Glyph2Vec: Learning Chinese Out-of-Vocabulary Word Embedding from Glyphs

Hong-You Chen∗
The Ohio State University
chen.9301@osu.edu

Sz-Han Yu∗
National Taiwan University
r04922007@ntu.edu.tw

Shou-De Lin
National Taiwan University
sdlin@csie.ntu.edu.tw

Abstract

Chinese NLP applications that rely on large text often contain huge amounts of vocabulary which are sparse in corpus. We show that characters' written form, *Glyphs*, in ideographic languages could carry rich semantics. We present a multi-modal model, *Glyph2Vec*, to tackle Chinese out-of-vocabulary word embedding problem. Glyph2Vec extracts visual features from word glyphs to expand current word embedding space for out-of-vocabulary word embedding, without the need of accessing any corpus, which is useful for improving Chinese NLP systems, especially for low-resource scenarios. Experiments across different applications show the significant effectiveness of our model.

1 Introduction

Word embedding encoded semantic and syntactic information (Mikolov et al., 2013a,b) in low-dimensional space have served as useful features for various NLP applications but often require large-scale corpus with billions of tokens to train.

A natural constraint of word embedding is that it is not practical to collect the entire vocabulary of any language with large enough frequency to train the embedding for every word, since some new words may appear in downstream tasks. A typical solution is to simply assign a specific *UNK* embedding to all out-of-vocabulary (OOV) words that do not appear in the training data.

Current solutions such as using subwords (e.g., characters) are mainly considering alphabetic languages (e.g., English and French) that are composed of small amount of characters. Such techniques may not be sufficient for ideographic lan-
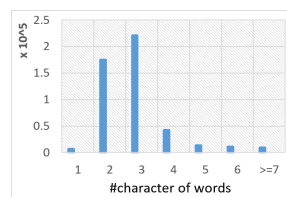

Figure 1: Statistics of length of Chinese words in Sinica Corpus.

guages (e.g., Chinese and Japanese) in which a word is often composed with characters of a large amounts. An example is that traditional Chinese includes about 17k distinct tokens. Therefore, it could be expected to suffer from underfitting not only word embedding but also character embedding. Even worse, words in ideographic languages are often composed of 2-3 characters only, unlike words in alphabetic languages are longer but with smaller types of characters. Figure 1 provides the statistics in Chinese Sinica Corpus.
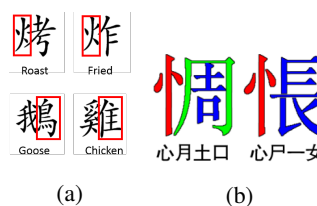

Figure 2: Example of compositionality of Chinese character components. (a) The same radical 火(fire) implies related meaning for 烤(roast) and 炸(fried). Components may also share similar semantics even they are different in graphs. 鳥 and 隹 are both refer to birds. (b) Cangjie input method. Each character can be presented as several keyboard inputs based on its components (e.g., 惆 is for 心+月+土+口).

The visual structure (or *glyph*) of a Chinese character contains rich semantics. A Chinese char-

---

∗ Equally contribution.

http://asbc.iis.sinica.edu.tw/index$_r$eadme.htm

acter is made up of several graphical components. Figure 2 shows some examples that components in characters represent similar semantic or pronunciation. In addition to glyphs, we propose to use the high-quality features provided by *Cangjie* input method to represent each character. Cangjie is a popular Chinese input method. Similar to radicals, characters are composed of 24 basic graphical units. Each unit is mapped to a corresponded letter key on a standard QWERTY keyboard. Building beyond character glyphs, one can intuitively guess the semantic of a word. Recent work (Chen et al., 2015; Xu et al., 2016; Yin et al., 2016; Liu et al., 2017; Su and Lee, 2017) have shown benefits of the compositionality at character level or visual feature of Chinese glyphs for some tasks.

In this work, we suggest that in the OOV scenario glyphs can be particularly useful. A key observation for solving OOV problem matches the intuition of human generalization in Chinese. When a Chinese user reads an unseen word or a character, by decomposing the structure, graphical components such as radicals for a character often help Chinese users understand the meaning and sometimes pronunciation of the character.

We study a novel application that recovers Chinese OOV word embeddings from glyphs. Our work is to answer a question : given the pre-trained word embeddings, *can we directly learn a mapping from word glyphs to their word embedding and generalize the mapping for the purpose of generating the embedding of OOV words?* We formulate it as a visual-to-text transfer learning problem and show that the visual structure of Chinese characters is helpful in learning Chinese OOV embeddings.

