Later in 2013, Mikolov et al. (Mikolov et al. 2013a) introduced the continuous bag-of-words model (CBOW), which places context words in the input layer and target word in the output layer, and Skip-gram model, which swaps the input and output in CBOW. They also proposed negative sampling which greatly speeds up training.

### Chinese Radical Embedding

(Chen et al. 2015) started to decompose Chinese words into characters and proposed a character-enhanced word embedding model (CWE). (Sun et al. 2014) started decompose Chinese characters to radicals and developed a radical-enhanced Chinese character embedding. However, they only selected one radical from each character to enhance the embedding. (Shi et al. 2015) began to train pure radical-based embedding for short-text categorization, Chinese word segmentation and web search ranking. Yin et al. extend the pure radical embedding in (Mikolov et al. 2013b) by introducing multi-granularity Chinese word embeddings. However, none of the above embeddings have considered incorporating sentiment information and apply the radical embeddings to the task of sentiment classification (Cambria 2016). To bridge such a gap, in this paper we develop radical-based hierarchical Chinese embeddings specifically for sentiment analysis.

## Decomposition of Chinese Characters

Chinese written language dates back to 1200-1050 BC from the Shang dynasty. It originates from an Oracle bone script, which was iconic symbols engraved on 'dragon bones'. From this time on was the first stage of Chinese written language development, Chinese written language was completely pictogram. However, different areas within China maintained different set of writing systems.

The second stage started from the unification in Qin dynasty. Seal script, which was an abstraction of the pictogram, became dominating over the empire from then on. Another apparent characteristic during this time was new Chinese characters were invented by combinations of existing and evolved characters. Under the mixed influence of foreign culture, development of science and technology and the evolution of social life, a great deal of Chinese characters were created during this time.

One feature of these characters is that they are no longer pictograms, but they are decomposable. Each of the decomposed elements (or radicals) carries a certain function. For instance, '声旁 (phoneme)' labels the pronunciation of this character and '形旁 (morpheme)' symbolizes the meaning of this character. Further details will be discussed in the following section.

The third stage occurred in the middle of the last century when the central government started advocating simplified Chinese. The old characters were simplified by reducing certain strokes within the character. The simplified Chinese characters dominate over mainland China ever since. Only Hong Kong, Taiwan and Macau retain the traditional Chinese characters.

### Chinese Radicals

Due to the second stage of the above discussion, all modern Chinese character can be decomposed to radicals. Radicals are graphical components of characters. Some of the radicals in the character acts like phonemes. For example, the radical '丙' appears in the right half of character '柄 (handle)' and symbolizes the pronunciation of this character. People even sometimes can correctly predict the pronunciation of a Chinese character which he or she does not know by recognizing certain radicals inside.

Some other radicals in the character act like morphemes that carry the semantic meaning of the character. For example, '木 (wood)' itself is both a character and a radical. It means wood. A character '林 (jungle)' which is made up of two '木' means jungle. A character '森 (forest)' which is made up of three '木' means forest. In another example, radical '父' is a formal form of word 'father'. It appears on top of character '爸' and this character means father exactly, but less formal, like 'dad' in English.

Moreover, the meaning of a character could be concluded from a integration of its radicals. A good example given by (Shi et al. 2015) is character '朝'. This character is made up of '十', '日', '十' and '月' four radicals. These four radicals are evolved from pictograms. '十' stands for grass. '日' stands for the sun. '月' stands for the moon. The integration of these four means the sun replaces the moon on the grass land, which is essentially the word 'morning'. Not surprisingly, the meaning of this character '朝' is indeed morning. This could continue. If the radical '氵' which means water was attached to the left of character '朝', then it is another character '潮'. Literally, this character means the water coming up in the morning. In fact, this '潮' means tide, which matches its literal meaning.

