Joint Learning Embeddings for Chinese Words and their Components via Ladder Structured Networks

Yan Song, Shuming Shi, Jing Li
Tencent AI Lab
{clksong,shumingshi,ameliajli}@tencent.com

Abstract

The components, such as characters and radicals, of a Chinese word are important sources to help in capturing semantic information of the word. In this paper, we propose a novel framework, namely, ladder structured networks (LSN), which contains three layers representing word, character and radical, and learns their embeddings synchronously. LSN captures not only the relations among words, but also the relations among their component characters and radicals, as well as the relations across layers. Each layer in LSN is pluggable so that any particular type of unit (word, character, radical) can be removed and the LSN is thus adjusted for particular types of inputs. In evaluating our framework, we use word similarity as the intrinsic evaluation and part-of-speech tagging and document classification as extrinsic evaluations. Experimental results confirm the validity of our approach and show superiority of our approach over previous work.

1 Introduction

Embeddings have been proven to be useful in natural language processing (NLP) with the rising of deep learning [Collobert *et al.*, 2011; Mikolov *et al.*, 2013b; Pennington *et al.*, 2014]. Of all levels of granularities that have been investigated for embedding learning [Mikolov *et al.*, 2013b; Kiros *et al.*, 2015; Le and Mikolov, 2014], word embeddings have received the widest attention mainly for the reason that words are conventionally considered the smallest element that can be uttered in isolation with semantic or pragmatic content,[1] especially for western languages such as English.

However, not all languages follow the practice of English. Words in some languages are assembled by smaller units that can be used separately to deliver semantic information. In Chinese, words are made of characters, where each character carries rich semantic knowledge so that the meaning of a Chinese word is highly related to the characters it is comprised of. For example, 汽车 ("automobile") and 火车 ("train") are types of 车 ("vehicle"); the meanings of the two words

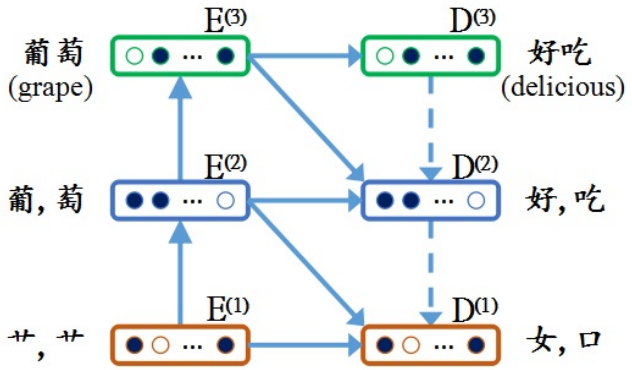

Figure 1: Illustration of the ladder structured networks for joint learning embeddings of words, characters and radicals.

are largely determined by the character they share. Moreover, characters in Chinese can be further decomposed into radicals, and the meanings of the characters are also highly dependent on their radicals. Especially majority number of Chinese characters are phono-semantic compounds,[2] whose radicals in general are the semantic part in the character. For example, characters 江 (river), 湖 (lake) and 海 (sea) all have on the left a radical 氵 meaning water, indicating that these characters have a semantic connection with water.

With such internal structure, learning Chinese word embeddings can be enhanced accordingly. Previous work has proven the validity of leveraging characters for learning word embeddings [Chen *et al.*, 2015; Xu *et al.*, 2016], and leveraging radicals for learning character embeddings [Li *et al.*, 2015] and word embeddings [Yin *et al.*, 2016; Su and Lee, 2017]. Thus, it is straightforward to consider learning word, character and radical in a full decomposition chain and learn their embeddings synchronously. In this paper, we propose a ladder structured networks (LSN) to do so based on the relations among words and their components. The LSN is inspired by learning with lateral connections among layers in ladder networks [Valpola, 2014],

[1] Controversially in linguistics, morpheme is the smallest unit of meaning, which however does not stand on its own in running texts.

[2] Xu Shen in *Shuowen Jiezi* (Explaining Graphs and Analyzing Characters) (100 AD) placed approximately 82% of characters into this type, while in the *Kangxi Dictionary* (1716 AD) the number is close to 90%, owing to the extremely productive use of this technique to extend the Chinese vocabulary.

which are designed to learn better intermediate representations with connections across layers [Rasmus *et al.*, 2015a; 2015b]. The ladder networks' structure has been proven to be effective in learning with better generalization ability, especially on image processing [Ronneberger *et al.*, 2015; Jégou *et al.*, 2016], for the reason that its design enables the model to encode information from different levels.

