cw2vec: Learning Chinese Word Embeddings with Stroke n-gram Information

Shaosheng Cao[1,2] and **Wei Lu**[2] and **Jun Zhou**[1] and **Xiaolong Li**[1]
[1] AI Department, Ant Financial Services Group
[2] Singapore University of Technology and Design
{shaosheng.css, jun.zhoujun, xl.li}@antfin.com
luwei@sutd.edu.sg

Abstract

We propose cw2vec, a novel method for learning Chinese word embeddings. It is based on our observation that exploiting stroke-level information is crucial for improving the learning of Chinese word embeddings. Specifically, we design a minimalist approach to exploit such features, by using stroke n-grams, which capture semantic and morphological level information of Chinese words. Through qualitative analysis, we demonstrate that our model is able to extract semantic information that cannot be captured by existing methods. Empirical results on the word similarity, word analogy, text classification and named entity recognition tasks show that the proposed approach consistently outperforms state-of-the-art approaches such as word-based word2vec and GloVe, character-based CWE, component-based JWE and pixel-based GWE.

1. INTRODUCTION

Word representation learning has recently received a significant amount of attention in the field of natural language processing (NLP). Unlike traditional one-hot representations for words, low-dimensional distributed word representations, also known as word embeddings, are able to better capture semantics of natural language words. Such representations were shown useful in certain down-stream NLP tasks such as text classification (Conneau et al. 2016; Xu et al. 2016), named entity recognition (Turian, Ratinov, and Bengio 2010; Collobert et al. 2011; Sun et al. 2015), machine translation (Devlin et al. 2014; Meng et al. 2015; Jean et al. 2015) and measuring semantic textual similarity (Shen et al. 2014; Wieting et al. 2015). It is therefore vital to design methods for learning word representations that can capture word semantics well.

Existing approaches only concentrated on learning such representations based on contextual information (Mikolov et al. 2010; 2013b; Ling et al. 2015; Levy, Goldberg, and Ramat-Gan 2014; Pennington, Socher, and Manning 2014) where words are regarded as atomic tokens. Recently, researchers also have started looking into incorporating subword level information to better capture word semantics (Bian, Gao, and Liu 2014; Cotterell and Schütze 2015; Bojanowski et al. 2016; Cao and Lu 2017). While these approaches were shown effective, they largely focused on European languages such as English, Spanish and German that employ the Latin script in their writing system. Therefore the methods developed are not directly applicable to languages such as Chinese that employ a completely different writing system.

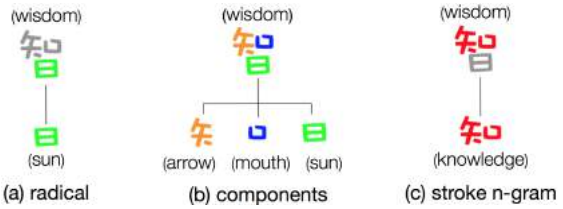

Figure 1: Radical v.s. components v.s. stroke n-gram

In Chinese, each word typically consists of less characters than English[1], where each character conveys fruitful semantic information (Wieger 1915; Liu et al. 2010). Given the rich internal structures of Chinese words and characters, approaches that exploit character level information (Chen et al. 2015) have been proposed for learning Chinese word embeddings. However, is such information sufficient for properly capturing the semantic information of words? Does there exist other useful information that can be extracted from words and characters to better model the semantics of words?

For internal structural information of words, we argue that characters alone are not sufficient for capturing the semantic information. For instance, two words "木材 (timber)" and "森林 (forest)" are semantically closely related. However, "木材 (timber)" is composed of two characters "木 (wood)" and "材 (material)", while "森林 (forest)" is made up of "森 (forest)" and "林 (jungle)". If only character level information is considered, there is no information that is shared across these two words as they consist of distinct characters.

While certain manually defined rules for extracting subword information such as radicals[2] (Sun et al. 2014; Li et al. 2015; Yin et al. 2016) and components can be exploited (Xin and Song 2017), such information might be incomplete

[1]Most modern Chinese words consist of only one or two characters (Chen, Liang, and Liu 2015).

[2]A component primarily used for indexing characters.

and noisy. As an illustrative example shown in Figure 1 (a), "日 (sun)" serves as the radical of "智 (intelligence)", but it hardly expresses any semantic information related to its character. Besides traditional radicals, components – the super collection of radicals, as summarized by HTTPCN[3], can be used to supplement the radicals. As shown in Figure 1 (b), "智 (intelligence)" is further reduced into the components "矢 (arrow)", "口 (mouth)" and "日 (sun)". However, all these components may not be relevant to the semantics of the character. Also, a pixel-based model that learns character features from font images is not shown to be better than original word2vec model as reported in (Su and Lee 2017). What constitutes the basic semantic unit that resides in a Chinese word and how to extract such information remain research questions to be answered.