2 Related Work

Exploiting Structure of Chinese Characters Recent work have explored the use of Chinese character structure in different settings (E and Xiang, 2017; Liu et al., 2017; Dai and Cai, 2017). Several work aim to use character-level feature to enhance standard word embedding learning models (e.g., Word2Vec or GloVe). CWE (Chen et al., 2015) propose to use character-level formulation for words in training word embeddings; SCWE (Xu et al., 2016) and Li et al. (2015) extends to consider the relations of characters compositionally. MGE (Yin et al., 2016) and Shi et al. (2015) further includes radical information associated to

characters. Yu et al. (2017) jointly embed Chinese words, characters, and radicals. GWE (Su and Lee, 2017) proposes to extract feature from character bitmaps as the inputs of Word2Vec and GloVe. Our work is different from all of them, since we emphasize on generating the OOV word embeddings, which is not handled by them.

Learning Embedding for OOVs To handle OOV words, an approach is operating character level embeddings, then averages them into word embeddings (Kim et al., 2016; Wieting et al., 2016). Morphology-based approaches take advantage of meaningful linguistic substructures (Botha and Blunsom, 2014; Luong et al., 2013; Bhatia et al., 2016). Morphology-based approaches often struggle with those vocabularies lacking linguistic substructures such as names and transliterations of foreign language, which often appears as OOV words. In all the models above, just like Word2Vec (Mikolov et al., 2013c)), the embeddings meed to learned by training over a large corpus.

The most similar work is Mimick model (Pinter et al., 2017). By learning a character language generating model, guided by minimizing the distance between the output embedding of LSTMs and pre-trained word embeddings, Mimick shows feasibility of generating OOV word embedding from character compositions. However, Mimick is mainly from the view of alphabetic languages that does not consider glyphs. Chinese words often consist of short sequences composed of many kinds of tokens that are difficult for language model approaches to handle (see Figure 1) and could suffer from under-fitting.

3 Our Model: Glyph2Vec

We formulate the task of learning OOV embeddings as a transfer learning problem. Formally, given a Chinese vocabulary set $\mathcal{V}$ of size $|\mathcal{V}|$, and a pre-trained embeddings matrix $\mathcal{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$ where each word w_i is associated with a vector e_i of dimension d as training set $\{w_i, e_i\}_{i=1}^{|\mathcal{V}|}$. We aim to learn a mapping $F : w \rightarrow \mathbb{R}^d$, where F projects the input word to the d dimension embedding space such that $F(w_i) \approx e_i$. In testing, a word w_t may be out of $\mathcal{V}$, while the model is still obliged to predict the embedding e_t with $F(w_t)$.

Given the glyphs for a word $x = [c_j]_1^{|x|}$ as a sequence of character 2D bitmaps c provided according to $\mathcal{V}$, we can considering a function $g : x \rightarrow \mathbb{R}^k$ that transforms glyphs into vi-
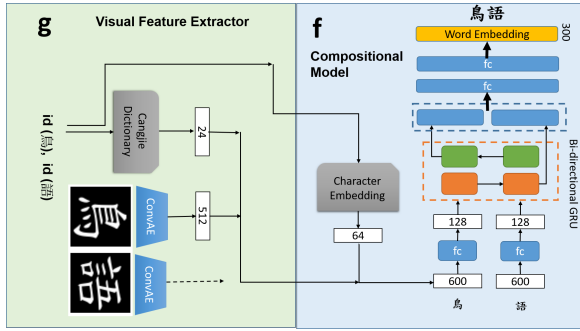
Figure 3: Complete network architecture of our Glyph2Vec. White boxes annotate the feature dimension of each *character*. Different features are combined by concatenating. GRU takes sequence of character feature as inputs.

sual features of k dimension. Another function $f : g(x) \to \mathbb{R}^d$ later maps the visual space to the word embedding space. The final embedding can be obtained with $e_i = F(x_i) = f(g(x_i))$, where input is glyph x_i. The overall framework is illustrated in Figure 3.

3.1 Visual Feature Extractor

We consider two implementations of visual feature extractor g.

ConvAE We adopt the convolutional autoencoder ConvAE (Masci et al., 2011) to capture the structure of characters bitmaps c. The architecture of the ConvAE follows Figure 6 in (Su and Lee, 2017). Eventually, the well-trained encoder is fixed as extractor that extracts 512-dimensional feature for every character c. The input bitmaps are 60×60 8-bit images in grayscale.