Different from conventional ladder networks, the intermediate layers in our model do not follow a typical auto-encoder with automatic generated latent representations, but are supervised by words and their components. Note that layers[3] are the basic modules in LSN. As a result, LSN is modular in structure; every "stair-step" (layer) of the LSN "ladder" is pluggable and can be easily removed. The model is thus easy to be adapted to the scenarios with only learning with word and character, word and radical, character and radical, etc. Based on the layers, our model has multiple objective functions with respect to units of different granularities, e.g., words, characters and radicals. Therefore, using our proposed framework, one is able to incorporate more contextual information in a hierarchical manner in learning Chinese word embeddings when compared to previous studies [Chen *et al.*, 2015; Li *et al.*, 2015; Xu *et al.*, 2016]. Experimental results suggest that our approach outperforms all baselines in both intrinsic and extrinsic evaluations, especially when the training data is limited.

2 Ladder Structured Networks

As illustrated in Figure 1, in our model, three layers from bottom to top represent radicals ($\mathrm{E}^{(1)}$, $\mathrm{D}^{(1)}$), characters ($\mathrm{E}^{(2)}$, $\mathrm{D}^{(2)}$) and words ($\mathrm{E}^{(3)}$, $\mathrm{D}^{(3)}$), respectively. Horizontally, the model is divided into two parts across layers. Following the terminology in ladder networks, we use "encoder path" and "decoder path" as in Rasmus *et al.* [2015b] to describe the left and right part of our model. In the encoder path, predictions (the solid upward arrow on the left side in Figure 1) are conducted from radicals ($\mathrm{E}^{(1)}$) to characters ($\mathrm{E}^{(2)}$), and from characters to words ($\mathrm{E}^{(3)}$). This path ensembles the process of how a word is composed, which complies with our intuition: when one has a word in his mind, he composes the word starting from radicals, and then characters. Formally, for each word w, predictions from $\mathrm{E}^{(1)}$ to $\mathrm{E}^{(2)}$ and from $\mathrm{E}^{(2)}$ to $\mathrm{E}^{(3)}$ is to maximize the likelihoods similar to the CBOW model [Mikolov *et al.*, 2013a], i.e.,

$$\mathcal{L}_{E^{(1)}E^{(2)}} = \sum_{c \in w} \log p(c \mid \sum_{r \in w} \upsilon_r) \tag{1}$$

$$\mathcal{L}_{E^{(2)}E^{(3)}} = \log p(w \mid \sum_{c \in w} \upsilon_c) \tag{2}$$

respectively. For each word w, c are the characters that compose this word and r are the radicals of these characters. υ_c and υ_r refer to the embeddings of these characters and radicals. Since a radical is the main component of a character, each word has equal number of characters and radicals. We use $r \in w$ to represent the relationship of a word and its radicals. In the decoder path, we do not have a prediction chain as that in the encoder path. The reason is that when a word ($\mathrm{D}^{(3)}$) is given, its component characters ($\mathrm{D}^{(2)}$) and radicals ($\mathrm{D}^{(1)}$) are constantly determined (the dotted arrows on the right side in Figure 1). Thus predictions following this path do not contribute to learning the model.

More importantly, there are predictions connecting different components from the two paths. On the same layers, we have word-word ($\mathrm{E}^{(3)} \rightarrow \mathrm{D}^{(3)}$), character-character ($\mathrm{E}^{(2)} \rightarrow \mathrm{D}^{(2)}$) and radical-radical ($\mathrm{E}^{(1)} \rightarrow \mathrm{D}^{(1)}$) predictions. The word-word prediction is essentially an SG model [Mikolov *et al.*, 2013b], which maximizes the following likelihood

$$\mathcal{L}_{E^{(3)}D^{(3)}} = \sum_{0<|i|\leq \mathcal{C}} \log p(w_i \mid \upsilon_w) \tag{3}$$

where w_i are words from w's context $w_{-\mathcal{C}}^{+\mathcal{C}}$, and υ_w is the embedding of w. This is the core prediction of the model because words are the direct input of the model. Characters and radicals are derived from the input words observed from the running texts. Similarly, the likelihoods of character-character and radical-radical predictions are formulated as

$$\mathcal{L}_{E^{(2)}D^{(2)}} = \sum_{0<|i|\leq \mathcal{C}} \sum_{c_i \in w_i} \log p(c_i \mid \sum_{c \in w} \upsilon_c) \tag{4}$$

$$\mathcal{L}_{E^{(1)}D^{(1)}} = \sum_{0<|i|\leq \mathcal{C}} \sum_{r_i \in w_i} \log p(r_i \mid \sum_{r \in w} \upsilon_r) \tag{5}$$

