Societal Risk Classification of Post Based on Paragraph Vector and KNN Method

Jindong Chen, Xijin Tang
Institute of Systems Science, Academy of Mathematics and Systems Science
Chinese Academy of Sciences, Beijing, 100190 P.R. China

Keywords: Societal risk classification, Tianya Forum, Deep learning, Paragraph vector, KNN

Abstract. The posts of online forum reflect what are concerned by netizens, hence by labeling those online posts with relevant societal risk categories, the societal risks levels can be sensed within a period. Selecting Tianya Zatan board of Tianya Forum as the data source, applying socio psychology research results, risk classification of post is investigated. Due to the complexity of online corpus, using bag-of-words features to the paragraph of post, without consideration in semantics and the order of words, the performances of risk classification are unsatisfied. In this paper, with the combination of the semantics and the order of words, a deep learning model is applied to train post vector and realize the distributed representation of post. Based on the post vector, KNN method is used to tag the societal risk category of testing posts. Several experiments are implemented to choose appropriate post vector length and the parameters of KNN. The results show the effectiveness of post vector, but further investigation is required in the future.

Introduction

China is undergoing grand social transformation, the contradictions and conflicts among different classes become more intensified (Zheng & Tok, 2007). Accurately and timely sensing the types and intensities of societal risks is required by the Harmonious Society construction. The traditional questionnaire survey method is limited by the data size and time delay, which is unsuited to current societal risks monitoring (Robert & Dennis, 2009). In China, more and more people treat social media (such as blog, microblog, BBS, etc.) as one way to express their opinions toward the daily phenomena and social events openly and freely, hence it is a better choice to monitor the societal risk based on online data (Tang, 2013). “Tianya Zatan board is one of the most popular and influential board of Tianya Forum, which is a famous Internet forum in China, and provides BBS, blogs, micro-blogs and photo album services etc.” [1], and the posts of Tianya Zatan board cover the hot and sensitive topics of current society (Cao & Tang, 2014), it is a microcosm of society. The risk topics and frequency expressed on Tianya Zatan board can reflect the risk level of current society. Therefore, Tianya Zatan board of Tianya Forum is selected as data source for societal risk monitoring.

Through comprehensive analysis and comparison (Tang, 2013), the framework of societal risk indicators including 7 categories and 30 sub categories based on word association tests which is constructed by Zheng et al. (2009) is chosen as risk categories. As can be found, the contents of the posts of Tianya Forum are mainly textual information; only a minority of posts is attached with pictures or other media information, then text classification is the first choice to classify the posts of Tianya Zatan board (Chen & Tang, 2014). However, the risk classification of posts has several specific features from standard text classification, such as the dynamical variation of context, the limitation of training samples, the bad quality of corpuses, etc., which make risk classification of posts more difficult than standard text classification discussed intensively by professionals. Through similarity analysis of posts in same risk category and different risk categories, it is shown that risk classification of posts is feasible but difficult (Chen & Tang, 2014). Hence, how to improve the risk classification accuracy of posts is the main issue addressed here.

The basic principle of text classification is utilizing learning strategies to assign predefined categories labels to new documents based on the likelihood suggested by a trained set of labels and documents (Zhang, et al., 2008). Generally, two main procedures affect the accuracy of text classification: document representation and classifier construction. The traditional document representation method is called as one-hot representation: the document vector size equals the vocabulary size, the element at the word index is “1” while the other elements are “0”s (Bengio, et al., 2003). One-hot representation is mainly through feature word extraction and feature word selection to improve the quality of document vector (Zhang, et al., 2011). Based on one-hot representation, many research works have been done on classifier construction and shown their effectiveness in text

[1] http://en.wikipedia.org/wiki/Tianya_Club

classification field (Hu & Tang, 2013; Zhang, et al., 2008). The strategies which can be divided into three classes: supervised, unsupervised and semi-supervised, while supervised methods have shown advantages in text classification. The representative supervised machine learning method for text classification is support vector machine (SVM). However, due to the limitations of one-hot representation (such as the curse of dimensionality, no syntactic or semantic information and losing the order of words) and the difficulty in risk classification of posts, the performance of risk classification based on one-hot representation and SVM is unsatisfied, even though we increased the training set to ten thousands of posts and improved the feature word selection method (Chen & Tang, 2014).