Cangjie Composition We propose to use Cangjie input codes as high-level annotations of characters, which can be easily collected from the input method dictionary. We construct a *Bag-of-Root (BoR)* vector for each character according to the Cangjie dictionary. Each BoR binary vector of 24 dimensions representing the roots that a character possesses.

3.2 Compositional Model: From Characters to Words

After the visual features of every character in a word are extracted, we still need to compose them to word level. A compositional model f takes a sequence of characters' visual feature and projects them onto the word embedding space. The right portion of Figure 3 shows the architecture of f. We construct a bi-directional RNN network with

GRU cells (Cho et al., 2014) to compute the expected word embedding over the character feature sequence. Finally, the 300D word embeddings are predicted. To calculate the loss for backpropagation, we adopt squared Euclidean distance between the prediction $F = f(g(x))$ and the gold word embedding w: $\|F(x) - w\|_2$.

3.3 Pre-trained Chinese Character Embedding

Unlike alphabetical languages, each Chinese character carries its own meaning. State-of-the-art Chinese word embedding models (Chen et al., 2015; Xu et al., 2016; Yin et al., 2016) often consider learning character embedding jointly. We demonstrate how to incorporate pre-trained character embedding to further improve the performance. The character embeddings are concatenated with the glyph features and the BoR Cangjie vectors as inputs. Character embedding is a huge embedding matrix. In Table 1, we summarized the required #parameters. We note that Glyph2Vec can infer OOV embedding directly from glyphs without character embedding.

Model	#Para
Mimick	1449k
Glyph2Vec + Pretrained Char	1362k
Glyph2Vec (w ConvAE)	517k
Glyph2Vec	306k

Table 1: Number of parameters required by Mimick and Glyph2Vec. Mimick based on character embedding and can be initialized with pre-trained character embedding (64 dimension). Note that ConvAE can be pre-trained and discarded during training Glyph2Vec.

4 Experiment

4.1 Setup

We adopt the Word2Vec traditional Chinese 300d word embedding pre-trained on public-available Sinica Corpus 4.0 which includes about 10M tokens. For optimization, we train 100 epochs with RMSProp optimizer with learning rate 4e-4 with batch-size 128. We note the models compared in the following experiments here. **M** is for Mimick baseline (Pinter et al., 2017) based on the authors' code. For the proposed feature, we test several combinations. **C** is for using Cangjie BoR vector; **V** is for using glyph visual feature; **Char** is for appending pre-trained character embedding. We uti-
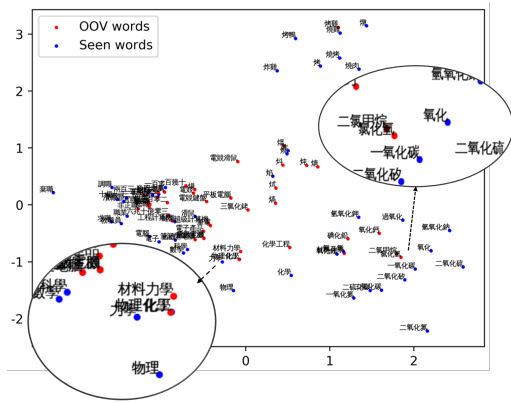
---

http://asbc.iis.sinica.edu.tw/

Figure 4: Principal component analysis visualization of the produced word embedding. Zoom in for better resolution.

lize the embeddings from Polyglot (Al-Rfou et al., 2013).

As a sanity check, in Fig. 4 we visualize the embedding of seen and OOV words. One could observe meaningful clusters that have similar visual structure. For example, 烤雞 (roast chicken) could be mapped with 烤鴨 (roast duck) because 雞 (chicken) and 鴨 (duck) have different glyphs both about bird. Some cooking verbs that have the radical 火 (fire) like 烤 (roast) and 燒烤 (roast) are also mapped closely. Some unseen characters (or "words" with only one character) can also be predicted reasonably.

5 Nearest Neighbor Examples

We qualitatively analyze Glyph2Vec with nearest neighbor (NN) sanity check. Table 2 shows the results of retrieved nearest neighbors with OOV word queries for Mimick and our Glyph2Vec embeddings (using **V**), respectively.

We observe Glyph2Vec is able to model visual semantic by associating those characters that share related visual features since Glyph2Vec learns from the images of characters. For example, 鰻(eel) in 蛇鰻(snake-eel) shares the radicals of 魚(fish) with 石鱸(Haemulidae, fish name). 銠(Rh) and 氯(Cl) in 三氯化銠(RhCl3) associate some visual features relate to chemicals like 金 in 鈰(Ce), 气 in 氟(F), 酉 in 酸(acid), and more.

