# Investigating Stroke-Level Information for Learning Chinese Word Embeddings

Shaosheng Cao[1,2], Wei Lu[2], Jun Zhou[1], and Xiaolong Li[1]

[1] AI Department, Ant Financial Services Group
[2] Singapore University of Technology and Design, Singapore
{shaosheng.css,jun.zhoujun,xl.li}@antfin.com
luwei@sutd.edu.sg

**Abstract.** We propose a novel method for learning Chinese word embeddings. Different from previous approaches, we investigate the effectiveness of the Chinese stroke-level information when learning Chinese word embeddings. Empirically, our model consistently outperforms several state-of-the-art methods, including skipgram, cbow, GloVe and CWE, on the standard word similarity and word analogy tasks.

**Keywords:** Chinese word embeddings, stroke $n$-grams, Wikipedia

## 1 Introduction

Wikipedia is one of the most important knowledge bases for many semantic web tasks [1, 3, 13]. With the extensive development of deep learning, word (or concept) embeddings are more and more widely used in entity linking [12], question-answering [11], knowledge representation [9], and so on. Therefore, learning better word embeddings on Wikipedia has profound significance to improve such tasks.

Most existing research efforts on learning word embeddings have conducted experiments on the English Wikipedia data, which were shown effective [6, 10]. However, with the availability of the large amount of Chinese Wikipedia pages, how to develop word embedding learning models that can specifically exploit the unique characteristics associated with the Chinese language becomes an important topic.

Existing works such as CWE [2] yield better results by incorporating Chinese characters as subword information. We further investigate finer grained structural information than character-level information for capturing Chinese word embeddings. Specifically, we investigate the usefulness of stroke-level information for such a task.

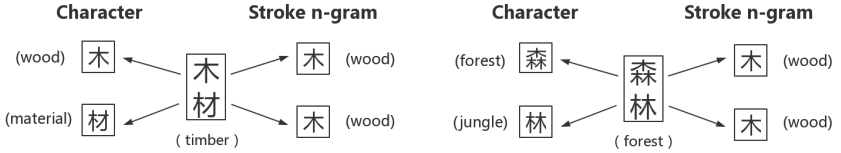


**Fig. 1.** Character v.s. stroke $n$-gram

As shown in Fig. 1, "木材 (timber)" and "森林 (forest)" are two semantically related words. There is no information that is shared across them at the character level. However, these two words are semantically related, and this can be captured by their stroke-level $n$-gram information that yields the common semantically meaningful component "木 (wood)".

# 2 Our Model

## 2.1 Stroke $n$-grams

In general, Chinese strokes can be summarized in five different types, i.e., horizontal stroke, vertical stroke, left-falling stroke, right falling stroke and turning stroke. We convert a word into stroke $n$-grams with four steps:

1. break down a word into characters;
2. obtain stroke sequence of each character by character-stroke mapping table;
3. concatenate such sequences of characters in a word;
4. use a sliding window with size $n$ to generate stroke $n$-grams.

## 2.2 Objective Function

word2vec [6, 7] aims to measure the semantic similarity between current word and its context words, where each word is treated as an atomic unit. In contrast, we split the current word $w$ into stroke $n$-gram collection $S(w)$ and retain its context words. As such, we define the following similarity function:

$$sim(w, c) = \sum_{q \in S(w)} \boldsymbol{q} \cdot \boldsymbol{c} \tag{1}$$

where $sim(w, c)$ is the similarity score between current word $w$ and a context word $c$, $q$ is an entry of collection $S(w)$, where $\boldsymbol{q}$ is the embedding of $q$. Inspired by NCE (noise-contrastive estimation) technology [4, 7, 8], we aim to optimize:

$$\mathcal{L} = \sum_{w \in D} \sum_{c \in T(w)} \Big( \log \sigma(sim(w, c)) + \lambda \mathbb{E}_{c' \sim P}[\log \sigma(-sim(w, c'))] \Big) \tag{2}$$

where $\mathcal{L}$ is the objective function, $D$ is the collection of words, and $T(w)$ is the collection of context words of $w$ within a sliding window. $\mathbb{E}_{c' \sim P}[\cdot]$ denotes the expectation, where $c'$ follows word unigram distribution $P$. In the euqation, $c'$ denotes a negative sample that does not occur in the context but randomly selected from the whole word vocabulary, and $\lambda$ represents the number of negative samples. In addition, $\sigma$ is sigmoid function described as $\sigma(x) = (1 + \exp(-x))^{-1}$.

# 3 Evaluation

## 3.1 Wikipedia Corpus Preprocessing

We download Chinese Wikipedia dump[3] as our training corpus, and preprocess the dataset as follows:

1. convert original `XML` format into plain text by a script[4] in `gensim` toolkit;
2. change traditional characters into simplified Chinese using `opencc`[5];
3. segment Chinese words in sentences by `ansj` toolkit[6].