To conclude, radicals entail more information than characters alone. Character-level research can only study the semantics expressed by characters. However, deeper semantic information and clues could be found at radical-level analysis (Peng, Cambria, and Hussain 2017). This motivates us to apply deep learning techniques to extract this information. Prior to that, as we discussed in the related works, most works are in English language. Since English is very different from Chinese in many aspects, especially in decomposition, we conduct a comparison in Table 1.

| English | | Chinese | |
|---|---|---|---|
| Hierarchy | Example | Hierarchy | Example |
| Character | a, b, ... , y, z | Radical | 艹, 宀, 彳... |
| Word | sometimes, Naïve | Character (Single-character word) | 资, 词, 不 |
| Phrase | pay debt, good luck | Multi-character word (Phrase) | 蛤蟆, 一颗赛艇 |
| Sentence | Today is a good day. | Sentence | 很惭愧, 只做了一点微小的工作。 |

Table 1: Comparison between English and Chinese in composition

As we could see from Table 1, character level is the minimum composition level in English. However, the equivalent level in Chinese is one level down than character, which is radical level. Unlike English, semantics are hidden within each character in Chinese. Secondly, Chinese word can be made up of single character or multi-character. Moreover, there is no space between words in Chinese sentence. All the above observations indicate that normal English word embedding can not be directly applied to Chinese. Extra processing like word segmentation, which will introduce errors, need to be conducted first.

Furthermore, if a new word or even a new character is out-of-vocabulary (OOV), normal word-level or character-level have no reasonable solution except giving a random vector. In order to address the above issues and also to extract the semantics within Chinese characters, a radical-based hierarchical Chinese embedding method is proposed in this paper.

## Hierarchical Chinese Embedding

In this section, we firstly introduce the deep neural network used in training our hierarchical embeddings. Then, we discuss our radical embedding. Finally, we present the hierarchical embedding model.

### Skip-Gram Model

We employ the Skip-gram neural embedding model proposed by (Mikolov et al. 2013a) together with the negative sampling optimization technique in (Yin et al. 2016). In this section, we briefly summarize the training objective and the model. Skip-gram model can be understood as a one-word context version CBOW model (Mikolov et al. 2013a) working over $C$ panels, where $C$ is the number of context words of target word. Opposite to CBOW model, the target word is at input layer whereas context words are at the output layer. By generating the most probable context words, the weight matrix can be trained and embedding vectors can be extracted.

Specifically, it is a one hidden layer neural network (Rong 2014). For each input word, it was denoted with an input vector $V_{wi}$. The hidden layer is defined as:

$$h = V_{\text{wi}}{}^{\text{T}}$$

where $h$ is the hidden layer, $wi$ is the $i$th row of input-hidden weight matrix $W$. At the output layer, $C$ multinomial distributions were output, given each of the output is computed with the hidden-output matrix as:

$$p(w_{c,j} = w_{O,c}|w_i) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

where $w_{c,j}$ is the $j$th word on the $c$th panel of the output layer; $w_{O,c}$ is the $c$th word in the output context words; $w_i$ is the input word vector; $y_{c,j}$ is the output of the $j$th unit on the $c$th panel of the output layer; $u_{c,j}$ is the net input of the $j$th unit on the $c$th panel of the output layer. Furthermore the objective function is to maximize the formula below:

$$\sum_{(w,c)\in\mathbb{D}} \sum_{w_j \in c} \text{logP}(w|w_j)$$

where $w_j$ is the $j$th word in contexts $c$, given the target word $w$.

### Radical-Based Embedding

Traditional radical researches like (Sun et al. 2014) only take out one radical from each character to improve the Chinese character embedding. Moreover, to the best of our knowledge, no sentiment-specific Chinese radical embedding has ever been proposed yet. Thus, we propose the following two radical embeddings for Chinese sentiment analysis.

Inspired by the facts that Chinese characters can be decomposed to radicals and these radicals carry semantic meanings, we directly break characters into radicals and concatenate them in the order from left to right. We treat the radicals as the fundamental units in texts. Specifically, for any sentence we decompose each character into its radicals and concatenate these radicals from different characters as a new radical string. Then we did the above preprocessing to all sentences in the corpus. Finally, a radical-level embedding model is built on this radical corpus using skip-gram model. We call this type of radical embedding as semantic radical embedding ($R_{\text{semantic}}$), because the major information extracted from this type of corpus is semantic between radicals. In order to extract the sentiment information between radicals, we developed the second type radical embedding which is sentic radical embedding ($R_{\text{sentic}}$).

After studying the radicals, we have found that radicals themselves do not convey much sentiment information. What carries the sentiment information is the sequence or combination of different radicals. Thus, we take advantages of existing sentiment lexicons as our resource to study the sequence. Like we did before, we collect all the sentiment words from two different popular Chinese sentiment lexicons, Hownet (Dong and Dong 2006) and NTUSD (Ku, Liang, and Chen 2006) and break them into radicals. Then we employ skip-gram model to learn the sentiment related radical embedding ($R_{\text{sentiment}}$).

