Exploiting Word Internal Structures for Generic Chinese Sentence Representation

Shaonan Wang[1,2], **Jiajun Zhang**[1,2], **Chengqing Zong**[1,2,3]
[1] National Laboratory of Pattern Recognition, CASIA, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China
{shaonan.wang,jjzhang,cqzong}@nlpr.ia.ac.cn

Abstract

We introduce a novel mixed character-word architecture to improve Chinese sentence representations, by utilizing rich semantic information of word internal structures. Our architecture uses two key strategies. The first is a mask gate on characters, learning the relation among characters in a word. The second is a max-pooling operation on words, adaptively finding the optimal mixture of the atomic and compositional word representations. Finally, the proposed architecture is applied to various sentence composition models, which achieves substantial performance gains over baseline models on sentence similarity task.

1 Introduction

To understand the meaning of a sentence is a prerequisite to solve many natural language processing problems. Obviously, this requires a good representation of the meaning of a sentence. Recently, neural network based methods have shown advantage in learning task-specific sentence representations (Kalchbrenner et al., 2014; Tai et al., 2015; Chen et al., 2015a; Cheng and Kartsaklis, 2015) and generic sentence representations (Le and Mikolov, 2014; Hermann and Blunsom, 2014; Kiros et al., 2015; Kenter et al., 2016; Wang et al., 2017). To learn generic sentence representations that perform robustly across tasks as effective as word representations, Wieting et al. (2016b) proposes an architecture based on the supervision from the Paraphrase Database (Ganitkevitch et al., 2013).

Despite the fact that Chinese has unique word internal structures, there is no work focusing on learning generic Chinese sentence representations. In contrast to English, Chinese characters contain rich information and are capable of indicating semantic meanings of words. As illustrated in Figure 1, the internal structures of Chinese words express two characteristics: (1) Each character in a word contribute differently to the compositional word meaning (Wong et al., 2009) such as the word "出租车(taxi)". The first two characters "出租(rent)" are descriptive modifiers of the last character "车(car)", and make the last character play the most important role in expressing word meaning. (2) The atomic and compositional representations contribute differently to different types of words (MacGregor and Shtyrov, 2013). For instance, the meaning of "机场(airport)", a low-frequency word, can be better expressed by the compositional word representation, while the non-transparent word "虹桥(Hongqiao)" is better expressed by the atomic word representation.

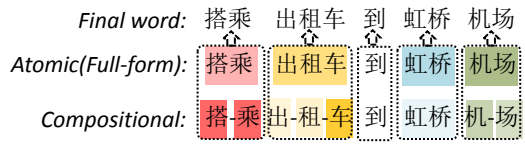

Figure 1: An example sentence that consists of five words as "搭乘(take) 出租车(taxi) 到(to) 虹桥(Hongqiao) 机场(airport)". Most of these words are compositional, namely word "搭乘" consists of characters "搭(take)" and "乘(ride)", word "出租车" constitutes characters "出(out)", "租(rent)" and "车(car)", and word "机场" is composed of characters "机(machine)" and "场(field)". The color depth represents (1) contributions of each character to the compositional word meaning, and (2) contributions of the atomic (which ignore inner structures) and compositional word to the final word meaning. The deeper color means more contributions.

The word internal structures have been proven to be useful for Chinese word representations. Chen et al. (2015b) proposes a character-enhanced word representation model by adding the averaged character embeddings to the word embedding. Xu et al. (2016) extends this work by using weighted character embeddings. The weights are cosine similarities between embeddings of a word's English translation and its constituent characters' English translations. However, their work calculates weights based on a bilingual dictionary, which brings lots of mistakes because words in two languages do not mantain one-to-one relationship. Furthermore, they only consider the first characteristic of word internal structures, but ignore the contributions of the atomic and compositional word to the final word meaning. Similar ideas of adaptively utilizing character level informations have also been investigated in English recently (Hashimoto and Tsuruoka, 2016; Rei et al., 2016; Miyamoto and Cho, 2016). It should be noted that these studies are not focus on learning sentence embeddings.