For cross layer scenarios, two more prediction lines are drawn from a word ($\mathrm{E}^{(3)}$) to characters ($\mathrm{D}^{(2)}$) and from characters ($\mathrm{E}^{(2)}$) to radicals ($\mathrm{D}^{(1)}$). They functionalize as auxiliary tasks to enhance representations at every layer of the model by introducing extra contextual information for characters and radicals from different granularities, which are proven to be useful in previous studies [Li *et al.*, 2015; Xu *et al.*, 2016]. These two prediction lines are formulated as

$$\mathcal{L}_{E^{(3)}D^{(2)}} = \sum_{0<|i|\leq \mathcal{C}} \sum_{c_i \in w_i} \log p(c_i \mid \upsilon_w) \tag{6}$$

$$\mathcal{L}_{E^{(2)}D^{(1)}} = \sum_{0<|i|\leq \mathcal{C}} \sum_{r_i \in w_i} \log p(r_i \mid \sum_{c \in w} \upsilon_c) \tag{7}$$

Given a corpus with vocabulary V and N tokens, our LSN is thus to maximize

$$\mathcal{L}_V = \frac{1}{N} \sum_{\substack{i=1 \\ w_i \in V}}^{N} \mathcal{L}_{LSN} \tag{8}$$

over the entire corpus, in which

$$\begin{aligned} \mathcal{L}_{LSN} = {} & \mathcal{L}_{E^{(1)}E^{(2)}} + \mathcal{L}_{E^{(2)}E^{(3)}} + \mathcal{L}_{E^{(1)}D^{(1)}} \\ & + \mathcal{L}_{E^{(2)}D^{(2)}} + \mathcal{L}_{E^{(3)}D^{(3)}} \\ & + \mathcal{L}_{E^{(3)}D^{(2)}} + \mathcal{L}_{E^{(2)}D^{(1)}} \end{aligned} \tag{9}$$

Similar to the SG model, at each training step, LSN first obtains a word pair from a training instance to feed the $\mathrm{E}^{(3)}$ and $\mathrm{D}^{(3)}$ position, then decomposes their components for other positions. Once all layers are assembled, LSN jointly

[3] Layers herein correspond to different decomposition levels, i.e., radical (R), character (C) and word (W).

Example	Nearest Words	Nearest Characters	Nearest Radicals
word: 航线 (airline)	航班 (flight), 包机 (charter flight), 航点 (waypoint)	航 (navigate), 国 (country), 运 (transport)	舟 (boat), 氵 (water), 斤 (axe),
character: 航 (navigate)	航线 (airline), 航空 (aviation), 航运 (shipping)	舱 (cabin), 艘 (ship), 船 (boat)	舟 (boat), 飞 (fly), 氵 (water)
radical: 舟 (boat)	高速 (high speed), 官兵 (officers and men), 战争 (war)	艘 (ship), 舰 (ship), 舷 (boat)	走 (to walk), 尤 (particularly), 斤 (axe)

Table 1: Top 3 nearest words, characters and radicals of given examples according to cosine similarities of their embeddings.

国际 (international)	新 (new)
航空 (aviation)	服务 (service)
年 (year)	机场 (airport)
日 (day)	渡轮 (ferry)
开通 (launch)	香港 (Hong Kong)

Table 2: Top 10 context words of 航线 extracted from Wiki data.

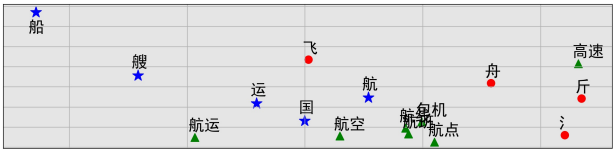

Figure 2: Partial t-SNE visualization of example words (green triangles), characters (blue stars) and radicals (red dots) in Table 1.