While one can manually design methods to extract semantic features from Chinese characters, automatic feature learning could serve as an alternative approach (Bengio, Courville, and Vincent 2013). In this work, we develop an approach that can automatically acquire the meaningful latent representations associated with Chinese words, which requires minimal prior assumptions about how the words and characters are constructed. Specifically, we take a minimalist approach by exploiting stroke[4] n-gram information conveyed by Chinese words, which flexibly captures morphological and semantic information of the words. As shown in Figure 1(c), our proposed stroke n-gram "知 (knowledge)" is such a morphological structure of the character "智 (intelligence)", where more details will be given in Section 2.1.

We perform extensive experiments and through both qualitative and quantitative analysis, we show that our model is able to learn better Chinese word representations than other state-of-the-art approaches, including word-based word2vec (Mikolov et al. 2013a) and GloVe (Pennington, Socher, and Manning 2014), character-based CWE (Chen et al. 2015), component-based JWE (Xin and Song 2017) and pixel-based GWE (Su and Lee 2017). To the best of our knowledge, this is the first work that exploits stroke level information for learning Chinese word embeddings.

2. CW2VEC MODEL

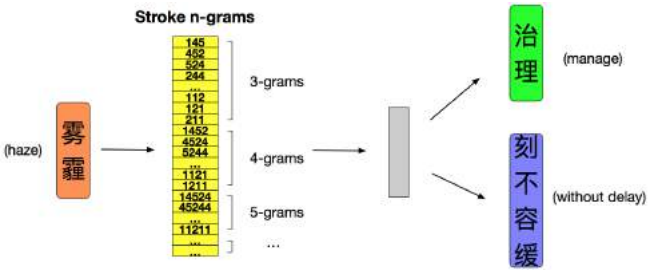

Figure 2: The overall architecture of our approach.

We provide a quick high-level overview of our cw2vec model. The overall architecture of the proposed model is shown in Figure 2 with an illustrative example. In this example, we have a short phrase "治理 雾霾 刻不油缓 (mange haze problem without delay)", where the current word is "雾霾 (haze)", and its context words are "治理 (manage)" and "刻不油缓 (without delay)".

We first reduce the current word into stroke n-grams as described in Section 2.1, where each stroke n-gram has a representational vector, *i.e.*, stroke n-gram embedding. Also, each context word is assumed to be associated with a word embedding of the same dimension. In our setting, the same word appearing at different positions in the corpus shares the same embedding, and so does the same stroke n-gram. After that, we optimize the carefully designed objective function and obtain final word embeddings and stroke n-gram embeddings based on the entire training corpus, as detailed in Section 2.2.

Stroke Name	Horizontal	Vertical	Left-falling	Right-falling	Turning
Shape, ID	一 (㇀), 1	丨(亅), 2	丿, 3	乀 (㇏), 4	乛(㇄㇇), 5

Figure 3: General shapes of Chinese strokes.

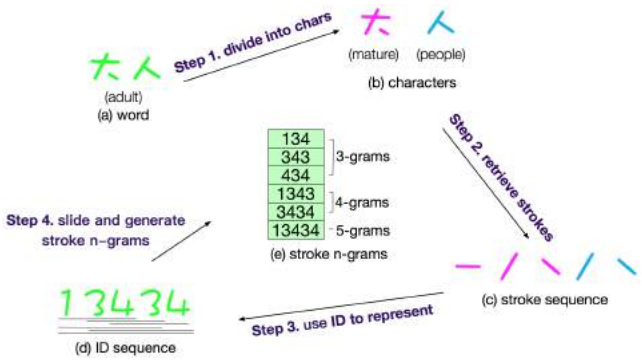

Figure 4: An illustrative example to show the procedures of the generation of stroke n-grams from a word.

2.1 Stroke n-grams

We classify the strokes into five different types in Figure 3. For convenience, we assign each stroke an integer ID, from 1 to 5 respectively.

The Chinese writing system provides some guidelines on what should be the natural order for strokes involved in each character[5]. With such stroke order information, one could design various models that can exploit such sequential information when modeling words. For example, recurrent neural networks (RNNs) such as LSTM (Hochreiter and Schmidhuber 1997) could be used here. However, as noted by (Mikolov et al. 2013a), such models typically involve an expensive training process. Inspired by the models built in the word2vec toolkit (2013a), we adopt a simpler and more efficient approach in this work by employing stroke n-gram information for characterizing Chinese words.

As described in Figure 4, we map a Chinese word into

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

[4]The basic patterns in the writing of Chinese characters: https://en.wikipedia.org/wiki/Stroke_(CJKV_character)

[5]https://en.wikipedia.org/wiki/Stroke_order

stroke n-grams using the following steps: (1) dividing the current word into characters, (2) retrieving the stroke sequence from each character and concatenating them together, (3) using stroke ID to represent stroke sequence, and (4) imposing a slide window of size n to generate stroke n-grams[6].