In this paper, we explore word internal structures to learn generic sentence representations, and propose a mixed character-word architecture which can be integrated into various sentence composition models. In the proposed architecture, a mask gate is employed to model the relation among characters in a word, and pooling mechanism is leveraged to model the contributions of the atomic and compositional word embeddings to the final word representations. Experiments on sentence similarity (as well as word similarity) demonstrate the effectiveness of our method. In addition, as there are no publicly available Chinese sentence similarity datasets, we build a dataset to directly test the quality of sentence representations. The code and data will be publicly released.

2 Model Description

The problem of learning compositional sentence representations can be formulated as $g^{comp} = f(x)$, where f is the **composition function** which combines the **word representations** $x = \langle x_1, x_2, ..., x_n \rangle$ into the compositional sentence representation g^{comp}.

2.1 Mixed Character-Word Representation

In our method, the final word representation is a fusion of the atomic and compositional word embeddings. The atomic word representation is calculated by projecting word level inputs into a high-dimensional space by a look up table, while the compositional word representation is computed as a gated composition of character representations:

$$x_i^{comp} = \sum_{j=1}^{m} v_{ij} \cdot c_{ij}, \tag{1}$$

where c_{ij} is the *j*-th character representation in the *i*-th word. The mask gate $v_{ij} \in \mathbb{R}^d$ controls the contribution of the *j*-th character in the *i*-th word. This is achieved by using a feed-forward neural network operated on the concatenation of a character and a word, under the assumption that the contribution of a character is correlated with both character itself and its relation with the corresponding word:

$$v_{ij} = tanh(W \cdot [c_{ij}; x_i]), \tag{2}$$

where $W \in \mathbb{R}^{d \times 2d}$ is a trainable parameter. The proposed mask gate is a vector instead of a single value, which introduces more variations to character meaning in the composition process.

Then, the atomic and compositional word representations are mixed with max-pooling:

$$x_i^{final} = \max_{k=1}^{d}(x_{ik}^{atomic}, x_{ik}^{comp}), \tag{3}$$

the *max* is an element-wise function to capture the most important features (i.e., the highest value in each dimension) in the two word representations.

2.2 Sentence Composition Model

Given word embeddings, we make a systematic comparison of five different composition models for sentence representations as follows:

1. $g = Average(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$
2. $g = Matrix(x) = \frac{1}{n}\sum_{i=1}^{n} f(W_m x_i)$
3. $g = Dan(x) = f(W_d(\frac{1}{n}\sum_{i=1}^{n} x_i) + b)$
4. $g = RNN(x) = f(W_x x_i + W_h h_{i-1} + b)$
5. $g = LSTM(x) = o_t \odot f(c_i)$, where $c_i = f_i \cdot c_{i-1} + i_i \cdot \widetilde{c_i}$ and $\widetilde{c_i} = \sigma(W_{xc} x_i + W_{hc} h_{i-1})$

Average model, as the simplest composition model, represents sentences with averaged word vectors which are updated during training. The

Matrix and Dan models are proposed in Zanzotto et al. (2010) and Iyyer et al. (2015), respectively. By using matrix transformations and nonlinear functions, the two models represent sentence meaning in a more flexible way (Wang and Zong, 2017). We also include RNN and LSTM models, which are widely used in recent years. The parameters $\{i_t, f_t, o_t\} \in \mathbb{R}^d$ denote the input gate, the forget gate and the output gate, respectively. $c_t \in \mathbb{R}^d$ is the short-term memory state to store the history information. $\{W_m, W_d, W_x, W_h, W_{xc}, W_{hc}\} \in \mathbb{R}^{d \times d}$ are trainable parameters. h_{i-1} denotes representations in hidden layers. Sentence representations in RNN and LSTM models are hidden vectors of the last token.