We crawl the stroke sequences from `Xinhua` Dictionary[7] so as to build our stroke n-gram table.

[3] https://dumps.wikimedia.org/zhwiki/latest/
[4] https://radimrehurek.com/gensim/corpora/wikicorpus.html
[5] https://github.com/BYVoid/OpenCC
[6] https://github.com/NLPchina/ansj_seg
[7] http://xh.5156edu.com/

### 3.2 Benchmarks and Evaluation Metrics

Following [2], we evaluate the word embeddings on word similarity and word analogy tasks.

*Word similarity.* The task is a common approach to evaluate word embedding methods, which measures the semantic relatedness between two words. We conduct the experiments on two human-annotated datasets, including wordsim-240 and wordsim-296 [5], and Spearman's rank correlation coefficient $\rho$ [14] is used as the similarity metric.

*Word analogy.* The task is another important measure to validate the quality of word embeddings. It aims to infer a fourth word $t$ given three words $u$, $v$, $s$ that satisfies "$u$ is to $v$ what $s$ is to $t$". `3CosAdd` and `3CosMul` are two approaches that can be used to obtain the most suitable word $t$. We use the analogical data labeled from [2] and compare the accuracy percentages performed by different methods.

### 3.3 Baseline Algorithms

In order to validate the effectiveness of our proposed model, we compare the results to the following state-of-the-art algorithms as baselines.

- word2vec [6, 7] is a very popular model proposed by Google due to its high effectiveness and efficiency. It consists of two algorithms: `skipgram` and `cbow`.[8]
- GloVe [10] is a count-based model that leverages word-word co-occurrence statistics globally. It is reported that it performs better than word2vec on some tasks.[9]
- CWE [2] is recently introduced to specifically learn Chinese word embeddings which jointly learns representations for words and characters.[10]

In order to have a fair comparison, we set both window size and negative samples as 5 and fix the embedding size as 100 for all the models, and then remove the rare words that occur less than 10 times in the corpus.

### 3.4 Empirical Results

As shown in Table 1, CWE overall outperforms skipgram and cbow on word similarity and word analogy tasks, since it leverages certain subword information – character-level information. Interestingly, GloVe, as a count-based approach, is better than these three models on the word analogy task. We can observe that, however, our model achieves overall the best results, thanks to the stroke $n$-gram information incorporated.

## 4 Conclusion

We investigated the use of the stroke-level information when learning Chinese word embeddings. We demonstrated the effectiveness of such information. In our future work, we would like to investigate other application scenarios where stroke-level $n$-gram information could be used. For example, the task of joint learning of word embeddings and concept (or mention) embeddings for entity linking.

[8] code available: https://code.google.com/archive/p/word2vec/
[9] code available: http://nlp.stanford.edu/projects/glove/
[10] code available: https://github.com/Leonard-Xu/CWE

**Table 1.** Performance on word similarity and word analogy tasks. The evaluation metric is $\rho \times 100$ for word similarity and accuracy percentage for word analogy.

| Model | Word Similarity | | Word Analogy | |
|---|---|---|---|---|
| | wordsim-240 | wordsim-296 | 3CosAdd | 3CosMul |
| skipgram (mikolov et al., 2013) | 47.2 | 44.0 | 57.7 | 56.6 |
| cbow (mikolov et al., 2013) | 44.6 | 50.1 | 60.8 | 55.9 |
| GloVe (pennington et al., 2014) | 42.4 | 41.1 | 68.7 | 63.5 |
| CWE (chen et al., 2015) | 49.2 | 49.9 | 66.8 | 62.7 |
| Our Model (stroke $n$-gram) | **49.8** | **52.0** | **69.2** | **67.9** |

## Acknowledgement

The authors would like to thank the three anonymous reviewers for their valuable comments, as well as Ziqi Liu for his helpful suggestions on the work.

## References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. The semantic web (2007)
2. Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H.: Joint learning of character and word embeddings. In: IJCAI. pp. 1236–1242 (2015)
3. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI (2007)
4. Gutmann, M.U., Hyvärinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. JMLR 13(1), 307–361 (2012)
5. Jin, P., Wu, Y.: Semeval-2012 task 4: evaluating chinese word similarity. In: SemEval. pp. 374–377. ACL (2012)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
8. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: NIPS. pp. 2265–2273 (2013)
9. Ni, Y., Xu, Q.K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H.J., Cao, S.S.: Semantic documents relatedness using concept graph representation. In: WSDM. ACM (2016)
10. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. pp. 1532–1543 (2014)
11. Serban, I.V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., Bengio, Y.: Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. ACL (2016)
12. Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X.: Modeling mention, context and entity with neural networks for entity disambiguation. In: IJCAI. pp. 1333–1339 (2015)
13. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic wikipedia. In: WWW. ACM (2006)
14. Zar, J.H.: Significance testing of the spearman rank correlation coefficient. Journal of the American Statistical Association pp. 578–580 (1972)