2.2 Objective Function

We consider measuring the similarity between a word and its context. In most models that regard words as atomic units, the similarity between current word w and one of its context words c is defined as $sim'(w, c) = \vec{w} \cdot \vec{c}$, where $\vec{w}$ and $\vec{c}$ are the embeddings of w and c respectively. In this work, we assign an embedding to each stroke n-gram and each context word, and define the similarity function between w and c based on the embeddings of stroke n-grams of the current word and the context word.

We scan every word throughout the corpus and obtain a stroke n-gram dictionary S, and we use $S(w)$ to denote the collection of stroke n-grams of the word w. We define the following similarity function between w and c:

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

where q is a stroke n-gram element of set $S(w)$, and the vector $\vec{q}$ is the embedding of q.

Similar to (Mikolov et al. 2013a; Bojanowski et al. 2016), we are interested in modeling the prediction of the context word c based on the current word w. One could use the softmax function to model the probability of predicting c given w:

$$p(c|w) = \frac{\exp(sim(w, c))}{\sum_{c' \in V} \exp(sim(w, c'))}$$

where c' is a word in the word vocabulary V. Direct calculation of the denominator can be very expensive, as it involves $|V|$ operations. To resolve the issue, we employ the *negative sampling* method, which is related to the noise-contrastive estimation proposed by (Gutmann and Hyvärinen 2012) and later extensively applied to word embedding learning tasks (Mnih and Kavukcuoglu 2013; Mikolov et al. 2013b), measuring semantic text similarity tasks (Huang et al. 2013; Shen et al. 2014), and node embedding learning tasks (Tang et al. 2015; Cao, Lu, and Xu 2015). The key idea of negative sampling is to replace the expensive denominator with a collection of context words "negatively" sampled based on a distribution. This gives rise to the following objective function defined throughout the entire corpus:

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

where λ is the number of negative samples, $\mathbb{E}_{c' \sim P}[\cdot]$ is the expectation term and a selected negative sample c' conforms to the distribution P, which could be set as the word unigram distribution U. In practice, to overcome the data sparseness issue, we raise U to the the 3/4rd power, following (Mikolov et al. 2013b). The activation function σ is the sigmoid function: $\sigma(x) = (1 + \exp(-x))^{-1}$. where $T(w)$ is the set of context words given current word within a window size, and D is the set of all words within the training corpus.

We optimize the above objective function using standard gradient based methods. After the learning process completes, we directly use the context word embeddings as our output word embeddings.

3. EXPERIMENTAL SETUP

In this section, we describe our data, benchmarks, evaluation methods, and baseline algorithms.

3.1 Data

We downloaded Chinese Wikipedia dump[7] on November 20, 2016, which consists of 265K Chinese Wikipedia articles. We use a script in the `gensim` toolkit to convert data from XML into text format[8]. Based on our observation, the corpus consists of both simplified and traditional Chinese characters. Hence we utilize the `opencc` toolkit[9] to normalize all characters as simplified Chinese. Following (Chen et al. 2015), all characters whose Unicode falls into the range between `0x4E00` and `0x9FA5` are Chinese characters and are retained. We use the `ansj` toolkit[10] for word segmentation.

We scan the training corpus and record all the characters so as to collect stroke n-grams. After that, we call the API of Chinese character information search service of the `Juhe Data`[11], which crawls stroke information from the online `Xinhua` Dictionary[12].

3.2 Benchmarks and Evaluation Metrics

Word Similarity Task Following (Chen et al. 2015; Xu et al. 2016), we adopt two manually-annotated datasets for Chinese word similarity task, *i.e.*, wordsim-240 and wordsim-296 (Jin and Wu 2012). These datasets are translated from English benchmarks and manually scored again, where words that have multiple senses or are difficult to translate are removed.

The word similarity task aims at the evaluation of the model's ability to capture semantic closeness and relatedness between two words. We compare the score calculated by the model against human-assigned scores using the Spearman's rank correlation coefficient ρ (Zar 1972) to assess the quality of word embeddings.

Word Analogy Task Another commonly used task for evaluating word embeddings is the word analogy task. This task examines the ability to deduce the semantic relations between different words with the learned word embeddings. In this task, three words a, b, and s are given, the goal is to infer a fourth word t that satisfies "a is to b that is similar to s is to t". The test dataset contains 1,124 testing instances.

[6]For all the experiments, we set the values of n from 3 to 12.

[7]https://dumps.wikimedia.org/zhwiki/20161120/

[8]https://radimrehurek.com/gensim/corpora/wikicorpus.html

[9]https://github.com/BYVoid/OpenCC

[10]https://github.com/NLPchina/ansj_seg

[11]https://www.juhe.cn/docs/api/id/156

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

Given the learned word embeddings, we use `3CosAdd` (Mikolov, Yih, and Zweig 2013) and `3CosMul` function (Levy, Goldberg, and Ramat-Gan 2014) to calculate the most appropriate word t. We employ both methods and adopt the same test data used in (Chen et al. 2015) for evaluations.