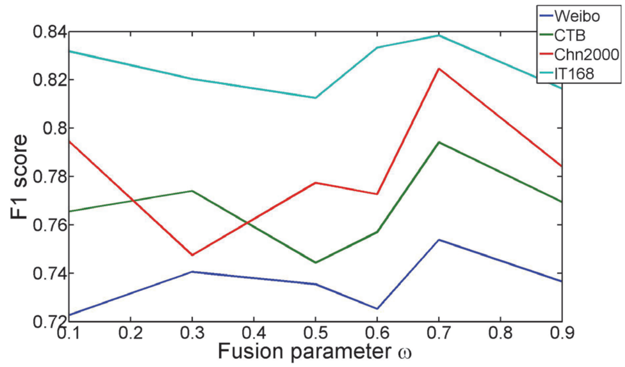


Figure 1: Performance on four datasets at different fusion parameter

Since we want the radical embedding have both semantic information and sentiment information, we therefore conduct a fusion process of the previous two embeddings. The fusion formula is given as:

$$R_{sentic} = (1 - \omega) \cdot R_{semantic} + \omega \cdot R_{sentiment}$$

where $R_{sentic}$ is the resulting radical embedding that integrates both semantic and sentiment information; $R_{semantic}$ is the semantic embedding and $R_{sentiment}$ is the sentiment embedding; $w$ is the weight of the fusion. If $w$ equals to 0, then the $R_{sentic}$ is pure semantic embedding. If $w$ equals to 1, then the $R_{sentic}$ is pure sentiment embedding.

In order to find the best fusion parameter, we conduct tests on separated development subsets of four real Chinese sentiment datasets, namely: Chn2000, It168, Chinese Treebank (Li et al. 2014) and Weibo dataset (details in next section). We train a convolutional neural network (CNN) to classify the sentiment polarity of sentences in the datasets. The features we use are the sentic radical embedding, but we apply the features at different fusion parameter value. The classification accuracies of different fusion values on four datasets are shown in Fig. 1. As the heuristics from Fig. 1 suggest, we take the fusion parameter of value 0.7 which performs best.

### Hierarchical Embedding

Hierarchical embedding is based on the assumption that different level of embeddings will capture different level of semantics. According to the hierarchy of Chinese in Table 1, we have already explored the semantics as well as sentiment at radical level. The next higher level is character level, followed by word level (multi-character word). However, we only select character-level embedding ($C_{semantic}$) to be integrated in our hierarchical model because characters are naturally segmented by Unicode (no pre-processing or segmentation needed). Although existing Chinese word segmenter could achieve certain accuracy, it can still introduce segmentation errors and thus affect the performance of word embedding. In the hierarchical model, we also use skip-gram model to train independent Chinese character embeddings. Then we fuse the character embeddings with either the semantic radical embedding ($R_{semantic}$) and the sentic radical embedding ($R_{sentic}$) to form two types of hierarchical embeddings: $H_{semantic}$ and $H_{sentic}$, respectively. The fusion formula is the same with that in radical embeddings, except that with a different fusion parameter value of 0.5 based on our development tests. A graphical illustration of the hierarchical model is depicted in Fig. 2.

## Experimental Evaluation

We evaluate our proposed method on Chinese sentence-level sentiment classification task in this section. Firstly, we introduce the datasets used for evaluations. Then, we demonstrate the experimental settings. Lastly, we present the experimental results and provide an interpretation for them.

### Dataset

There are four sentence-level Chinese sentiment datasets used in our experiments. The first is Weibo dataset (Weibo) which is a collection of Chinese micro blogs from NLP&CC, with about 2000 blogs for either positive or negative category. The second dataset is a Chinese Tree Bank (CTB) introduced by (Li et al. 2014). For each sentiment category, we have obtained over 19000 sentences after mapping their sentiment values to polarity. The third dataset Chn2000, contains about 2000 hotel reviews from customers[1]. The last dataset IT168, have around 1000 digital product reviews[2]. All the above datasets are labeled as positive or negative at sentence level. In order to prevent overfitting, we conduct 5-fold cross validations on all our experiments.

### Experimental Setting

As embedding vectors are usually used as features in classification tasks, we compare our proposed embeddings with three baseline features: character-bigram, word embeddings and character embeddings. In choosing the classification model, we take advantage of state-of-the-art machine learning toolbox scikit-learn (Pedregosa, Varoquaux, and et al. 2011).