On the other hand, we observe some properties including composition (e.g., numbers) and character semantic that both Glyph2Vec and Mimick can provide. (1) Composition: composing characters that have very different mean-

ing after splitting them. For instance, 茲尼約夫 is a transliteration of Seleznev (Russian name), for which every character is meaningless alone but a meaningful transliteration when combined. With character-level compositional model in Glyph2Vec, it could be retrieved given 克羅迪歐(Claudio, western name). Moreover, Glyph2Vec preserves correct meaning of a character when attaching with the other characters. For example, 驟(abrupt)減(decrease) can retrieve 減少(cut back) and 減低(reduce) properly when 減 (subtract) is associated to different characters. (2) character semantic: associating different characters with similar meaning. For example, 道(street) is related to 巷(lane) or 弄(alley) and they are retrieved by our model given 學府二道(Xuefu 2nd Street) as the OOV word even though the characters look completely different.

5.1 Joint Tagging of Parts-of-Speech (POS) and Morphosyntactic Attributes

We follow the experiment protocols of parts-of-speech tagging and morphosyntactic attributes tagging stated in Mimick (Pinter et al., 2017) for this experiment. There are two parts-of-speech tagging tasks based on the Chinese thread of Universal Dependencies (UD) scheme (De Marneffe et al., 2014). To avoid tuning towards those OOV words, we consider the similar evaluation protocols of generalized zero-shot learning (Chao et al., 2016; Xian et al., 2017) that the embedding of not only unseen but also seen words need to be generated. Both word-level LSTM and character LSTM are reported (Table 3). With visual feature available, Glyph2Vec consistently outperforms Mimick. On the other hand, we observe using pretrained character embedding only helps on accuracy of seen words but not OOV words, which suggests that it is necessary for a module like Mimick or Glyph2Vec to learn to compose characters for OOV words.

5.2 Wikipedia Title Classification

As we introduced in Sec. 1, in real-world scenario Chinese systems could suffer from severe OOV problem. An example is Wikipedia encyclopedia. It contains lots of rarewords that easily become OOV words such as terminologies, scientific names, geography locations, ... etc. We utilize Wikipedia Title dataset (Liu et al., 2017) to

Query Word	Top 5 Nearest Neighbors
一百幾十(numbers)	平樂縣(city) 四千多(numbers) 通山縣(county) 七千多 一千三百二十多萬(numbers)
全劇(drama)	殘夢(dream) 活脫(lividly) 黃曉若(name) 茱莉紐瑪爾(Juliette Binoche) 市川實(name)
驟減(slump)	水蓄存(water resource) 猴蝦(shrimp) 投藥量(dosage) 百萬分之八(proportion) 河塘(pond)
供職(provided job)	猴蝦(shrimp) 管制課(office) 鄭龍營(name) 劉百真(name) 疾管課(office)
蛇鰻(snakebird)	廣鹽性(euryhaline) 石鱸(fish) 紅鱈(fish) 蒼燕鷗(gull) 沙蠶(worm)
克羅迪歐(Claudio)	查氏(Cha) 塞立格(Selig) 薩梅(Same) 歐卡南(Okana) 拉杜爾(Ladur)
三氯化銠(RhCl3)	炆(stew) 粘稠(viscous) 投藥量(medicine) 許敏昌(name) 放射線菌(bacteria)
杳無蹤跡(idiom)	潛水鏡(goggles) 捉蟹(catch crab) 堤邊(riverside) 十點多(time) 溪兩旁(riverside)
學府二道(street)	魏昭雄(name) 猴蝦(shrimp) 地仙(person) 陳建村(name) 張玉田(name)
一百幾十(numbers)	一百多(numbers) 兩百多(numbers) 二十多(numbers) 八十多(numbers) 五十多(numbers)
全劇(drama)	齣(unit for drama) 裸戲(naked play) 舞劇(dance drama) 歌舞劇(musical drama) 戲碼(drama)
驟減(slump)	減(slump) 減少(slump) 逐年(year by year) 大幅度(dramatically) 減低(slump)
供職(provided job)	現職(job) 專職(job) 軍職(military service) 聘任(hire) 任用(hire)
蛇鰻(snakebird)	魚類(fish) 廣鹽性(euryhaline) 石鱸(fish) 筍殼魚(fish) 性魚(fish)
克羅迪歐(Claudio)	柯普奇夫(Puchkov) 齊默特(Chimet) 采夫(Tsev) 茲尼約夫(Seleznev) 伊特金(Itkine)
三氯化銠(RhCl3)	無機酸(inorganic acid) 鈰(Ce) 氟二氯(FCl2) 化學式(chemical Eq.) 陽極板(anode plate)
杳無蹤跡(idiom)	無可奈何(idiom) 未必盡然(idiom) 不足為奇(idiom) 莫可奈何(idiom) 處之泰然(idiom)
學府二道(street)	三十九弄(street) 二二一巷(street) 二八五巷(street) 一百七十四巷(street) 三十弄(street)

Table 2: Nearest neighbors examples retrieved by Mimick (upper) and Glyph2Vec (lower). Top 5 NNs are listed. Words are translated or given with explanation.