Text Classification Text classification is a common method to validate word embeddings on downstream tasks. We download *Fudan Corpus*[13], which contains 9,804 documents in 20 different topics. Similar to how we process the training data, non-Chinese characters are removed and `ansj` is used for performing Chinese word segmentation. Following (Xu et al. 2016), we select documents from 5 topics: 1,218 environment, 1,022 agriculture, 1,601 economy, 1,025 politics and 1,254 sports documents. Motivated by (Xu et al. 2016; Tang et al. 2015), we average the embeddings of the words occurred in a document as the feature representations of the document. We build classifiers with the `LIBLINEAR`[14] (Fan et al. 2008), where 70% of total data is used for training and the rest are used for evaluation. We report the accuracy score using different embeddings generated by different methods.

Named Entity Recognition We evaluate word embeddings via the Named Entity Recognition (NER) task as well. We implement the model described in (Ma and Hovy 2016) for performing NER, where only word embeddings are fed into the input layer as features. We used a publicly available dataset fully annotated with named entity labels[15], which contains 7 attributes for each Chinese word, including *time*, *location*, *person*, *organization*, *company*, *product* and *others*. The complete data consists of 1,999 documents, and 70% is randomly selected for training and the remaining 30% is used for evaluation. Note that for both text classification and NER tasks, we did not fine-tune the learned word embeddings.

Qualitative Evaluation We also conduct qualitative analysis of our results, where the top 10 words that are most similar to our target word are presented. The similar words are retrieved based on the cosine similarity calculated using the learned embeddings. Through the qualitative analysis, we can observe the different characteristics of each approach.

3.3 Baseline Algorithms

In order to assess the effectiveness of our model, we compared with several state-of-the-art algorithms listed below.

- word2vec[16] (Mikolov et al. 2013a; 2013b) is an effective and efficient toolkit for learning word embeddings, which implements two models, *i.e.*, `skip-gram` and `cbow`. Both of them will be regarded as our baselines.
- GloVe[17] (Pennington, Socher, and Manning 2014) is another state-of-the-art approach that collects global word-word co-occurrence information from the entire dataset. Both GloVe and word2vec are general word-based embedding learning models that are not specifically designed for any particular language.
- CWE[18] (Chen et al. 2015) is a character-based model aiming at learning Chinese word embeddings that exploits character level information by jointly learning character and word embeddings.
- GWE[19] (Su and Lee 2017) leverages pixel-level information, which exploits character features from font images by convolutional autoencoders.
- JWE[20] (Xin and Song 2017) reduces Chinese words into components of characters, as the superset of radicals. Components are extracted from the `HTTPCN` website, and component-based JWE was shown to be much more effectively than MGE (Yin et al. 2016) - another recently proposed purely radical-based Chinese word embedding model.

For a fair comparison between different algorithms, we use same dimension size for all word embeddings, and removed the rare words that appeared less than 10 times in the training corpus. The window size and negative samples were both set to 5.

4. EMPIRICAL RESULTS

In this section, we will show the empirical results of our `cw2vec` model compared to each baseline method.

4.1 Word Similarity and Word Analogy Results

As shown in Table 1, skip-gram performs better on word analogy than cbow, but worse on word similarity task. The results of CWE are generally better than skip-gram and cbow model on both word similarity and word analogy tasks. It is consistent with the findings reported in (Chen et al. 2015), since the character level information can improve Chinese word embeddings. We also observe that the embeddings produced by the GloVe model perform better on Chinese word analogy task than skip-gram and cbow. By leveraging a large number of Chinese components, JWE achieves good performance on wordsim-296 dataset and word analogy task[21]. Besides, GWE is not more effective than skip-gram and cbow model on our experiments in conformity with the conclusion reported in (Su and Lee 2017). Overall, thanks to the effectiveness of the proposed stoke n-grams, our model achieves the best results on wordsim-240 dataset. Our model also achieves the best results on wordsim-296,

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

[14] https://www.csie.ntu.edu.tw/c̃jlin/liblinear/

[15] http://bosonnlp.com/resources/BosonNLP_NER_6C.zip

[16] https://code.google.com/archive/p/word2vec/

[17] http://nlp.stanford.edu/projects/glove/

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

[19] https://github.com/ray1007/gwe

[20] https://github.com/hkust-knowcomp/jwe

[21] We were unable to obtain the original source code of MGE (Yin et al. 2016), but conducted experiments using the reproduced version provided by (Su and Lee 2017), whose results are consistently worse than JWE on each dataset in accordance with (Xin and Song 2017).