trains embeddings for them according to Eq 1 to 7. As a result, learning embeddings for words, characters and radicals in each training instance is directly computed by stochastic gradient descent (SGD) following a dimension-wise updating from the partial derivatives $\frac{\partial \mathcal{L}_{LSN}}{\partial v_w}$, $\frac{\partial \mathcal{L}_{LSN}}{\partial v_c}$ and $\frac{\partial \mathcal{L}_{LSN}}{\partial v_r}$:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{LSN}}{\partial v_w} &= \frac{\partial \mathcal{L}_{E^{(3)}D^{(3)}}}{\partial v_w} + \frac{\partial \mathcal{L}_{E^{(3)}D^{(2)}}}{\partial v_w} \\
\frac{\partial \mathcal{L}_{LSN}}{\partial v_c} &= \frac{\partial \mathcal{L}_{E^{(2)}E^{(3)}}}{\partial v_c} + \frac{\partial \mathcal{L}_{E^{(2)}D^{(2)}}}{\partial v_c} + \frac{\partial \mathcal{L}_{E^{(2)}D^{(1)}}}{\partial v_c} \\
\frac{\partial \mathcal{L}_{LSN}}{\partial v_r} &= \frac{\partial \mathcal{L}_{E^{(1)}E^{(2)}}}{\partial v_r} + \frac{\partial \mathcal{L}_{E^{(1)}D^{(1)}}}{\partial v_r}
\end{aligned} \tag{10}$$

Therefore, for example, when word embeddings are updated, character and radical embeddings will be affected synchronously until the model is converged.

With LSN, words, characters and radicals are learned in linearly correlated vector spaces since the embedding updating processes are intertwined across different granularities. As illustrated in Figure 1, 葡萄 (grape) is associated with not only the word 好吃 (delicious), but also its characters 好 (good), 吃 (eat) and its radicals 女 (woman) and 口 (mouth). This spatial characteristic of embeddings facilitates leveraging word components as useful knowledge source to provide different levels of semantics. For example, 吃 and 口 are closely related to 葡萄 in semantics since "grape" can be "eaten" by "mouth". Therefore, with only one pair of words, LSN could capture more semantics than other models with such decomposition. The learned embeddings of these character and radical components can further influence the embedding learning for other words shared with the same characters and radicals. In this way, LSN automatically learns the roles of the characters and radicals in different words. As a comparison, to distinguish the internal characters in a word, similarity between a word and its component characters has to be explicitly calculated, such as in SCWE [Xu *et al.*, 2016].

Another advantage of using the hierarchical design is that the layers in the full stacked LSN (W+C+R) are pluggable so that it is easy to restructure the framework by taking off any particular layer and assemble the rest parts, e.g., LSN (W+C) or LSN (C+R), which is trained only with words and characters, or characters and radicals, respectively. This modular characteristic ensures our model being detachable and thus it can be used in a more flexible way for different scenarios.

3 Experiments

3.1 Experiment Settings

We use two corpora to train our embeddings. The first one is the manually word segmented corpus composed of People Daily of January 1998 (PD98), with 1M words. Its vocabulary includes 56K words, 4.7K characters and 253 radicals. The second one is the Chinese Wikipedia dump (Wiki)[4]. We follow the procedures that done in Xu *et al.* [2016] for preprocessing, by removing pure digits and non-Chinese characters and use ANSJ[5] to segment words. The resulted dataset has 145M words in total. Its vocabulary includes 3M words, 18K characters and 275 radicals. For radicals, we use *online Xinhua Dictionary*[6] to extract character-radical mappings.

In our experiments, CBOW and SG, character-enhanced word embedding model (CWE) [Chen *et al.*, 2015][7], similarity based CWE (SCWE) and its extension multiple-prototype SCWE (SCWE+M) [Xu *et al.*, 2016][8], character CBOW (charCBOW) and character SG (charSG) [Li *et al.*, 2015], multi-granularity embedding (MGE) [Yin *et al.*, 2016][9] are used as our baselines in different tasks. For all the models in comparison, we set the dimension of embedding vectors to 200, the size of window to 5, the frequency cut-off to 5, the initialized learning rate to 0.025. The numbers of window-size and frequency cut-off are conducted on the word level. In our model, we use hierarchical softmax for the predictions.

[4]http://download.wikipedia.com/zhwiki/
[5]https://github.com/NLPchina/ansj_seg
[6]http://xh.5156edu.com/
[7]https://github.com/Leonard-Xu/CWE
[8]https://github.com/JianXu123/SCWE
[9]We reimplement charCBOW, charSG and MGE with our own codes according to their papers.

	WS-240	WS-296
CBOW	19.22	10.94
SG	19.58	11.73
CWE	19.29	10.71
SCWE	18.88	11.03
SCWE+M	20.23	10.82
MGE	18.55	9.75
LSN (W+R)	17.79	10.30
LSN (W+C)	29.38	13.90
LSN (W+C+R)	**34.23**	**18.23**

Table 3: Word similarity results ($\rho \times 100$) on WordSim-240 and WordSim-296 with the embeddings trained from the PD98 dataset.

