Semantic Clustering and Convolutional Neural Network for Short Text Categorization

Peng Wang, Jiaming Xu, Bo Xu, Cheng-Lin Liu, Heng Zhang
Fangyuan Wang, Hongwei Hao
{peng.wang, jiaming.xu, boxu}@ia.ac.cn, liucl@nlpr.ia.ac.cn
{heng.zhang, fangyuan.wang, hongwei.hao}@ia.ac.cn
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, P.R. China

Abstract

Short texts usually encounter data sparsity and ambiguity problems in representations for their lack of context. In this paper, we propose a novel method to model short texts based on semantic clustering and convolutional neural network. Particularly, we first discover semantic cliques in embedding spaces by a fast clustering algorithm. Then, multi-scale semantic units are detected under the supervision of semantic cliques, which introduce useful external knowledge for short texts. These meaningful semantic units are combined and fed into convolutional layer, followed by max-pooling operation. Experimental results on two open benchmarks validate the effectiveness of the proposed method.

1 Introduction

Conventional texts mining methods based on bag-of-words (BoW) easily encounter data sparsity and ambiguity problems in short text modeling (Chen *et al.*, 2011), which ignore semantic relations between words (Sriram *et al.*, 2010). How to acquire effective representation for short text has been an active research issue (Chen *et al.*, 2011; Phan *et al.*, 2008).

In order to overcome the weakness of BoW, researchers have proposed to expand the representation of short text using latent semantics, where the words are mapped to distributional representations by Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003) and its extensions. Phan et al. (2008) presented a general framework to expand the short and sparse text by appending topic names discovered using LDA. Yan et al. (2013) presented a variant of LDA, dubbed Biterm Topic Model (BTM), especially for short text modeling to alleviate the problem of sparsity. However, the methods discussed above still view a piece of text as BoW. Therefore, they are not effective in capturing fine-grained semantic information for short texts modeling.

Recently, neural network related methods have received much attention, including learning word embeddings (Bengio *et al.*, 2003; Mikolov *et al.*, 2013a) and performing semantic composition to obtain phrase or sentence level representations (Collobert *et al.*, 2011; Le and Mikolov, 2014). For learning word embedding, the training objective of continuous Skip-gram model (Mikolov *et al.*, 2013b) is to predict its context. Thus, the co-occurrence information can be effectively used to describe a word, and each component of word embedding might have a semantic or grammatical interpretation.

In embedding spaces, semantically close words are likely to cluster together and form semantic cliques (or word embedding cliques). Moreover, the embedding spaces exhibit linear structure that the word vectors can be meaningfully combined using simple additive operation (Mikolov *et al.*, 2013b), for example:

$$vec\,(Germany) + vec\,(Capital) \approx vec\,(Berlin) \quad (1)$$

$$vec(Athlete) + vec\,(Football) \approx vec\,(Football\_Player) \quad (2)$$

The above examples indicate that the additive composition can often produce meaningful results. In Equation (1), the token $'Berlin'$ can be viewed that it has an embedding offset $vec\,(Capital)$ to the token $'Germany'$ in embedding spaces. Furthermore, the embedding offsets represent the syntactical and semantic relations among words.

In this paper, we propose a method to model short texts using semantic clustering and convolutional neural network (CNN). Firstly, the fast clustering algorithm (Rodriguez and Laio, 2014), based on searching density peaks, is utilized to cluster word embeddings and discover semantic cliques, as shown in Figure 1. Then semantic composition is performed over n-gram embeddings to

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 352–357, Beijing, China, July 26-31, 2015.