Model	Word Similarity		Word Analogy		Text Classification	Named Entity Recognition
	wordsim-240	wordsim-296	3CosAdd	3CosMul		
skip-gram (Mikolov et al. 2013b)	44.2	44.4	58.3	58.9	93.4	65.1
cbow (Mikolov et al. 2013b)	47.0	50.2	54.3	53.5	93.4	59.6
GloVe (Pennington, Socher, and Manning 2014)	45.2	44.3	68.8	66.7	94.2	66.0
CWE (Chen et al. 2015)	50.0	51.5	68.5	69.6	93.2	65.8
GWE (Su and Lee 2017)	50.0	49.1	50.8	50.6	94.3	65.5
JWE (Xin and Song 2017)	48.0	**52.7**	74.2	76.3	94.2	67.9
`cw2vec` (stroke n-grams)	**50.4**	**52.7**	**78.1**	**80.5**	**95.3**	**71.7**

Table 1: Performance on word similarity, word analogy task, text classification and named entity recognition. The embeddings are set as 300 dimensions. The evaluation metric is $\rho \times 100$ for word similarity, accuracy percentage for word analogy and text classification, $F1$-measure for named entity recognition task.

along with JWE. On the other hand, our model improves around 4 points over JWE on word analogy task.

Although morphological analysis for Chinese presents some unique challenges (Packard 2000), we believe that the rich stroke n-gram information used in our model does capture certain level of morphological information of Chinese. Words that share the same morphologically meaningful structure may tend to convey similar semantics – capturing such information can lead to improved modeling of word representations. Indeed from the results of the word similarity task we can see that our approach can be used to better capture the similarity between semantically related words, leading to improved word similarity results. Such stroke n-gram information, together with the contextual information exploited in the learning process, results in improved representations for Chinese words that also perform well on the word analogy task.

4.2 Text Classification and Name Entity Recognition Results

One of the goals for learning word embeddings is to use them in certain downstream NLP tasks. We first conduct experiments on the text classification task, in order to validate the effectiveness of the learned Chinese word embeddings. For each document, we construct a representation using word embeddings generated by different models. Thus high quality word embeddings will lead to good accuracy under the same classifier. As we can see from Table 1, skip-gram, cbow and CWE achieves over 93% accuracy, while the results of GloVe, GWE and JWE are above 94% on such a task. Overall, our model can reach above 95%, which gives a 1% absolute improvement in accuracy over the best baseline approach.

We next conduct an experiment on NER to observe the effectiveness of the embeddings on such a structured prediction task. The key of the task is to extract named entities and their associated types. $F1$ score of NER will be higher if better word embeddings are fed into the neural networks. As shown in Table 1, cbow does not perform well on this task, which might be due to the averaged context word embeddings, and skip-gram, GloVe, CWE and GWE are much better. JWE outstrips other baseline methods by around 2 points. Overall, our model outperforms JWE by 3.8 points, confirming the effectiveness of our stroke n-gram based approach.

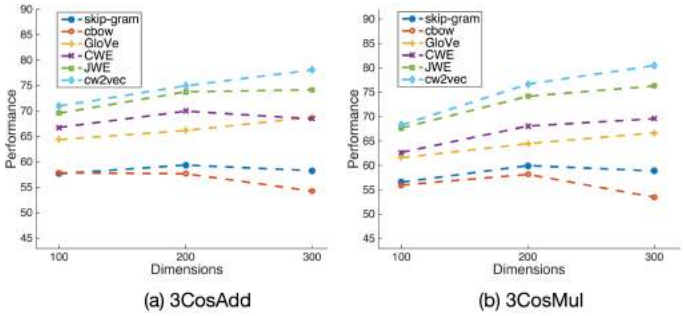

Figure 5: Performance on word analogy over dimensions

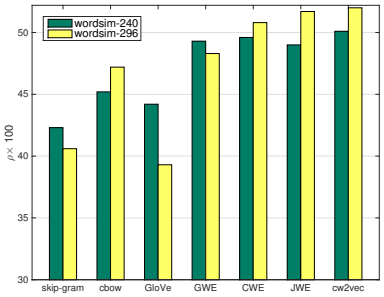

Figure 6: Performance on word similarity, trained on the front 20% wikipedia articles. The embeddings are set as 100 dimensions.

4.3 Performance v.s. Dimensions

In order to study the performance over different dimensions of embeddings, we conduct experiments on word analogy task for each model. As described in Figure 5, we can conclude that the results of skip-gram and cbow achieve the best scores when the dimension of the embeddings are set to 200. GloVe performs much better than these two methods over different dimensions, and CWE is better than GloVe especially when dimension is set to 200. With the component level information, JWE is able to outperform other baseline methods. Finally, our model achieves the best performance across different dimensions with finer grained subword structure - stroke n-grams.