To address these issues of one-hot representation, Bengio et al. (2003) proposed the distributed representation of words, which has garnered significant attention in the recent past. Instead of a one-hot vector, a word is represented by a real-valued vector with a much smaller size. The distributed representation is without the curse of dimensionality problem. Moreover, the syntactic and semantic information of words can be encoded in the distributed vector space. However, the computational complexity of this model is originally too high for real world tasks. With Recurrent Neural Network (RNN), Mikolov (2012) developed a more efficient deep learning language model. To face the real world task, Mikolov et al. (2013a) improved the deep learning language model and carried out two neural network models for representation learning: CBOW and Skip-gram, which is also called as Word2Vec. The distributed representation of word is just the beginning, from the word to sentence and paragraph is our attention. But simply adding the word vector together will lose the order information of words in the paragraph, which is unreasonable. Le et al. (2014) proposed a paragraph vector method based on Word2Vec model, which realized the distributed representation of paragraph and the encoding of order information, and show state of the art performance in sentiment classification and information index. Therefore, to improve the performance of risk classification of posts, the paragraph vector method is another choice.

Following the method of Le et al. (2014), we focus on realizing the distributed representation of post paragraph, and testing the paragraph vector in risk classification of posts in this study. The rest of this paper is organized as follows. Section 2 presents the algorithm of paragraph vector. The experiment steps and data set is explained in section 3. The results and discussions are presented in Section 4. Finally, conclusion and further research plan are given in Section 5.

Algorithm of Paragraph Vector

Before discussing the algorithm of paragraph vector, the previous works of learning word vector are presented first, which is the inspiration of the algorithm of paragraph vector.

Distributed Representation of words. This part introduces the concept of distributed representation of words. A framework for learning the word vectors is presented in Fig.1 (Mikolov, 2012). The task is to predict next word through given words in the context. As shown in Fig.1, context of three words (“the,” “cat,” and “sat”) is used to predict the fourth word (“on”).

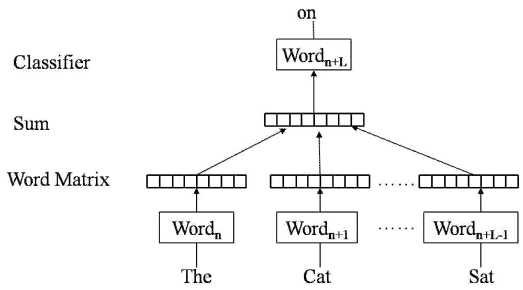

Fig.1 A framework for learning word vectors

In this framework, every word is mapped to a unique real-valued vector, represented by a column in a matrix $\boldsymbol{W}$. The column is indexed by position of the word in the vocabulary. The sum of the vectors is then used as features for prediction of the next word. In practice, to predict one word in the sentence, the input features not only consider the words before, but also take the words after into consideration.

More formally, given a sequence of training words w_1, w_2, w_3,…w_T, T is the number of words, taking k words before or after w_t into consideration, the objective of the word vector model is to maximize the average log probability,

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t \mid w_{t-k},\ldots,w_{t+k}) \quad (1)$$

The prediction task is typically done via a multiclass classifier, such as softmax, as shown in Eq.2,

$$p(w_t \mid w_{t-k},\ldots,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \quad (2)$$

Through Eq.3, the un-normalized log-probability for each output word *i* is computed

$$y = b + Uh(w_{t-k},\ldots,w_{t+k};W) \quad (3)$$

where U and b are the softmax parameters, h is constructed by the concatenation of word vectors extracted from matrix $\boldsymbol{W}$.

In practice, hierarchical softmax (Mikolov et al., 2013b) is preferred to softmax for fast training. In Word2Vec, the structure of the hierarical softmax is a binary Huffman tree, where short codes are assigned to frequent words. This is a good speedup trick because common words are accessed quickly.

Many research works have validated the effectiveness of Word2Vec. We also do several tests with Word2Vec: using the new post data set of Tianya Zatan board (1.3G) to train Word2Vec model, the word frequency is bigger than 5, the length of word vector is 200. After the training converges, words with similar meaning are mapped to a similar position in the vector space. For example, “拆迁(Demolition)”and” 征地(Land Requisition)” are close to each other, the cosine similarity is 0.859; “抢劫(Rob)”and “盗窃(Steal)” are close to each other, the cosine similarity is 0.787. The difference between word vectors also carries meaning. For example, the word vectors can be used to answer analogy questions using simple vector algebra: “中国(China)”-“北京(Beijing)”+“陕西(Shaanxi)”=“西安(Xian)”, “习近平(Xi Jinping)”-“主席(Chairman)”+ “普京(Putin)”=“总统(President)” , which means the distance between “中国(China)” and “北京(Beijing)” is similar to the distance between “陕西(Shaanxi)” and “西安(Xian)”, the distance between “习近平(Xi Jinping)” and “主席(Chairman)” is similar to the distance between “普京(Putin)” and “总统(President)”. From these points, Word2Vec also shows good performance on constructing the Chinese word vector.