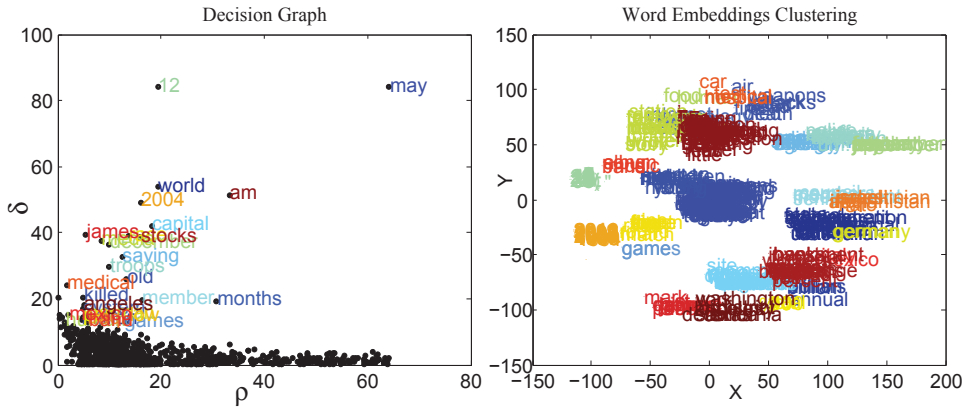

Figure 1: Fast clustering based on density peaks of embeddings

detect candidate Semantic Units[1](abbr. to SUs) appearing in short texts. The part of candidate SUs meeting the preset threshold are chosen to constitute semantic matrices, which are used as input for the CNN, otherwise dropout. In this stage, semantic cliques are used as supervision information, which guarantee meaningful SUs can be extracted.

The motivation of our work is to introduce extra knowledge by pre-trained word embeddings and fully exploit the contextual information of short texts to improve their representations. The main contributions include: (1) semantic cliques are discovered using fast clustering method based on searching density peaks; (2) for fine-tuning multi-scale SUs, the semantic cliques are used to supervise the selection stage.

The remainder of this paper is organized as follows. The related works are briefly reviewed in Section 2. Section 3 introduces the semantic clustering based on fast searching density peaks. Section 4 describes the architecture of the proposed method. Section 5 demonstrates the effectiveness of our method with experiments. Finally, concluding remarks are offered in Section 6.

2 Related Works

Traditional statistics-based methods usually fail to achieve satisfactory performance for short texts classification due to their sparsity of representations (Sriram *et al.*, 2010). Based on external Wikipedia corpus, Phan et al. (2008) proposed a method to discover hidden topics using LDA and expand short texts. Chen et al. (2011) proved that leveraging topics at multiple granularity can model short texts more precisely.

Neural networks have been used to model languages, and the word embeddings can be learned simultaneously (Mnih and Teh, 2012). Mikolov et al. (2013b) introduced the continuous Skip-gram model that is an efficient method for learning high quality word embeddings from large-scale unstructured text data. Recently, various pre-trained word embeddings are publicly available, and many composition-based methods are proposed to induce the semantic representation of texts. Le and Mikolov (2014) presented the Paragraph Vector algorithm to learn a fixed-size feature representation for documents.

Kalchbrenner et al. (2014) introduced the Dynamic Convolutional Neural Network (DCNN) for modeling sentences. Their work is closely related to our study in that k-max pooling is utilized to capture global feature vector and do not rely on parse tree. Kim (2014) proposed a simple improvement to the convolutional architecture that two input channels are used to allow the employment of task-specific and static word embeddings simultaneously.

Zeng et al. (2014) developed a deep convolutional neural network (DNN) to extract lexical and sentence level features, which are concatenated and fed into the softmax classifier. Socher et al. (2013) proposed the Recursive Neural Network (RNN) that has been proven to be efficient in terms of constructing sentences representations. In order to reduce the overfitting of neural network especially trained on small data set, Hinton et al. (2012) used random dropout to prevent

[1]Semantic units are defined as n-grams which have dominant meaning of text. With n varying, multi-scale contextual information can be exploited.

complex co-adaptations. To exploit more structure information of text, based on CNN and direct embedding of small text regions, an alternative mechanism for effective use of word order for text categorization was proposed (Johnson and Zhang, 2014).

Although the popular methods can capture high-order information and word relations to produce complex features, they cannot guarantee the classification performance for very short texts. In this paper, we design a method to exploit more contextual information for short text classification using semantic clustering and CNN.