2.3 Objective Function

This paper aims to learn the general-purpose sentence representations based on supervision from Chinese paraphrase pairs. Following the approach of Wieting et al. (2016b), we employ the max-margin objective function to train sentence representations by maximizing the distance between positive examples and negative examples.

3 Experimental Setting and Dataset

3.1 Experimental Setting

We construct four groups of models (G1˜G4) which serve as baselines to test the proposed mixed character-word models (*G5*). Group *G1* includes six baseline models, which have shown impressive performance in English. The first two are averaged word vectors and averaged character vectors. Followed by PV-DM model which uses auxiliary vectors to represent sentences and trains them together with word vectors, and FastSent model which utilizes a encoder-decoder model and encodes sentences as averaged word embeddings. The last two are Char-CNN model which is CNN model with character n-gram filters, and Charagram model which represents sentences with a character n-gram count vector. Group *G2* are the sentence representation models proposed by Wieting et al. (2016b), which utilize only word level information. We also compared our method with word representation models of Chen et al. (2015b) and Xu et al. (2016) in Group *G3* and *G4* respectively, by incorporate them into five sentence composition models in Section 2.2.

In all models, the word and character embeddings are initialized with 300-dimension vectors trained by Skip-gram model (Mikolov et al., 2013) on a corpus with 3 billion Chinese words. All models are implemented with Theano (Bergstra et al., 2010) and Lasagne (Dieleman et al., 2015), and optimized using Adam (Kingma and Ba, 2014). The hyper-parameters[1] are selected by testing different values and evaluating their effects on the development set. In this paper, we run all experiments 5 times and report the mean values.

3.2 Training Dataset

The training dataset is a set of paraphrase pairs in which two sentences in each pair represent the same meanings. Specifically, we extract Chinese paraphrases in machine translation evaluation corpora NIST2003[2] and CWMT2015[3]. Moreover, we select aligned sub-sentence pairs between paraphrases to enlarge the training corpus. Specifically, we first segment the sentences into sub-sentences according to punctuations of *comma, semicolon, colon, question mark, ellipses, and periods*. Then we pair all sub-sentences between a paraphrase and select sub-sentence pairs (s_1, s_2) which satisfy the following two constraints: (1) the number of overlapping words of sub-sentence s_1 and s_2 should meet the condition: $0.9 > len(overlap(s_1, s_2))/min(len(s_1), len(s_2)) > 0.2$, where $len(s)$ denotes the number of words in sentence s; (2) the relative length of sub-sentence should meet the condition: $max(len(s_1), len(s_2))/min(len(s_1), len(s_2)) <= 2$. Finally, we get 30,846 paraphrases (18,187 paraphrases from NIST including 11,413 sub-sentence pairs, and 12,659 paraphrases from CWMT which include 7,912 sub-sentence pairs).

3.3 Testing Dataset

We also build the testing dataset, which are sentence pairs collocated with human similarity ratings. We choose candidate sentences from the People's Daily and Baidu encyclopedia corpora. To assure sentence pairs to be representative of the full variation in semantic similarity, we choose

[1] We use a mini-batch of 25 and tune the initial learning rate over {0.001, 0.005, 0.0001, 0.0005}. For the Dan and the Matrix models, we tune over activation function (tanh or linear or rectified linear unit) and number of layers (1 or 2).

[2] which contains 1,100 English sentences with 4 Chinese translations and can be found at: https://catalog.ldc.upenn.edu/LDC2006T04

[3] which contains 1,859 English sentences with 4 Chinese translations and can be found at: http://www.ai-ia.ac.cn/cwmt2015/evaluation.html

high similarity sentence pairs[4] and then randomly pair the single sentences to construct low similarity sentence pairs. To collect human similarity ratings for sentence pairs, we use online questionnaire[5] and follow the gold standard[6] to guide the rating process of participants. The subjects are paid 7 cents for rating each sentence pair within a range of 0 5 score. In total, we obtain 104 valid questionnaires and every sentence pair is evaluated by average 8 persons. We use the average subjects' ratings for one paraphrase as its final similarity score, and the higher score means that the two sentences have more similar meaning. We then randomly partition the datasets into test and development splits in 9:1.