4.4 Performance v.s Training Data Size

Besides the whole Chinese Wikipedia dump corpus as the data for training word embeddings, we also explore the results on a smaller dataset that includes only the first 20% Wikipedia articles for learning. The results are presented in Figure 6. With less training data, word-based skip-gram,

cbow and GloVe do not perform well, while the rest of models appear to be more robust than these models, which we believe is largely due to its ability to capture subword information. CWE outperforms other baseline methods on wordsim-240, while JWE performs better on wordsim-296 dataset. Our model, on the other hand, is able to obtain the best results on both datasets. We believe it is because our approach that exploits n-gram information is more robust to the data-sparseness issue during learning, in comparison to alternative approaches when there is limited training data available.

4.5 Qualitative Results

To better understand the quality of the learned word embeddings for each model, we conduct qualitative analysis by performing some case studies in Table 2 to illustrate the most similar words to certain target words under different methods. As shown in Table 2, the first example word we consider is "水污染 (water pollution)". It is an environment related term and is thus semantically related to words under such a theme.

GWE produces several words relevant to "污 (polluted)", such as "污日 (sludge)", "污渍 (stain)" and "污垢 (filth)", which are not directly semantically related to the target word. Compared with the GWE model, JWE appears to tend to emphasize on the last two characters "污染 (pollution)". It learns a representation for the target word "水污染 (water pollution)" that is similar to that of "油污 (oil pollution)", which seems not very semantically related. As a global count-based model, GloVe yields several words such as "循环系统 (circulatory system)" and "神经系统 (nervous system)", which do not appear to be semantically close to the target word. All the top-ranked words identified by CWE contain the character "水 (water)" or "污 (sewage)", which embodies its joint learning idea. Certain words such as "渗水 (leakage)" do not appear to be semantically very closely related to the target word. In contrast, the words identified by our model do not exhibit such an issue. They are largely semantically relevant to the target word and also share certain structural similarity with the target word. For example, the only model that can identify "水木 (water quality)" as a similar word is our model, yet this word is semantically highly relevant to the target word. Overall, our model can capture semantic relatedness well, since it appear to be able to extract additional semantic information due to the unique subword information it is able to capture.

As another example, we intentionally select the target word "孙悟智 (Sun Wukong)", the protagonist of both the popular Chinese classic novel known as "西游记 (Journey to the West)" and the famous Japanese cartoon known as "沙珠 (Dragon Ball)". GWE and CWE yields irrelevant words such as "小悟 (Xiao Wu)", "阿悟 (A Wu)", "甘悟 (Gan Wu)", and "玉悟 (Yu Wu)", which share the same character "悟 (Wu)". Again, GloVe generates several irrelevant words such as "色狼 (lady-killer)" and "阿日 (A Ge)". Conversely, all the similar words computed by our model are from the figure names (or alias) or production names in either of these two artworks.

The learning procedures for CWE and GWE involve learning separate embeddings for words and characters (and pixel information for GWE), combining them using simple operations such as averaging. One issue with such an approach is that the frequent characters may play a dominating factor in the learned embeddings to a certain extent from the above examples. In contrast, our model directly learns a single embedding for each word by constructing a similarity function between stroke n-grams and context words. Such an approach is shown to be able to alleviate the above-mentioned issue.

5. RELATED WORK

Most models for learning word embeddings are based on the idea of modeling the relationship between a word and its context. There are two main families of models for learning word embeddings – neural based and co-occurrence count-based models. The underlying principle based on which such models are proposed is the *distributional hypothesis* (Harris 1954): similar words tend to appear in similar contexts. Word2vec (Mikolov et al. 2013a) and GloVe (Pennington, Socher, and Manning 2014) are two representative models for learning word embeddings, which are widely used in the community due to their effectiveness and efficiency in learning.

5.1 Learning with Subword Information

Although the above-mentioned models are effective, they regard individual words as atomic tokens and the potentially useful internal structured information of words is ignored. This observation has led to the investigations on models that exploit subword information. Luong, Socher, and Manning (2013) proposed to use a recursive neural network to extract morphological information from words. Botha and Blunsom (2014) introduced a scalable model to integrate compositional morphology. Bian, Gao, and Liu (2014) explored different types of subword features to enhance the morphological property of word embeddings. Cotterell and Schütze (2015) also proposed models to integrate morphological information in the embedding learning process. Bojanowski et al. (2016) and Wieting et al.(2016) introduced models to learn word embeddings with character n-gram information. Cao and Lu (2017) presented a model based on convolutional neural networks that learns word embeddings with character trigrams, word affixes and inflections.