3 Semantic Clustering

Since the neighbors of each word are semantically related in embedding space (Mikolov *et al.*, 2013b), clustering methods (Rodriguez and Laio, 2014) can be used to discover semantic cliques. For implementation, two quantities of data point i are computed, include: local density ρ_i, defined as follows,

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{3}$$

where d_{ij} is the distance between data points, d_c is a cutoff distance. Furthermore, distance δ_i from points of higher density is measured by,

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i} (d_{ij}) & , \ if \ \ \rho_i < \rho_{\max} \\ \max\limits_{j}(d_{ij}) & , \ \ otherwise \end{cases} \tag{4}$$

An example of semantic clustering is illustrated in Figure 1. The decision graph shows the two quantities ρ and δ of each word embedding. According to the definitions above, these word embeddings with large ρ and δ simultaneously are chosen as cluster centers, which are labeled using the corresponding words.

4 Proposed Architecture

As shown in Figure 2, the proposed architecture use well pre-trained word embeddings to initialize the lookup table, and higher levels extract more complexity features.

For short text $S = \{w_1, w_2, \cdots, w_N\}$, its projected matrix $\mathbf{PM} \in \mathbf{R}^{d \times N}$ is obtained by table looking up in the first layer, where d is the dimension of word embedding. The second layer is used to obtain multi-scale SUs to constitute the semantic

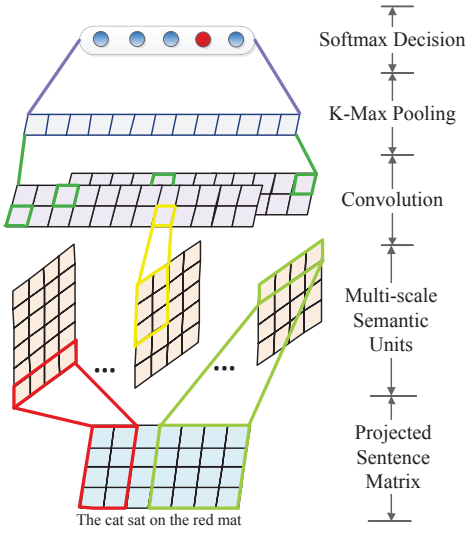

Figure 2: Architecture for short text modeling

matrices, which are combined and fed into convolutional layer, followed by k-max pooling operation. Finally, a softmax function is employed as classifier.

4.1 Detection for Multi-scale SUs

Methods for modeling short text S mainly have problem that its semantic meaning is determined by a few of key-phrases, however, these meaningful phrases may appear at any position of S. Thus, simply combining all words of S may introduce unnecessary divergence and hurt the overall semantic representation. Therefore, the detection for SUs are useful, which capture salient local information, as shown in Figure 2.

In particular, to obtain the representations of candidate SUs, multiple windows with variable width over word embeddings are used to perform element-wise additive composition, as follows:

$$[\mathbf{SU}_1, \mathbf{SU}_2, \cdots, \mathbf{SU}_{N-m+1}] = \mathbf{PM} \otimes \mathbf{E}_{win} \tag{5}$$

where, $E_{win} \in R^{d \times m}$ is a window matrix with all weights equal to one, and

$$\mathbf{SU}_i = \sum_{j=1}^{|\mathbf{PM}^{win,i}|} \mathbf{PM}_j^{win,i} \tag{6}$$

$\mathbf{PM}_j^{win,i}$ is the jth column from the sub-matrix $\mathbf{PM}^{win,i}$, which is windowed on projected matrix $\mathbf{PM}$ by E_{win} with the ith times sliding. m is the width of the window matrix E_{win}. With m varying, multi-scale contextual information can be exploited, which is helpful to reduce the impact of ambiguous words.

The meaningful SUs are assumed that they have one close neighbor at least in embedding space. Thus, we compute Euclidean distance between candidate SUs and semantic cliques. If the distance between candidate SUs and nearest word embeddings are smaller than the preset threshold, the candidate SUs are selected to constitute the semantic matrices, otherwise dropout.