	WS-240	WS-296
CBOW	51.25	53.82
SG	51.91	54.05
CWE	51.75	53.64
SCWE	52.11	54.20
SCWE+M	52.85	55.26
MGE	53.13	53.33
LSN (W+R)	52.01	53.44
LSN (W+C)	53.47	55.58
LSN (W+C+R)	**54.14**	**57.04**

Table 4: Word similarity results ($\rho \times 100$) on WordSim-240 and WordSim-296 with the embeddings trained from the Wiki dataset.

3.2 Qualitative Analysis

Since our approach is able to learn the embeddings of Chinese words, characters and radicals in linearly correlated vector spaces, it is straightforward to compute their relations through their embeddings. Herein we select word 航线 (airline) and its component character 航 (navigate) and the component radical 舟 (boat) to investigate their top 3 nearest words, characters and radicals in Table 1. The embeddings for this analysis are from the LSN (W+C+R) model trained on the Wiki data.

For word 航线, its component character 航 and radical 舟 are all included in the nearest characters and radicals, respectively. Especially, 氵 and 斤 are also included in its nearest radicals. Consider that 斤 (axe) is the radical of 新 (new), it makes sense that "airline" is intensively associated with "new" in the corpus. Interestingly, besides 舟, the other radical 纟 (silk) of 航线 is not found in its nearest radicals, which is a good example demonstrates the effectiveness of LSN in distinguishing radical roles. As a result, 纟 is semantically closer to other words (characters), such as 纺织 (textile).

To better understand the results, Table 2 lists the top ten frequent context words of 航线 in the Wiki data.[10] For a given word, although its nearest words (or characters, radicals) computed in the vector space are not necessarily from its context words, it is still a good reference of the high frequent context words to explain the observations we have in Table 1, e.g., all the nearest radicals in Table 1 can be found in the words in Table 2. Particularly, for radical 舟, its nearest words are more likely to be the context words with characters containing this radical, which is straightforward since it is in multiple characters. Considering that in our model there is no direct link between radical and word layer, the most related words to radical 舟 are the ones related to all the characters sharing this radical. For its most related radicals, 走 (to walk) has the highest similarity is mainly because multiple characters sharing this radical, such as 赶 (swiftly), 起 (rise), 超 (surpass), etc., which have close semantic relations to the characters containing the radical 舟. The radical 尤 (particularly) appeared in the top 3 nearest radicals is because it is the component of the character 尤 (particularly), which is one of the characters in a common adverb 尤其 (particularly) in Chinese. The visualization of the aforementioned examples on a 2D plots using t-SNE [van der Maaten and Hinton, 2008] further confirms the observations in above texts.

3.3 Word Similarity

The intrinsic evaluation is word similarity assessment. Normally the correlation between the scores generated by a model and human judgment indicates how good the model performs. We use WS-240 and WS-296 [Jin and Wu, 2012] in this paper as our evaluation datasets. The Spearman's rank correlation (ρ) is adopted for calculating the correlation. We evaluate our models trained on the PD98 and Wiki datasets. The results are reported in Table 3 and 4, respectively.

We have the following observations. First, LSN model is effective to learn meaningful embeddings. When trained on either PD98 or Wiki dataset, LSN models yield better or close ρ scores to state-of-the-art models. Second, jointly modeling embeddings on the levels of word, character, and radical is important. By exploring the joint effects of radicals, characters, and words in learning embeddings, LSN (W+C+R) produces better ρ scores than their counterparts, i.e., LSN (W+C) and LSN (W+R), without modeling radicals or characters.

3.4 Part-of-Speech Tagging

It is by our intuition to hypothesize that the semantic attributes of the internal components of Chinese words can help to identify word syntactic types. For example, words with the radical 扌 (hand) are very likely to be verbs, such as 打扰 (interrupt) and 搅拌 (stir); In addition, radical 木 (wood) often contributes to nouns, e.g., 森林 (forest) and 桌椅 (desk and chair). In this task, we use part-of-speech (POS) tagging as an extrinsic evaluation to assess how different embeddings performed in solving syntax problems.

We implement the BiLSTM-CRF model [Huang *et al.*, 2015] as our POS tagger, fed by pretrained word embeddings from different approaches. We set the hidden layer size to 256, with 10 epochs[11] for training. Note that in spite of input embeddings, no extra features are used in our model. We use Penn Chinese Treebank v5.0 (CTB-5) [Xue *et al.*, 2005] as the evaluation data, under the standard split with training/test as 18086/348 sentences, respectively. Similar to word similarity task, we also test the emebeddings trained from PD98 and Wiki. The results are reported in Table 5.