4 Results and Discussion

We use the Pearson's correlation coefficient to examine relationships between the averaged human ratings and the predicted cosine similarity scores of all models. Moreover, the Wilcoxon's test shows that significant difference ($p < 0.01$) exits between our models with baseline models.

From Table 1, we can see superiority of the proposed mixed character-word models (*G5*), which have significantly improved the performance over both word and character-word based models. This result indicates that it is important to find the appropriate way to fuse character and word level informations. Using mask gate alone and max pooling alone yield an improvement of 1.05 points and 0.83 points respectively, and using both strategies improves the averaged character-word models by 1.52 points. Another observation is that models with character level information (*G3, G4, G5*) perform better than word based models (*G2*), which indicates the great potential of Chinese characters in learning sentence representations. Comparing different composition functions, we can see that two simple models outperform others in all groups: the DAN model and the Matrix model. The simplest Average model achieves competitive results while the most complex LSTM model does not show advantages.

Group	Model	Test
G1: Baselines	Add (character)	0.6737
	Add (word)	0.7518
	PV-DM (Le and Mikolov, 2014)	0.7561
	FastSent (Hill et al., 2016)	0.7369
	Char-CNN (Kim et al., 2016)	0.8095
	Charagram(Wieting et al., 2016a)	0.8382
G2: Word level (Wieting et al., 2016b)	Average	0.8199
	Matrix	0.8382
	Dan	0.8385
	RNN	0.8121
	LSTM	0.7834
G3: Averaged Character-Word (Chen et al., 2015)	Average	0.8245
	Matrix	0.8427
	Dan	0.8407
	RNN	0.8185
	LSTM	0.7895
G4: Weighted Character-Word (Xu et al., 2016)	Average	0.8196
	Matrix	0.8428
	Dan	0.8413
	RNN	0.8344
	LSTM	0.7858
G5: Mixed Character-Word (Ours)	**Average**	**0.8471**
	Matrix	**0.8517**
	Dan	**0.8521**
	RNN	**0.8408**
	LSTM	**0.8000**

Table 1: Correlation coefficients of model predictions with subject similarity ratings on Chinese sentence similarity task. The bold data refers to best among models with same composition function.

4.1 Effects of Mask Gate and Max Pooling

The mask gate assigns different weights to characters in a word, hopefully leading to better word representations. To intuitively show effects of the mask gate, we check characters whose l2-norm increase after applying the mask gate approach. We find that characters like "罪(crime)" in "罪状(guilty)", "虎 (tiger)" in "美洲虎 (jaguar)" and "瓜 (melon)" in "黄瓜 (cucumber)" achieve more weights. The above results show that the mask gate approach successfully model the first characteristic of word internal structure (i.e., assigning more weights to key characters). To quantitatively display the results, we extract the word representations calculated by the five composition models in four different groups and evaluate their quality on WordSim-297 dataset[7] using the Pearson correlation method. As shown in Table 2, the mask gate approach significantly improves the quality of word representations.

[4]Here we choose high similarity sentence pairs by using edit distance and human post-processing.

[5]https://wj.qq.com/

[6]http://alt.qcri.org/semeval2015/task2/index.php?id=semantic-textual-similarity-for-english

[7]https://github.com/Leonard-Xu/CWE/tree/master/data

	G2	*G3*	*G4*	*G5*(Ours)
Average	0.4311	0.4584	0.4789	**0.5245**
Dan	0.4470	0.5410	0.5561	**0.5716**
Matrix	0.4496	0.5458	0.5548	**0.5694**
RNN	0.4562	0.5656	0.5550	**0.5674**
LSTM	0.4535	0.5674	0.5627	**0.5734**

Table 2: Correlation coefficients of model predictions with subject similarity ratings on Chinese word similarity task, where *G2* $\sim$ *G5* are the same as in Table 1.