		POS		Attr.
Model		Acc	**OOV Acc**	F1
Word-based LSTM				
UNK		0.888	0.474	0.931
M		0.909	0.617	0.934
V		**0.924**[†]	0.741	0.946
C		0.921	0.709	0.942
V + C		**0.924**[†]	**0.747**[†]	**0.950**[†]
Character-based LSTM				
UNK		0.910	0.618	0.948
M		0.929	0.768	0.954
V		0.933	0.800	0.955
V + C		0.935	0.801	0.956
M (Char)*		0.931	0.768	0.955
V + Char		0.936	0.805	0.958
C + Char		0.934	0.794	0.958
V + C + Char		**0.938**[†]	**0.810**[†]	**0.959**[†]

Table 3: Results for parts-of-speech and morphosyntactic attributes tagging based on word-level and character-level LSTM. *Initializing Mimick with pretrained character embedding. [†]Best model passing significant test against Mimick (M) with p-value < 0.05.

Model	Acc
UNK	0.431
M	0.497
C	0.499
V	0.501
V + C	0.513
V + C + Char	**0.516**[†]

Table 4: Wikipedia Title Classification Accuracy

study the problem. The dataset is a collection of 593K Chinese articles from Wikipedia and categorizing them into 12 classes based on their titles. We preprocessed the data by removing punctuation, special characters, and other non-Chinese in-

stances, and turning Arabic numbers into Chinese text. We use opensource Jieba toolkit to segment each title into words. 52.5% are OOV based on Sinica Corpus, and we generate their embeddings by Glyph2Vec.

We construct a neural network classifier with the generated word embedding as input to evaluate our method. The classifier is consist of 3 fully-connected (FC) layers on top of the averaged word embedding of titles. Results are shown in Table 4. With glyph feature and Cangie BoR feature provided, the performance could be improved significantly compared to neglecting OOV (as UNK) in such challenging setting.

6 Conclusion

In this work, we propose a multi-modal framework that expand pre-trained embedding space to include OOV words using character visual features such as Cangjie feature and Chinese character glyphs. We have demonstrated the effectiveness of Glyph2Vec on traditional Chinese, and we believe Glyph2Vec can also be applied to other ideographic languages to handle OOV words as well.

https://github.com/fxsjy/jieba

We note that the accuracy cannot be compared with the report in (Liu et al., 2017) since they did not consider OOV and char/word embeddings. Here we only use the dataset to examine the performance of OOV embedding.

For simplified Chinese, we suggest users to first translate into traditional Chinese since traditional characters have richer structures and probably more semantics can be extracted through Glyph2Vec.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 183–192.

Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. 2016. Morphological priors for probabilistic neural word embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 490–500.

Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1899–1907.

Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. 2016. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. *ECCV*.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huan-Bo Luan. 2015. Joint learning of character and word embeddings. In *IJCAI*, pages 1236–1242.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Falcon Dai and Zheng Cai. 2017. Glyph-aware embedding of chinese characters. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP, Copenhagen, Denmark, September 7, 2017*, pages 64–69.

M.-C De Marneffe, T Dozat, N Silveira, K Haverinen, F Ginter, Joakim Nivre, and C.D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. pages 4585–4592.

Shijia E and Yang Xiang. 2017. Chinese named entity recognition with character-word mixed embedding. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 2055–2058.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2741–2749.

Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. 2015. Component-enhanced chinese character embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 829–834.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning character-level compositionality with visual features. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2059–2068. Association for Computational Linguistics.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013c. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword rnns. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical embedding: Delving deeper to chinese radicals. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 594–598.

Tzu-ray Su and Hung-yi Lee. 2017. Learning chinese word representations from glyphs of characters.

In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 264–273.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1504–1515.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning - the good, the bad and the ugly. *CVPR*.

Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. 2016. Improve chinese word embeddings by exploiting internal structure. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1050.

Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. 2016. Multi-granularity chinese word embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 981–986.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 286–291.