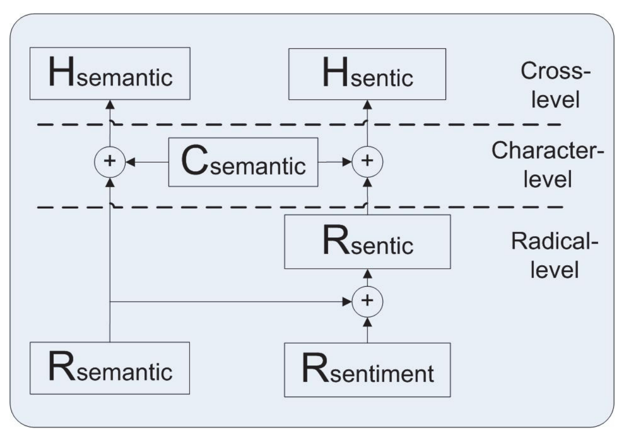


Figure 2: Framework of hierarchical embedding model

[1]http://searchforum.org.cn/tansongbo/corpus
[2]http://product.it168.com

| | | Bigram(%) | | | Rsemantic(%) | | | Rsentic(%) | | | Hsemantic(%) | | | Hsentic(%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Weibo | LSVC | 71.65 | 71.60 | 71.58 | 66.77 | 66.64 | 66.57 | 67.46 | 67.35 | 67.30 | 71.02 | 70.94 | 70.91 | 72.74 | 72.66 | 72.63 |
| | LR | 74.38 | 74.32 | 74.30 | 65.51 | 65.28 | 65.15 | 65.29 | 65.12 | 65.02 | 70.39 | 70.31 | 70.27 | 72.47 | 72.37 | 72.33 |
| | NB | 63.84 | 63.01 | 62.15 | 57.60 | 55.74 | 52.90 | 58.67 | 56.73 | 54.21 | 59.16 | 55.97 | 51.74 | 60.42 | 57.63 | 54.58 |
| | MLP | 72.54 | 72.50 | 72.48 | 67.02 | 66.93 | 66.89 | 67.31 | 67.25 | 67.22 | 70.53 | 70.49 | 70.47 | 73.03 | 73.00 | 72.99 |
| | CNN | - | - | - | 75.27 | 73.71 | 73.19 | 75.44 | 75.41 | 75.38 | 73.88 | 72.91 | 72.55 | 75.82 | 75.60 | **75.58** |
| CTB | LSVC | 76.45 | 76.32 | 76.29 | 67.22 | 67.19 | 67.17 | 66.34 | 66.28 | 66.25 | 68.57 | 68.55 | 68.54 | 69.15 | 69.11 | 69.10 |
| | LR | 78.12 | 77.99 | 77.97 | 65.29 | 65.25 | 65.22 | 64.91 | 64.85 | 64.81 | 68.25 | 68.22 | 68.21 | 69.24 | 69.20 | 69.19 |
| | NB | 66.60 | 62.80 | 60.46 | 60.99 | 60.43 | 59.86 | 60.41 | 59.59 | 58.64 | 61.24 | 60.04 | 58.90 | 63.52 | 62.50 | 61.74 |
| | MLP | 76.13 | 76.01 | 75.98 | 67.71 | 67.68 | 67.66 | 66.92 | 66.79 | 66.72 | 70.98 | 70.96 | 70.95 | 70.01 | 69.78 | 69.69 |
| | CNN | - | - | - | 77.68 | 77.67 | 77.65 | 79.59 | 79.42 | 79.42 | 80.77 | 80.77 | **80.76** | 80.79 | 80.69 | 80.65 |
| Chn2000 | LSVC | 82.43 | 82.21 | 82.22 | 70.64 | 67.32 | 67.12 | 66.00 | 61.26 | 59.70 | 73.73 | 72.74 | 72.87 | 74.57 | 73.61 | 73.71 |
| | LR | 83.22 | 82.68 | 82.76 | 69.99 | 55.50 | 48.04 | 68.50 | 51.34 | 39.51 | 70.62 | 67.65 | 67.50 | 72.38 | 68.24 | 67.86 |
| | NB | 67.06 | 66.68 | 65.68 | 67.23 | 66.93 | 66.54 | 63.34 | 63.36 | 62.95 | 64.93 | 64.44 | 64.33 | 67.92 | 67.75 | 67.59 |
| | MLP | 80.71 | 80.42 | 80.47 | 69.00 | 68.57 | 68.59 | 67.47 | 66.85 | 66.83 | 74.00 | 73.63 | 73.62 | 73.23 | 73.06 | 73.05 |
| | CNN | - | - | - | 79.96 | 81.83 | 80.14 | 82.01 | 83.50 | 82.47 | 87.45 | 86.71 | **87.02** | 86.06 | 87.07 | 86.12 |
| IT168 | LSVC | 81.95 | 82.06 | 81.93 | 72.53 | 70.23 | 70.18 | 72.72 | 69.85 | 69.71 | 79.55 | 79.00 | 79.11 | 80.77 | 80.30 | 80.44 |
| | LR | 83.86 | 83.72 | 83.74 | 71.40 | 60.82 | 57.32 | 73.58 | 56.10 | 48.47 | 77.58 | 75.46 | 75.71 | 79.46 | 76.80 | 77.09 |
| | NB | 63.84 | 63.01 | 62.15 | 64.73 | 63.62 | 63.45 | 63.50 | 62.62 | 62.46 | 67.75 | 66.21 | 66.12 | 71.90 | 70.09 | 70.16 |
| | MLP | 83.35 | 83.35 | 83.29 | 71.83 | 71.04 | 71.08 | 73.86 | 72.71 | 72.80 | 78.10 | 77.70 | 77.68 | 79.48 | 79.31 | 79.27 |
| | CNN | - | - | - | 84.38 | 84.33 | **84.33** | 83.95 | 83.87 | 83.83 | 85.39 | 84.50 | 84.07 | 83.75 | 83.43 | 83.15 |