5.2 Chinese Word Embeddings

Models specifically designed for learning Chinese word embeddings have also been studied in the literature. Yang and Sun (2015) used a Chinese synonym dictionary known as *Cilin* (Jiaju et al. 1983) to disambiguate multiple senses of words or characters. Xu et al. (2016) utilized a translation tool to extract semantic knowledge from other languages for capturing the semantic information of characters in a word. While these approaches improve the quality of embeddings with external knowledge, there also exist approaches that exploit internal structure information associated with words. Chen et al. (2015) designed CWE model for jointly learning Chinese character and word embeddings, which makes full use of character level structured information for improving

Targets	GWE	JWE	GloVe	CWE	cw2vec
水污染 (water pollution)	污染源(pollutant src)	荒漠化(desertification)	公害(public nuisance)	污染源(pollutant src)	污染(pollution)
	污染(pollution)	污染孩(pollutant)	废弃孩(garbage)	污染(pollution)	污染孩(pollutant)
	水害(water damage)	内涝(waterlogging)	洪涝(flood)	污染孩(pollutant)	水木(water quality)
	污日(sludge)	排污(pollution discharge)	奶制品(dairy product)	水害(water damage)	水资源(water resource)
	沙漠化(desertization)	油污(oil pollution)	循环系统(circulatory sy)	污日(sludge)	污染源(pollutant src)
	污水(sewage)	沙漠化(desertization)	神经系统(nervous sy)	污水(sewage)	废水(waste water)
	污渍(stain)	地表水(surface water)	市油(city appearance)	污渍(stain)	荒漠化(desertification)
	废水(waste water)	盐碱化(salinization)	职业病(occupational ds)	污孩(dirt)	地下水(groundwater)
	渗水(leakage)	渗小(seepage)	结构性(designability)	废水(waste water)	地表水(surface water)
	污垢(filth)	公害(public nuisance)	污染(pollution)	渗水(leakage)	沙漠化(desertization)
孙悟智 (Sun Wukong)	孙悟天(Son Goten)	唐僧(Monk Tang)	唐僧(Monk Tang)	孙悟天(Son Goten)	沙悟净(Sha Wujing)
	孙悟饭(Son Gohan)	孙悟饭(Son Gohan)	孙悟饭(Son Gohan)	孙悟饭(Son Gohan)	白骨精(Bai Gujing)
	小悟(Xiao Wu)	白骨精(Bai Gujing)	白骨精(Bai Gujing)	小悟(Xiao Wu)	西游记(J. to the West)
	沙珠(Dragon Ball)	沙悟净(Sha Wujing)	西游记(J. to the West)	阿悟(A Wu)	沙僧(Monk Sha)
	甘悟(Gan Wu)	西游记(J. to the West)	沙珠(Dragon Ball)	沙悟净(Sha Wujing)	水王(Monkey King)
	阿悟(A Wu)	唐三藏(Xuanzang)	三打(three strikes)	甘悟(Gan Wu)	孙悟天(Son Goten)
	玉悟(Yu Wu)	贝三塔(Vegeta)	沙悟净(Sha Wujing)	董悟(Dong Wu)	唐三藏(Xuanzang)
	天大(extremely big)	红孩儿(Red Boy)	唐三藏(Xuanzang)	玉悟(Yu Wu)	贝三塔(Vegeta)
	真飞沙(really dragon)	水王(Monkey King)	色狼(lady-killer)	西游记(J. to the West)	沙珠(Dragon Ball)
	悟(Wu)	沙僧(Monk Sha)	阿日(A Ge)	沙珠(Dragon Ball)	孙悟饭(Son Gohan)

Table 2: Case study for qualitative analysis. Given the target word, we list the top 10 similar words from each algorithm so as to observe the differences. Due to limited space, we use the following abbreviations: *src* for *source*, *J.* for *Journey*, *sy* for *system*, and *ds* for *disease*.

the quality of Chinese word embeddings. Sun et al. (2014) and Li et al. (2015) used a radical dictionary to extract subword features during learning, and Xin and Song (2017) introduced a model called JWE that is based on components that is an extended radical collection. Liu et al. (2017) illustrated visual character embeddings from their images to reduce the data sparseness issue related to rare words, and GWE, proposed by Su and Lee (2017), directly extracts character features from the images using convolutional autoencoders .

6. CONCLUSION AND FUTURE WORK

In this work, we introduced `cw2vec`, a new model for learning Chinese word embeddings. From a linguistic point of view, unlike rule-based approaches that exploit rigid character or component level information, our approach provides a flexible way of capturing semantically meaningful subword information for Chinese. We validated our argument from an empirical point of view through both quantitative and qualitative analysis with comparisons against baseline approaches.

Generally speaking, stroke n-grams flexibly provide finer grained information associated with words, which may be overcomplete. On the other hand, similar to radical-based, component-based and pixel-based features, stroke n-grams might express ambiguous information under certain situations. For example, the two words "土 (soil)" and "士 (army person)" share exactly the same stroke n-grams though they are semantically unrelated. Nonetheless, our model outperforms other subword based models from experimental results. Although our model is introduced to learn Chinese word embeddings, the idea can also be applied to other languages that share a similar writing system. In the future, we would like to further explore learning representations for traditional Chinese words and Japanese Kanji, and we are also interested in investigating alternative neural architectures for capturing stroke information of Chinese words.