Distributed Representation of Paragraph. In practice, the word representation is just the beginning of text representation, the distributed representations of sentence and paragraph attract more attention. Simply using the sum of word vectors to represent the paragraph vector is unreasonable, because the order information of words and semantic of paragraph will lose.

Le et al. (2014) proposed an approach for learning paragraph vectors which is inspired by the methods for learning the word vectors. The inspiration is that the word vectors are asked to contribute to a prediction task about the next word in the sentence. So despite the fact that the word vectors are initialized randomly, they can eventually capture semantics as an indirect result of the prediction task. Based on this idea, the paragraph vectors are realized in a similar way. The paragraph vectors are also asked to contribute to the prediction task of the next word given many contexts sampled from the paragraph. Two models are proposed by Le et al. (2014): PV-DM and PV-DBOW. As the tests of the two models shown the influence of PV-DBOW is unobvious, so this model is not considered here. Following the PV-DM method of Le et al. (2014), we carry out learning post vector in a similar way.

In Post Vector framework (Fig. 2), the post vector (Post ID) is treated as another word, and is combined with other words to train the model. Hence, every post is mapped to a unique vector, represented by a column in matrix $\boldsymbol{D}$ and every word is also mapped to a unique vector, represented by a column in matrix $\boldsymbol{W}$. The post vector and word vectors are concatenated to predict the next word in a context. More formally, the only change in this model compared to the word vector framework is in Eq.3, where h is constructed from W and D.

In the training process, the contexts are fixed-length and sampled from a sliding window over the paragraph. The post vector is shared across all contexts generated from the same post but not across posts. Hence, the post vector acts as a memory that remembers what is missing from the current context – or the topic of the paragraph. The word vector matrix $\boldsymbol{W}$ is shared across posts. i.e., the vector for “拆迁(Demolition)” is the same for all posts.

The post vectors and word vectors are trained using stochastic gradient descent and the gradient is obtained via back propagation. At every step of stochastic gradient descent, one can sample a fixed-length context from a random post, compute the error gradient from the network in Fig.2 and use the gradient to adjust the parameters in the model.

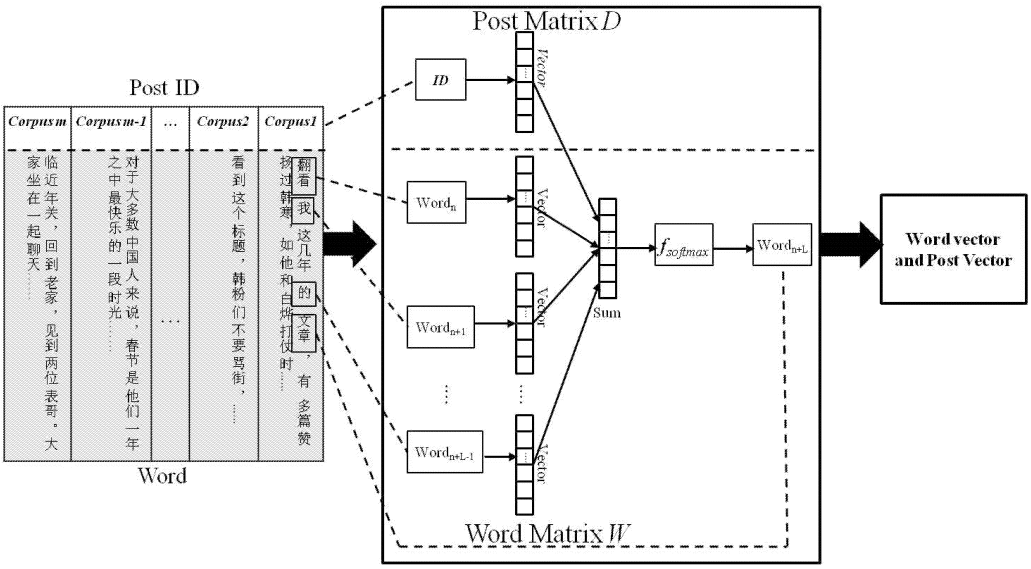

Fig.2 A framework for learning word vectors and post vectors

In prediction, an inference step is carried out to compute the post vector for a new post based on the trained model. This is also obtained by gradient descent. The difference is the parameters for the rest of the model (the word vectors ***W*** and the softmax weights) are fixed.