Table 2: Comparison with traditional feature on four datasets

| | | W2V(%) | | | C2V(%) | | | Rsemantic(%) | | | Rsentic(%) | | | Hsemantic(%) | | | Hsentic(%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Weibo | LSVC | 74.46 | 74.38 | 74.35 | 74.12 | 73.98 | 73.94 | 66.77 | 66.64 | 66.57 | 67.46 | 67.35 | 67.30 | 71.02 | 70.94 | 70.91 | 72.74 | 72.66 | 72.63 |
| | LR | 73.91 | 73.72 | 73.66 | 73.60 | 73.43 | 73.37 | 65.51 | 65.28 | 65.15 | 65.29 | 65.12 | 65.02 | 70.39 | 70.31 | 70.27 | 72.47 | 72.37 | 72.33 |
| | NB | 60.63 | 57.97 | 55.15 | 61.04 | 58.08 | 55.02 | 57.60 | 55.74 | 52.90 | 58.67 | 56.73 | 54.21 | 59.16 | 55.97 | 51.74 | 60.42 | 57.63 | 54.58 |
| | MLP | 73.68 | 73.58 | 73.55 | 74.49 | 74.43 | 74.41 | 67.02 | 66.93 | 66.89 | 67.31 | 67.25 | 67.22 | 70.53 | 70.49 | 70.47 | 73.03 | 73.00 | 72.99 |
| | CNN | 72.57 | 72.55 | 72.52 | 75.15 | 75.11 | 75.11 | 75.27 | 73.71 | 73.19 | 75.44 | 75.41 | 75.38 | 73.88 | 72.91 | 72.55 | 75.82 | 75.60 | **75.58** |
| CTB | LSVC | 71.15 | 71.12 | 71.11 | 68.92 | 68.90 | 68.90 | 67.22 | 67.19 | 67.17 | 66.34 | 66.28 | 66.25 | 68.57 | 68.55 | 68.54 | 69.15 | 69.11 | 69.10 |
| | LR | 70.87 | 70.84 | 70.83 | 68.50 | 68.48 | 68.47 | 65.29 | 65.25 | 65.22 | 64.91 | 64.85 | 64.81 | 68.25 | 68.22 | 68.21 | 69.24 | 69.20 | 69.19 |
| | NB | 67.56 | 67.51 | 67.49 | 63.49 | 62.61 | 61.96 | 60.99 | 60.43 | 59.86 | 60.41 | 59.59 | 58.64 | 61.24 | 60.04 | 58.90 | 63.52 | 62.50 | 61.74 |
| | MLP | 71.17 | 71.16 | 71.15 | 69.78 | 69.54 | 69.44 | 67.71 | 67.68 | 67.66 | 66.92 | 66.79 | 66.72 | 70.98 | 70.96 | 70.95 | 70.01 | 69.78 | 69.69 |
| | CNN | 78.56 | 78.56 | 78.56 | 78.56 | 77.93 | 77.75 | 77.68 | 77.67 | 77.65 | 79.59 | 79.42 | 79.42 | 80.77 | 80.77 | **80.76** | 80.79 | 80.69 | 80.65 |
| Chn2000 | LSVC | 81.05 | 79.77 | 80.05 | 72.04 | 70.73 | 70.85 | 70.64 | 67.32 | 67.12 | 66.00 | 61.26 | 59.70 | 73.73 | 72.74 | 72.87 | 74.57 | 73.61 | 73.71 |
| | LR | 78.87 | 74.74 | 74.96 | 70.32 | 64.29 | 63.00 | 69.99 | 55.50 | 48.04 | 68.50 | 51.34 | 39.51 | 70.62 | 67.65 | 67.50 | 72.38 | 68.24 | 67.86 |
| | NB | 72.25 | 71.25 | 71.34 | 69.62 | 69.55 | 69.44 | 67.23 | 66.93 | 66.54 | 63.34 | 63.36 | 62.95 | 64.93 | 64.44 | 64.33 | 67.92 | 67.75 | 67.59 |
| | MLP | 79.53 | 79.18 | 79.24 | 70.84 | 70.65 | 70.67 | 69.00 | 68.57 | 68.59 | 67.47 | 66.85 | 66.83 | 74.00 | 73.63 | 73.62 | 73.23 | 73.06 | 73.05 |