Acknowledgments

We would like to thank all the anonymous reviewers for their thoughtful and constructive comments, as well as Jiaqi Zhang and Rong An for their help on the NER toolkit. This work is partly supported by MOE Tier 1 grant SUTDT12015008.

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *TPAMI*.

Bian, J.; Gao, B.; and Liu, T.-Y. 2014. Knowledge-powered deep learning for word embedding. In *ECML-PKDD*.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Botha, J. A., and Blunsom, P. 2014. Compositional morphology for word representations and language modelling. In *ICML*.

Cao, S., and Lu, W. 2017. Improving word embeddings with convolutional feature learning and subword information. In *AAAI*.

Cao, S.; Lu, W.; and Xu, Q. 2015. Grarep: Learning graph representations with global structural information. In *CIKM*.

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

Chen, H.; Liang, J.; and Liu, H. 2015. How does word length evolve in written Chinese? *PloS one*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *JMLR*.

Conneau, A.; Schwenk, H.; Barrault, L.; and Lecun, Y. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*.

Cotterell, R., and Schütze, H. 2015. Morphological word-embeddings. In *NAACL*.

Devlin, J.; Zbib, R.; Huang, Z.; Lamar, T.; Schwartz, R. M.; and Makhoul, J. 2014. Fast and robust neural network joint models for statistical machine translation. In *ACL*.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *JMLR*.

Gutmann, M. U., and Hyvärinen, A. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *JMLR*.

Harris, Z. S. 1954. Distributional structure. *Word*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2015. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*.

Jiaju, M.; Yiming, Z.; Yunqi, G.; and Hong-Xiang, Y. 1983. Tongyici cilin. *Shanghai: Shanghai Lexicon Publishing Company*.

Jin, P., and Wu, Y. 2012. Semeval-2012 task 4: evaluating chinese word similarity. In *SemEval*.

Levy, O.; Goldberg, Y.; and Ramat-Gan, I. 2014. Linguistic regularities in sparse and explicit word representations. In *CoNLL*.

Li, Y.; Li, W.; Sun, F.; and Li, S. 2015. Component-enhanced chinese character embeddings. *arXiv preprint arXiv:1508.06669*.

Ling, W.; Dyer, C.; Black, A.; and Trancoso, I. 2015. Two/too simple adaptations of word2vec for syntax problems. In *NAACL*.

Liu, P. D.; Chung, K. K.; McBride-Chang, C.; and Tong, X. 2010. Holistic versus analytic processing: Evidence for a different approach to processing of chinese at the word and character levels in chinese children. *JECP*.

Liu, F.; Lu, H.; Lo, C.; and Neubig, G. 2017. Learning character-level compositionality with visual features. In *ACL*.

Luong, T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Meng, F.; Lu, Z.; Wang, M.; Li, H.; Jiang, W.; and Liu, Q. 2015. Encoding source language with convolutional neural network for machine translation. *arXiv preprint arXiv:1503.01838*.

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

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. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*.

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

Packard, J. L. 2000. *The morphology of Chinese: A linguistic and cognitive approach*. Cambridge University Press.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Shen, Y.; He, X.; Gao, J.; Deng, L.; and Mesnil, G. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*.

Su, T.-R., and Lee, H.-Y. 2017. Learning chinese word representations from glyphs of characters. *EMNLP*.

Sun, Y.; Lin, L.; Yang, N.; Ji, Z.; and Wang, X. 2014. Radical-enhanced chinese character embedding. In *ICONIP*. Springer.

Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; and Wang, X. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*.

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

Wieger, L. 1915. *Chinese characters: Their origin, etymology, history, classification and signification. A thorough study from Chinese documents*. Catholic mission press.

Wieting, J.; Bansal, M.; Gimpel, K.; and Livescu, K. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Wieting, J.; Bansal, M.; Gimpel, K.; and Livescu, K. 2016. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*.

Xin, J. Y. X. J. H., and Song, Y. 2017. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. *EMNLP*.

Xu, J.; Liu, J.; Zhang, L.; Li, Z.; and Chen, H. 2016. Improve chinese word embeddings by exploiting internal structure. In *NAACL-HLT*.

Yang, L., and Sun, M. 2015. Improved learning of chinese word embeddings with semantic knowledge. In *CCL & NLP-NABD*. Springer.

Yin, R.; Wang, Q.; Li, P.; Li, R.; and Wang, B. 2016. Multi-granularity chinese word embedding. *EMNLP*.

Zar, J. H. 1972. Significance testing of the spearman rank correlation coefficient. *JASA*.