The max-pooling approach is supposed to model different contributions of the atomic and compositional word vectors to the final word vector. To find out what have max-pooling method learned, we use contribution weights by calculating cosine similarities between the final word representation with the atomic and compositional word representations. The results show interesting relationships with word frequency. For high-frequency words, the contribution of compositional word representations are more dominant. While for low-frequency words, both high[8] and low contribution ratios of compositional word representations can be found. When looking into the words with the most lowest ratio, we find a large portion of English abbreviations like *NBA*, *BBC*, *GDP* etc., and a portion of metaphor words like "挂靴(retire, hanging boots)" and "扯皮(wrangle, pull skin)". Both kinds of these words are non-transparent, which indicates that the max-pooling method can successfully model the second characteristic of word internal structure and encode word transparency to some extent.

5 Conclusion and Further work

In this paper, we introduce a novel mixed character-word architecture to improve generic Chinese sentence representations by exploiting the complex internal structures of words. Extensive experiments and analyses have indicated that our models can encode word transparency and learn different semantic contributions across characters. We have also created a dataset to evaluate composition models of Chinese sentences, which could advance the research for related fields.

Future work includes applying the proposed method to other aspects of nominal semantics, such as understanding compound nouns in other languages, and to explore the compositionality of words and compounds.

[8]The high ratio is more reasonable because low-frequency words generally learn poor atomic word representations.

Acknowledgement

The research work has been supported by the Natural Science Foundation of China under Grant No. 61333018 and No. 61403379.

References

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: A cpu and gpu math compiler in python. *In Proceedings of the Python for Scientific Computing Conference (SciPy)*.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Shiyu Wu, and Xuanjing Huang. 2015a. Sentence modeling with gated recursive neural network. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,*, pages 793–798.

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

Jianpeng Cheng and Dimitri Kartsaklis. 2015. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,*, pages 1531–1542.

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, et al. 2015. Lasagne: First release. *Zenodo: Geneva, Switzerland*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2016. Adaptive joint learning of compositional and non-compositional phrase embeddings. *Proceedings of the* 54^{th} *Annual Meeting of the Association for Computational Linguistics,*, pages 205–215.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. *Proceedings of the* 52^{nd} *Annual Meeting of the Association for Computational Linguistics,*, pages 58–68.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *Proceedings of NAACL-HLT 2016,*, pages 1367–1377.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. *Proceedings of the* 53^{rd} *Annual Meeting of the Association for Computational Linguistics*, pages 1681–1691.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the* 52^{nd} *Annual Meeting of the Association for Computational Linguistics,*, pages 655–665.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. *Processings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, pages 3294–3302.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *Proceedings of the* 31^{st} *International Conference on Machine Learning*, pages 1188–1196.

Lucy J MacGregor and Yury Shtyrov. 2013. Multiple routes for compound word processing in the brain: evidence from eeg. *Brain and language*, 126(2):217–229.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,*, pages 1992–1997.

Marek Rei, Gamal KO Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. *Proceedings of COLING 2016, the* 26^th *International Conference on Computational Linguistics,*, pages 309–318.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2017. Learning sentence representation with guidance of human attention. *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Shaonan Wang and Chengqing Zong. 2017. Comparison study on critical components in composition model for phrase representation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(3):16.

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

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016b. Towards universal paraphrastic sentence embeddings. *ICLR*.

Kam-Fai Wong, Wenjie Li, Ruifeng Xu, and Zhengsheng Zhang. 2009. Introduction to chinese natural language processing. *Synthesis Lectures on Human Language Technologies*, 2(1):1–148.

Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. 2016. Improve chinese word embeddings by exploiting internal structure. *Proceedings of NAACL-HLT 2016,*, pages 1041–1050.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. *Proceedings of the* 23^{rd} *International Conference on Computational Linguistics,*, pages 1263–1271.