[10] We do not include stop words such as 的 (of), 是 (is), 和 (and) etc., in the list.

[11] Normally the model convergence requires less than 10 epochs, which is also addressed in Huang *et al.* [2015].

	PD98	Wiki
CBOW	94.26	94.96
SG	94.52	95.17
CWE	94.36	95.10
SCWE	94.48	95.35
SCWE+M	94.46	95.23
MGE	94.66	95.48
LSN (W+R)	94.54	95.20
LSN (W+C)	94.87	95.45
LSN (W+C+R)	**94.97**	**95.58**

Table 5: POS tagging accuracies with different input embeddings trained from PD98 and Wiki datasets.

We observe that POS taggers using different embeddings produce various accuracies. This indicates that the quality of word embeddings is vital for POS tagging, which is consistent with the conclusion in Huang *et al.* [2015]. It is also observed that POS taggers with LSN (W+C+R) trained on either PD98 or Wiki yield the best accuracy. This shows that characters and radicals may provide critical word type information, which is useful in POS tagging.

To further investigate the performance of taggers with different embeddings on particular POS tags, Table 6 compares the tagging performance on the top six POS tags[12] using our LSN (W+C+R) and SG embeddings trained from Wiki dataset. A ten-partition two-tailed paired t-test at $p < 0.05$ level is conducted on LSN results against the SG ones. It is verified that tagging accuracies on noun (*NN*), verb (*VV*), preposition (*P*) and adverbs (*AD*) are improved with using LSN embeddings, where significant improvements are observed on verbs, preposition and adverbs. We investigate some adverbs and found their component characters provide useful guidance. For example, 平稳 (smooth and steady) is tagged with *NN* by baseline embeddings; our embeddings correct its tag to *AD*, for its component character and radical 平 (long) has an JJ/AD attribute that has a strong indication for the adverb tag. The improvement on prepositions is mainly contributed from other POS tags, especially nouns and verbs. In Table 6 we also notice a slightly performance degradation on proper nouns (*NR*). The reason is that proper nouns are usually person and organization names, which typically do not have a pattern of characters and radicals, thus cannot provide semantic guidance for word embeddings.

Above observations meet our expectation that words' components can help POS tagging, and our embeddings are proved to be the most effective one in leveraging such information for POS tagging when compared to other models.

	SG	LSN
NN (noun)	94.46	**95.95**
VV (verb)	88.07	**91.34***
PU (punctuation)	100.0	100.0
NR (proper noun)	95.57	95.19
P (preposition)	87.78	**91.06***
AD (adverb)	86.27	**91.94***

Table 6: Tagging accuracies on top five POS tags that improved the most with using LSN when comparing to SG model. * indicates t-test significance at $p < 0.05$ level.

3.5 Document Classification

The second extrinsic evaluation is document classification. We experiment on *Fudan Corpus*[13], which contains 9.8K documents in 20 categories. We follow Xu *et al.* [2016] by constructing two datasets from the *Fudan corpus*, namely, *Fudan-large* and *Fudan-small*, with each dataset containing 5 categories of documents. The *Fudan-large* dataset includes categories of Agriculture, Economy, Environment, Politics and Sports, while the *Fudan-small* includes Education, Medical, Military, Philosophy and Transport. For each category, we randomly select 80 percent of the documents as the training dataset and reserve the rest as the test dataset. As a result, the training and test sets contain 4,894 and 1,227 documents for *Fudan-large* dataset, 233 and 61 for *Fudan-small* dataset. Similar to the Wiki dataset, pure digits and non-Chinese characters are removed and word segmentation is conducted. The publish information for each document is also removed because it contains strong indication of the categories, which will bias the classifier with unfair benefits.

This document classification experiment is performed in a conventional way as that in previous studies [Kiela *et al.*, 2015; Kiros *et al.*, 2015]. For all the documents in training and test datasets, we first construct document level representations by averaging the embeddings from all words in a given document. A logistic regression classifier is then trained on top of the resulted document level representations on the training set and evaluated on the test set.

In addition to word embeddings, we also evaluate character embeddings in this experiment. In order to do so, each document is represented as average embeddings of the characters in the document instead of words. Note for SCWE+M, a character may have multiple embeddings for its different prototypes, we then take the average embedding for the character from its multiple prototypes. In addition to the baselines we have for previous experiments, we add charCBOW and charSG as extra baselines in this task.[14] We compare with our variation LSN (C+R) in document classification task, which jointly models embeddings for characters and radicals. Although in Li *et al.* [2015], the bi-character embeddings achieve slightly better performance in document classification, we do not include them because, as stated in Xu *et al.* [2016], bi-characters are meaningless and may not form a Chinese word, thus are not comparable with words in other approaches. For this task, The input word and character embeddings are trained from the Wiki dataset.