Suppose that there are M posts in the data set, N words in the vocabulary, and post vectors such that each post is mapped to p dimensions and each word is mapped to q dimensions, then the model has the total of $M \times p + N \times q$ parameters (excluding the softmax parameters). Even though the number of parameters can be large when M is large, the updates during training are typically sparse and thus efficient.

After being trained, the post vectors can be used as features for the post (e.g., in lieu of or in addition to bag-of-words). These features can be directly fed to conventional machine learning methods such as logistic regression, support vector machines, or KNN.

Data Set and Experiment Steps

With Tianya Forum spider system of our group (Zhao & Tang, 2013), the daily new posts and updated posts are downloaded and parsed. To train post vector model, the new posts of Dec. 2011-Mar. 2013, more than 48 thousands posts, are used. To test the effectiveness of post vector in risk classification, Dec. 2011-Mar. 2012 four months labeled posts are used.

To tag the risk category of testing post, KNN classification method is used, which means the risk category of post is decided by the risk categories of the nearest k posts. The experiment steps are described as:

1) The corpuses are segmented with Ansj_Seg tool which is a JAVA package based on inner kernel of ICTCLAS[2].
2) Each post is assigned with a unique ID.
3) All the corpuses are fed into post vector model for training, and then the post vector is constructed for each post.
4) Based on cosine similarity between post vectors, KNN method is applied to label the risk category of testing posts.

Results and Discussions

[2] https://github.com/ansjsun/ansj_seg

Several experiments are carried out to verify the effectiveness of post vector. The computer is Dell Optiplex 9020, CPU is 8*3.4GHz, the memory is 8G, and all the experiments are run on Ubuntu 12.02.

Choose Proper Post Vector Length. To test the influence of post vector length and choose appropriate value for vector length, three post vector lengths are tested: 50, 100, and 200.

The training times of these three models are 20 minutes, 26minutes and 31minutes. The *k* value of KNN method is set as 1, which means the risk category of the maximum similar post to tag the testing post. Comparing the results of machine learning with annotations by human beings, the ratios of both are consistent with each other are presented in Fig.3. Fig.3a is the results of machine learning only based on all posts of the previous month; Fig.3b is the results of machine learning based on all posts of Dec. 2011 to the previous month.

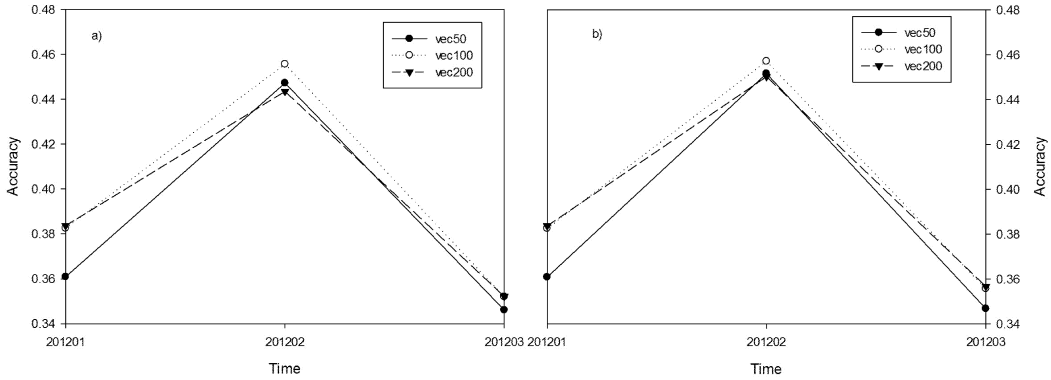

Fig.3 The testing results of different post vector length

From the results of Fig.3, it can be found that when the post vector length equals to 100, the best performance is obtained for the majority of experiments, hence the post vector length is set as 100.

Choose Proper *k* Value for KNN Method. Choose proper *k* value for KNN method, several experiments are tested: *k*=1, 10, 20 and 40. The major risk class is used to tag the testing post. The ratios that machine learning equals to human annotated are computed, all the results are presented in Fig.4. Fig.4a is the results of machine learning using the posts of the previous month, Fig.4b is the results of machine learning using the posts of Dec. 2011 to the previous month.