| | CNN | 82.50 | 82.50 | 82.50 | 85.77 | 86.21 | 85.95 | 79.96 | 81.83 | 80.14 | 82.01 | 83.50 | 82.47 | 87.45 | 86.71 | **87.02** | 86.06 | 87.07 | 86.12 |
| IT168 | LSVC | 82.43 | 81.15 | 81.46 | 78.68 | 77.80 | 78.00 | 72.53 | 70.23 | 70.18 | 72.72 | 69.85 | 69.71 | 79.55 | 79.00 | 79.11 | 80.77 | 80.30 | 80.44 |
| | LR | 82.11 | 77.73 | 78.11 | 77.79 | 72.69 | 72.67 | 71.40 | 60.82 | 57.32 | 73.58 | 56.10 | 48.47 | 77.58 | 75.46 | 75.71 | 79.46 | 76.80 | 77.09 |
| | NB | 60.63 | 57.97 | 55.15 | 71.12 | 69.78 | 69.89 | 64.73 | 63.62 | 63.45 | 63.50 | 62.62 | 62.46 | 67.75 | 66.21 | 66.12 | 71.90 | 70.09 | 70.16 |
| | MLP | 79.93 | 79.65 | 79.70 | 78.52 | 78.36 | 78.35 | 71.83 | 71.04 | 71.08 | 73.86 | 72.71 | 72.80 | 78.10 | 77.70 | 77.68 | 79.48 | 79.31 | 79.27 |
| | CNN | 82.23 | 81.50 | 81.40 | 82.69 | 82.63 | 82.65 | 84.38 | 84.33 | **84.33** | 83.95 | 83.87 | 83.83 | 85.39 | 84.50 | 84.07 | 83.75 | 83.43 | 83.15 |

Table 3: Comparison with embedding features on four datasets

Four classic machine learning classifiers were applied in our experiments: LinearSVC (LSVC), logistic regression (LR), Naïve Bayes classifier with a gaussian kernel (NB) and multi-layer perceptron (MLP) classifier. In evaluating the embedding features on these classic machine learning classifiers, an average embedding vector is computed to represent each sentence, given certain granularity of the sentence cells. For instance, if a sentence is broken into a string of radicals, then the radical embedding vector of this sentence is the arithmetic mean (average) of its component radical embeddings. Furthermore, we also apply CNN in the same way proposed in (Kim 2014), except that we reduce the embedding vector dimension to 128.

## Results and Discussion

Table 2 compared bigram feature with semantic radical embedding, sentic radical embedding, semantic hierarchical embedding and sentic hierarchical embedding using five classification models on four different datasets. Similarly, Table 3 also compared the proposed embedding features with word2vec and character2vec features. In all of the four datasets, our proposed features working with CNN classifier achieved the best performance. In Weibo dataset, sentic hierarchical embedding performed slightly better than character2vec, with less than 1% improvement. However in CTB and Chn2000 datasets, semantic hierarchical beat three baseline features by 2~6%. In the IT168 dataset, the sentic hierarchical embedding was second to bigram feature in MLP model.