4.2 Convolution Layer

In our network, the convolutional layer is used to extract local features. Kernel matrices $\mathbf{k}$ with certain width n are utilized to calculate convolution with the input matrices $\mathbf{M}$, as Equation (7).

$$C = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{d/2}]^{\mathrm{T}} = K^{\mathrm{T}} \otimes M \tag{7}$$

where,

$$K = [\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_{d/2}] \tag{8}$$

$$M = [\mathbf{M}_1^{win}, \mathbf{M}_2^{win}, \cdots, \mathbf{M}_{d/2}^{win}] \tag{9}$$

$$c_i^j = \mathbf{k}_i \cdot (\mathbf{M}_i^{win,j})^T \tag{10}$$

The c_i^j is generated from the jth n-gram in $\mathbf{M}$. Equation (7) produce the feature maps of convolutional layer.

4.3 K-Max Pooling

This operator is a non-linear sub-sampling function that returns the sub-sequence of K maximum values (LeCun *et al.*, 1998), which is used to capture the most relevant global features with fixed-length. Then, tangent transformation over the results of K-max pooling is performed, the output of which is concatenated to used as representation for the input short texts.

4.4 Network Training

The last layer is fully connected, where a softmax classifier is applied to predict the probability distribution over categories. The network is trained with the objective that minimizes the cross-entropy of the predicted distributions and the actual distributions (Turian *et al.*, 2010),

$$J(\theta) = -\frac{1}{t}\sum_{i=1}^{t} \log p(c^{\dagger}|\mathbf{x}_i, \theta) + \alpha\|\theta\|^2 \tag{11}$$

where t is number of training examples $\mathbf{x}$, and θ is the parameters set which comprises the kernels of weights used in convolutional layer and the connective weights from the fully connected layer.

Embedding	Senna[2]	GloVe[3]	Word2Vec[4]
Corpus	Wikipedia	Wikipedia	Google News
Dimension	50	50	300
$\|Vocab.\|$	130,000	400,000	3,000,000

Table 1: Details of word embeddings

Methods		Google Snippets	TREC
Semantic-CNN	Senna	83.6	96.4
	GloVe	84.4	**97.2**
	Word2Vec	**85.1**	95.6
DCNN (Kalchbrenner et al,2014)		–	93
SVMS (Silva et al., 2011)		–	95
CNN-TwoChannel (Kim, 2014)		–	93.6
LDA+MaxEnt (Phan et al., 2008)		82.7	–
Multi-Topics+MaxEnt (Chen et al., 2011)		84.17	–

Table 2: The classification accuracy of proposed method against other models

5 Experiments

5.1 Datasets

Experiments are conducted on two benchmarks: Google Snippets (Phan *et al.*, 2008) and TREC (Li and Roth, 2002).

Google Snippets This dataset consists of 10,060 training snippets and 2,280 test snippets from 8 categories. On average, each snippet has 18.07 words.

TREC The TREC questions dataset contains 6 different question types. The training dataset consists of 5,452 labeled questions whereas the test dataset consists of 500 questions.

5.2 Experimental Setup

Three pre-trained word embeddings for initializing the lookup table are summarized in Table 1. To discover semantic cliques, we take $\rho_{\min} = 16$ and $\delta_{\min} = 1.54$. Through our experiments, 6 kernel matrices in convolutional layer, $K = 3$ for max pooling, and mini-batch size of 100 are used.

5.3 Results and Discussions

5.3.1 Comparison with state-of-the-art methods

As shown in Table 2, we introduce 5 popular methods as baselines, and the details are described:

DCNN Kalchbrenner et al. (2014) proposed DCNN for sentence modeling with dynamic k-max pooling.