Table 7 and 8 report the classification accuracies evaluated on *Fudan-small* and *Fudan-large* datasets, respectively. In general, our approach outperforms baselines on both datasets. This task well demonstrates the advantage of learning Chi-

[12]Frequency above 300 in the test data.

[13]http://www.datatang.com/data/44139

[14]In previous experiments, we do not consider them in comparison because they are on character level and previous evaluation are done on word level.

	Word	Character
CBOW	85.25	88.52
SG	85.25	86.89
CWE	86.89	86.89
SCWE	88.52	88.52
SCWE+M	90.16	88.52
charCBOW	-	85.25
charSG	-	83.61
MGE	86.89	88.52
LSN (C+R)	-	86.89
LSN (W+R)	86.89	-
LSN (W+C)	91.80	90.16
LSN (W+C+R)	**93.44**	**91.80**

Table 7: Document classification results on *Fudan-small* dataset based on word and character embeddings from different methods. Numbers are classification accuracies.

	Word	Character
CBOW	91.69	90.63
SG	91.36	90.71
CWE	91.20	91.77
SCWE	91.93	91.85
SCWE+M	92.09	91.61
charCBOW	-	91.04
charSG	-	90.87
MGE	91.85	91.20
LSN (C+R)	-	91.61
LSN (W+R)	91.61	-
LSN (W+C)	92.75	92.18
LSN (W+C+R)	**93.07**	**92.50**

Table 8: Document classification results on *Fudan-large* dataset based on word and character embeddings from different methods. Numbers are classification accuracies.

nese embeddings with considering different levels of components. We also notice that LSN (C+R) model fails to yield good results, which might because of the restricted alphabet at character level. This reason may also explains the poor performance of charCBOW and charSG.

Similar to what previous experiments show, learning on multiple levels of word components is very important regardless whether training data is limited, especially for the case that many words are unseen in the test data. Word embeddings with considering characters and radicals could help in this scenario bridging the gap of the out-of-vocabulary words. For example, in our experiment on *Fudan-small* dataset, the word 航线 from test data never appears in the training data. However, with the shared character 航 (navigate) and radical 舟, the embeddings learned by LSN for 航线 have higher similarity than other embeddings to 航空, which is a in-vocabulary word in transport documents. Thus the document containing 航线 tend to have a closer distance to the transport category. This joint learning with word components ensures the stability of our approach in representing words' semantics. Therefore, it is observed from Table 7 and 8, our approach achieves comparable results across two datasets.

4 Related Work

Word representation models represent words as real valued vectors that convey semantic information via exploring word co-occurrence patterns in contexts of neighboring words [Bengio *et al.*, 2003; Collobert *et al.*, 2011; Mikolov *et al.*, 2013a; 2013b]. Despite of the success of `word2vec`, its continuous bag-of-words (CBOW) and skip-gram (SG) models are not capable of capturing morphological information underlying internal word structure, which has been proven to be useful in learning English word embeddings [Luong *et al.*, 2013; Botha and Blunsom, 2014; Trask *et al.*, 2015; Miyamoto and Cho, 2016].

In Chinese, radicals, i.e., the internal structure of Chinese characters, have shown their effectiveness in learning word-level or character-level embeddings [Chen *et al.*, 2015; Xu *et al.*, 2016; Yin *et al.*, 2016]. In exploiting such structure, Su and Lee [2017] proposed to enhance word embeddings with glyph features from character images. Nevertheless, these referred studies serve as extensions of CBOW model. Different from them, our model essentially employs the concept of SG model at word level with exploiting word co-occurrence patterns in local context. As indicated in Mikolov *et al.* [2013a; 2013b] as well as our experiments (see §3), SG outperforms CBOW in most NLP tasks when trained on small amount of training data and presents better on rare words. Our approach can be seen as started on a higher baseline than the previous work. Although Li *et al.* [2015] followed both CBOW and SG model for Chinese characters that combine radical information, they focus on exploiting the character-radical relations, where the joint effects of words, characters, and radicals are ignored. To the best of our knowledge, our approach is the first to explore different component levels of words with a hierarchical structure, which serves as a natural fit for learning embeddings of Chinese words and their components.