This result was not surprising because bigram feature can be understood as a sliding window with size of 2. Using the multi-layer perceptron classifier, the performance could be parallel to that of a CNN classifier. Even though, the other three proposed features working with CNN classifier beat all baseline features with any classifier. In addition to the above observations, we also obtained the following analysis.

Firstly, deep learning classifiers worked best on embedding features. The performance of all embedding features reduced sharply when applying on classic classifiers. Nevertheless, even if the performance of our proposed features in classic machine learning classifiers dropped greatly compared with CNN, they still paralleled or beat other baseline features. Moreover, the performance of the proposed features were never fine-tuned. Better performance can be expected after future fine tuning.

Secondly, the proposed embedding features do unveil certain information that can promote sentence-level sentiment analysis. Although we were not certain where exactly the extra information located, because the performance of our four proposed embedding features were not robust (no single one feature achieved the best performance over all four datasets), we proved radical-level embedding contribute to Chinese sentiment analysis.

## Conclusion

In this paper, we proposed Chinese radical-based hierarchical embeddings particularly designed for sentiment analysis. Four types of radical-based embeddings were introduced: radical semantic embedding, radical sentic embedding, hierarchical semantic embedding and hierarchical sentic embedding. By conducting sentence-level sentiment classification experiments on four Chinese datasets, we proved the proposed embeddings outperform state-of-the-art textual and embedding features. Most importantly, our study presents the first piece of evidence that Chinese radical-level and hierarchical-level embeddings can improve the Chinese sentiment analysis.

Meanwhile, this paper also suggests a few directions of future work. Firstly, as we only fused different embeddings at feature level in the paper: one possible improvement could be fusions at model level, where we will integrate the classification results from different embeddings. Secondly, we would like to make a deeper analysis of Chinese radicals. In this paper, we treat each radical in a character with equal importance which is not ideal. As radicals in a same character have different functions, which results to different contributions to sentiment, a weighted radical analysis within each character is expected to further improve performance.

## References


Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.

Cambria, E.; Poria, S.; Bajpai, R.; and Schuller, B. 2016. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *COLING*, 2666–2677.

Cambria, E. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems* 31(2):102–107.

Chen, X.; Xu, L.; Liu, Z.; Sun, M.; and Luan, H. 2015. Joint learning of character and word embeddings. In *IJCAI*, 1236–1242.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 160–167. ACM.

Dong, Z., and Dong, Q. 2006. *HowNet and the Computation of Meaning*. World Scientific.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Ku, L.-W.; Liang, Y.-T.; and Chen, H.-H. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI spring symposium: Computational approaches to analyzing weblogs*.

Li, C.; Xu, B.; Wu, G.; He, S.; Tian, G.; and Hao, H. 2014. Recursive deep learning for sentiment analysis over social data. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 180–185. IEEE Computer Society.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, 3.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Multi-granularity chinese word embedding. In *NIPS*, 3111–3119.

Mnih, A., and Hinton, G. 2007. Three new graphical models for statistical language modelling. In *ICML*, 641–648.

Mnih, A., and Hinton, G. E. 2009. A scalable hierarchical distributed language model. In *NIPS*, 1081–1088.

Mnih, A., and Kavukcuoglu, K. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*, 2265–2273.

Pedregosa, F.; Varoquaux, G.; and et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Peng, H.; Cambria, E.; and Hussain, A. 2017. A review of sentiment analysis research in chinese language. *Cognitive Computation*.

Poria, S.; Chaturvedi, I.; Cambria, E.; and Bisio, F. 2016. Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In *IJCNN*, 4465–4473.

Poria, S.; Cambria, E.; Bajpai, R.; and Hussain, A. 2017. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion* 37:98–125.

Rong, X. 2014. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Shi, X.; Zhai, J.; Yang, X.; Xie, Z.; and Liu, C. 2015. Radical embedding: Delving deeper to chinese radicals. In *ACL*, 594–598.

Sun, Y.; Lin, L.; Yang, N.; Ji, Z.; and Wang, X. 2014. Radical-enhanced chinese character embedding. In *LNCS*, volume 8835, 279–286.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*, 384–394.

Yin, R.; Wang, Q.; Li, R.; Li, P.; and Wang, B. 2016. Distributed representations of words and phrases and their compositionality. In *EMNLP*, 981–986.