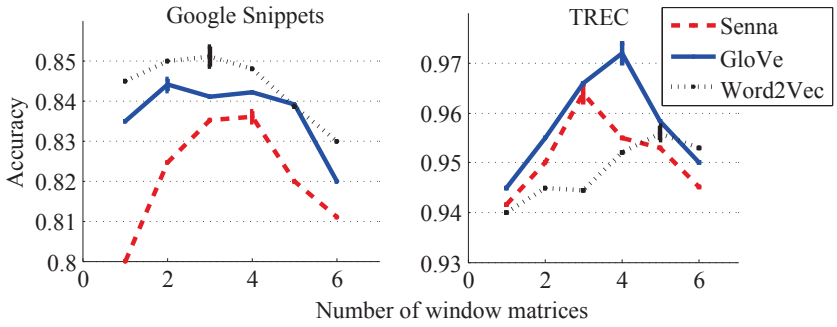

Figure 3: Number of windows for multi-scale SUs

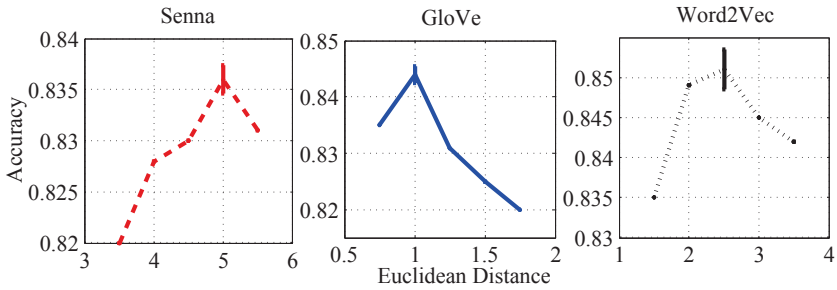

Figure 4: Influence of threshold in SUs detection

SVMs Parser, wh word, head word, POS, hypernyms, and 60 hand-coded rules were used as features to train SVMs (Silva *et al.*, 2011).

CNN-TwoChannel An improved CNN that allows task-specific and static word embeddings are used simultaneously (Kim, 2014).

LDA+MaxEnt LDA was used to discover hidden topics for expanding short texts (Phan *et al.*, 2008).

Multi-topics+MaxEnt Multiple granularity topics from LDA were utilized to model short texts (Chen *et al.*, 2011).

For valid comparisons, we respectively initialize the lookup table with the word embeddings in Table 1, and three experiments are conducted for each benchmark. As a whole, our method achieves the best performance, especially for TREC with 97.2% when the GloVe word embedding is employed. For Google snippets, our method achieves the highest result of 85.1% corresponding to the word embedding induced by Word2Vec.

5.3.2 Effect of Hyper-parameters

In Figure 2, for obtaining SUs with multi-scale, multiple window matrices with increasing width m are used. With respect to the variable m, the results are shown in Figure 3. We find small size of window may result in loss of critical information, however, the window with large size may introduce noise.

Figure 4 demonstrate how preset threshold d impact our method over benchmark Goggle snippets. We can draw a conclusion that when d is too small, only a few of SUs can be detected, whereas meaningless features are enrolled. The optimal threshold d can be chosen by cross-validation.

The impacts of other hyper-parameters like the number and size of the feature detectors in convolutional layer, and the variable k in k-max pooling layer are beyond the scope of this paper.

6 Conclusion

This paper proposes a novel semantic hierarchical model for short text classification. The model uses pre-trained word embeddings to introduce extra knowledge, and multi-scale SUs in short texts are detected.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61203281, No. 61303172, No. 61403385) and Hundred Talents Program of Chinese Academy of Sciences (No. Y3S4011D31).

[2]http://ml.nec-labs.com/senna/
[3]http://nlp.stanford.edu/projects/glove/
[4]https://code.google.com/p/word2vec/

References

Ainur Yessenalina and Claire Cardie. Compositional matrix-space models for sentiment analysis. In *EMNLP*, pages 172–182. Association for Computational Linguistics, 2011.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *SIGIR*, pages 841–842. ACM, 2010.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344, 2014.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429, 2010.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *ACL*, pages 384–394. Association for Computational Linguistics, 2010.

Mehran Sahami and Timothy D Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386. AcM, 2006.

Mengen Chen, Xiaoming Jin, and Dou Shen. Short text classification improved by learning multi-granularity topics. In *IJCAI*, pages 1776–1781. Citeseer, 2011.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, volume 1631, page 1642. Citeseer, 2013.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *WWW*, pages 1445–1456. International World Wide Web Conferences Steering Committee, 2013.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100. ACM, 2008.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

Joao Silva, Luísa Coheur, Ana Cristina Mendes, and Andreas Wichert. From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review*, 35(2):137–154, 2011.

Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*, 2014.