5 Conclusion

We proposed a novel framework, ladder structure networks (LSN), to jointly learn embeddings of Chinese words, characters and radicals. LSN represents word, character and radical as different layers in a hierarchical manner and learns their embeddings by maximizing an overall likelihood based on their relations. Experimental results from intrinsic and extrinsic evaluations confirmed the effectiveness of our approach.

LSN has several characteristics. First, the learning of LSN ensembles the formation process of Chinese words and captures their relations to other words and components. Second, words, characters and radicals are learned in linearly correlated vector spaces, thus it is easy to compute their relations through their embeddings. Third, with the character and radical information, LSN is able to learn word semantics without relying on external resource, especially when training data is very limited. This characteristic is highly important because word segmentation is vital and requires extra efforts for processing Chinese texts, especially in the cold-start scenario when entering a new domain. Overall, this work offers an

alternative way to learn better word embeddings when there is very few accurately (manually) segmented data. To further extend this work, different structures or prediction chains are worth exploring within current LSN framework, and the idea of LSN can be applied to other similar tasks and languages.

References

[Bengio *et al.*, 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, March 2003.

[Botha and Blunsom, 2014] Jan A. Botha and Phil Blunsom. Compositional Morphology for Word Representations and Language Modelling. *arXiv preprint*, abs/1405.4273, 2014.

[Chen *et al.*, 2015] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. Joint Learning of Character and Word Embeddings. In *Proceedings of IJCAI*, pages 1236–1242, 2015.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, Nov 2011.

[Huang *et al.*, 2015] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint*, abs/1508.01991, 2015.

[Jégou *et al.*, 2016] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *arXiv preprint*, abs/1611.09326, 2016.

[Jin and Wu, 2012] Peng Jin and Yunfang Wu. SemEval-2012 Task 4: Evaluating Chinese Word Similarity. In **SEM*, pages 374–377, Montréal, Canada, 7-8 June 2012.

[Kiela *et al.*, 2015] Douwe Kiela, Felix Hill, and Stephen Clark. Specializing Word Embeddings for Similarity or Relatedness. In *Proceedings of EMNLP*, pages 2044–2048, Lisbon, Portugal, September 2015.

[Kiros *et al.*, 2015] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought Vectors. In *NIPS*, pages 3294–3302, Cambridge, MA, USA, 2015.

[Le and Mikolov, 2014] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of ICML*, pages 1188–1196, Beijing, China, June 2014.

[Li *et al.*, 2015] Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. Component-Enhanced Chinese Character Embeddings. In *Proceedings of EMNLP*, pages 829–834, Lisbon, Portugal, September 2015.

[Luong *et al.*, 2013] Thang Luong, Richard Socher, and Christopher Manning. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of CoNLL*, pages 104–113, Sofia, Bulgaria, 2013.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint*, abs/1301.3781, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, pages 3111–3119. 2013.

[Miyamoto and Cho, 2016] Yasumasa Miyamoto and Kyunghyun Cho. Gated Word-Character Recurrent Language Model. *arXiv preprint*, abs/1606.01700, 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of EMNLP*, pages 1532–1543, Doha, Qatar, October 2014.

[Rasmus *et al.*, 2015a] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-Supervised Learning with Ladder Network. *arXiv preprint*, abs/1507.02672, 2015.

[Rasmus *et al.*, 2015b] Antti Rasmus, Harri Valpola, and Tapani Raiko. Lateral Connections in Denoising Autoencoders Support Supervised Learning. *arXiv preprint*, abs/1504.08215, 2015.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv preprint*, abs/1505.04597, 2015.

[Su and Lee, 2017] Tzu-ray Su and Hung-yi Lee. Learning Chinese Word Representations From Glyphs Of Characters. In *Proceedings of EMNLP*, pages 264–273, Copenhagen, Denmark, September 2017.

[Trask *et al.*, 2015] Andrew Trask, David Gilmore, and Matthew Russell. Modeling Order in Neural Word Embeddings at Scale. *arXiv pre-print*, abs/1506.02338, 2015.

[Valpola, 2014] Harri Valpola. From Neural PCA to Deep Unsupervised Learning. *arXiv pre-print*, abs/1411.7783, 2014.

[van der Maaten and Hinton, 2008] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[Xu *et al.*, 2016] Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. Improve Chinese Word Embeddings by Exploiting Internal Structure. In *Proceedings of NAACL-HLT*, pages 1041–1050, San Diego, California, June 2016.

[Xue *et al.*, 2005] Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2):207–238, 2005.

[Yin *et al.*, 2016] Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. Multi-Granularity Chinese Word Embedding. In *Proceedings of EMNLP*, pages 981–986, Austin, Texas, November 2016.