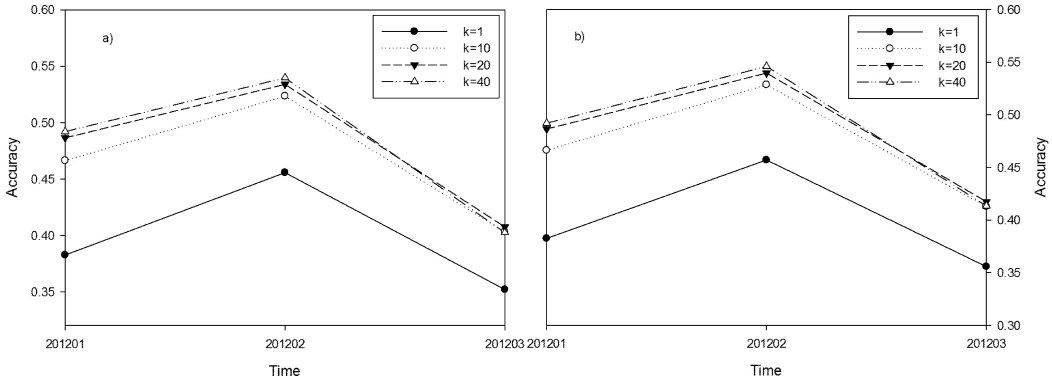

Fig.4 The testing results of different *k* value of KNN

From the results of Fig.4, comparing with *k*=1, when *k*=10, there is obvious improvement of consistency. However, when *k*=20, 40, the increase is unobvious, even decrease in several tests, which means simply increase the *k* value may not improve the performance of machine learning.

Choose Proper Threshold of Cosine Similarity. To improve the performance of machine learning, a threshold of cosine similarity is set, which means not only increase the *k* values, but also take the similar degree of the posts into consideration. The thresholds are set as 0.5, 0.4 and 0.3, but the maximum *k* is 10. All the results are presented in Fig.5. Fig.5-a is the results of machine learning using the posts of the previous month, Fig.5-b is the results of machine learning using the posts of Dec. 2011 to the previous month.

From the results of Fig.5, the performances of thresholds 0.3 and 0.4 are similar, both are much better than threshold value 0.5. Hence, the threshold value for Cosine similarity will be selected between 0.3-0.4.

The Effectiveness Test of Post Vector. After all the tests, set the post vector length as 100, threshold=0.3, the maximum *k* is 100, another experiment is carried out. The results are presented in Fig.6. The line "1M1M" means the results of machine learning using the posts of the previous month, the line "nM1M" means the results of machine learning using the posts of 2011.12 to the previous month.

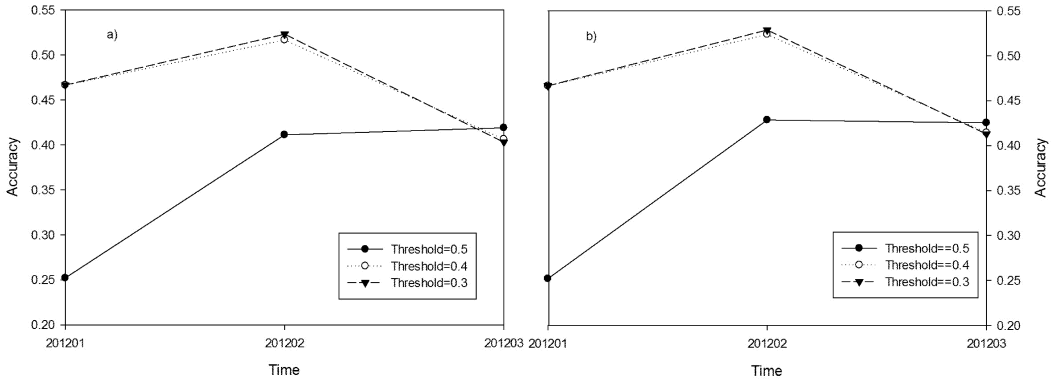

Fig.5 The testing results of different thresholds of cosine similarity

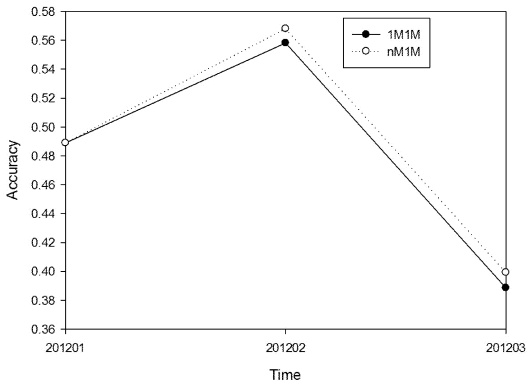

Fig.6 The performance of post vector length=100, threshold=0.3, max_*k*=100

From the results of Fig.6, the average ratios of consistency in three months are calculated: 1M1M is 0.4786, and nM1M is 0.4854. Comparing with the results of Fig.3, Fig.4 and Fig.5, the line "nM1M" obtains the highest ratio that machine learning equals to human annotated.

The KNN classification based on one-hot representation method is tested in this part, but due to the big time cost in the experiment, only the classification results of Jan. 2012 based on the posts of Dec. 2011 are calculated. Using the same parameters of KNN, the ratio that machine learning and human annotation are same based on one-hot representation method is 0.3999, but the ration of post vector is 0.4889, which is almost 10 percent better than one-hot representation.

Conclusions

Following the method of paragraph vector, we use the post vector model and KNN method to tag the risk category of the testing post. Several results are obtained in this study:

1) With the distributed representation of post, a KNN classification method based on post vector is proposed;
2) Through experiment tests, the influence of the length of post vector, *k* value of KNN and the threshold of cosine similarity are tested;
3) The effectiveness of the distributed representation method of post is validated in this study.

Acknowledgment

This research is supported by National Basic Research Program of China under Grant No. 2010CB731405 and National Natural Science Foundation of China under Grant No.71171187 and No. 71371107. The authors would

like to thank Mr. Yongliang Zhao and Mr. Zedai Zhang for their data collection work, the people who take part in labeling work and the other members of our team.

References

[1] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003), A neural probabilistic language model. Journal of Machine Learning Research 3:1137–1155.

[2] Cao, L.N., & Tang, X.J.(2014), Topics and Threads of the Online Public Concerns Based on Tianya Forum[J]. Journal of Systems Science and Systems Engineering, 23(2): 212-230.

[3] Chen, J.D., & Tang, X.J.(2014). Exploring Societal Risk Classification of the Posts of Tianya Club. International Journal of Knowledge and Systems Science, 5(1): 36-48.

[4] Hu, Y., & Tang, X.J. (2013), Using support vector machine for classification of Baidu hot word. *In Knowledge Science, Engineering and Management* (KSEM2013, Dalian, China. M. Wang, et al eds.). Lecture Notes in Computer Science, 8041, Springer Berlin Heidelberg, 580-590.

[5] Le, Q., &Mikolov, T. (2014), Distributed Representations of Sentences and Documents[A]. Proceedings of the 31st International Conference on Machine Learning, Beijing, China. JMLR:W&CP volume 32.

[6] Mikolov, T. (2012), Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology.

[7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a), Efficient Estimation of Word Representations in Vector Space[C]. International Conference on Learning Representations (ICLR) 2013. Scottsdale, Arizona, US:1-12.

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013b), Distributed representations ofphrases and their compositionality. In *Advances on Neural Information Processing Systems*.

[9] Robert, M.K., & Dennis, P.S. (2009), Psychological testing: Principles, applications, and issues. CA: Wadsworth.

[10] Tang, X.J. (2013), Exploring On-line Societal Risk Perception for Harmonious Society Measurement[J]. Journal of Systems Science and Systems Engineering, , 22(4): 469-486.

[11] Zhang, H.P., Yu, H., & Xiong, D. (2003), HHMM-based Chinese lexical analyzer ICTCLAS[A]. Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17: Association for Computational Linguistics[C], 184-187.

[12] Zhang, W., Yoshida, T., & Tang, X.J. (2008), Text classification based on multi-word with support vector machine. Knowledge-Based Systems, 21(8), 879-886.

[13] Zhang, W., Yoshida, T., & Tang, X.J. (2011), A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Systems with Applications, 38(3), 2758-2765.

[14] Zhao, Y.L., & Tang, X.J. (2013), A Preliminary Research of Pattern of Users' Behavior Based on Tianya Forum. *The 14th International Symposium on Knowledge and Systems sciences.* (Ningbo, P.R.China., Oct. 25-27, 2013). JAIST Press, 139-145.

[15] Zheng, R., Shi, K., & Li, S. (2009), The influence factors and mechanism of societal risk perception. Proceedings of the First International Conference on Complex Sciences: Theory and Application (Shanghai, China, J. Zhou eds.). Springer Berlin Heidelberg, 2266-2275.

[16] Zheng, Y. & Tok, S.K. (2007), "Harmonious Society" and "Harmonious World": China's Policy Discourse under Hu Jintao. Briefing Series, Issue 26, China Policy Institute, The University of Nottingham, UK.