Feature Weighting Improvement of Web Text Categorization Based on Particle Swarm Optimization Algorithm

Yonghe Lu*, Yanhong Peng
Sun Yat-sen University, Guangzhou, China.

* Corresponding author. Tel.: 15360606436; email: luyonghe@mail.sysu.edu.cn

Manuscript submitted February 15, 2015; accepted May 3, 2015.
doi: 10.17706/jcp.10.4.260-267

Abstract: It is usually true that some structures like title can express the main content of texts, and these structures may have an influence on the effectiveness of text categorization. However, the most common feature weighting algorithms, called term frequency-inverse document frequency (TF-IDF) doesn't think about the structural information of texts. To solve this problem, a new feature weighting algorithm based on Particle Swarm Optimization algorithm is put forward. It considers the structure information (i.e., HTML tags) of web pages. Firstly, web pages are crawled and pre-processed, at the same time, the content of four HTML tags is reserved; secondly, Chi-squared (CHI) is used to select features; thirdly, a new feature weighting algorithm, which is called the feature tag weighting algorithm, is come up with. In the feature tag weighting algorithm, we use particle swarm optimization (PSO) to calculate tag weighting coefficients; lastly, k-nearestneighbor (kNN) is used as the web text categorization. The experiment results show that feature tag weighting algorithm has better performance than TF-IDF in the effectiveness of web text categorization.

Key words: Text categorization, TF-IDF, PSO, web text, HTML tag.

1. Introduction

Text categorization is a process of dividing texts into one or multi classes. As the development of the web technology, the object of pure-text categorization has extended to web texts. Compared to the pure-texts, web texts have obvious identifier (i.e., HTML tags) to express its structural information. Usually, in the text extracting step, HTML tags are removed and extract plaintext from each web page [1]. However, there is much useful information about the content organization based on HTML tags. Many researches show that the structural information, especially HTML tags, like hyperlink, table layout and so on, can be used to improve the effectiveness of web text categorization [2]-[6].

Feature weighting is used to measure the importance or separation of the terms in the text representation. Different feature weighting algorithms will have different influence on categorization results [7]. At present, the most common feature weighting algorithms in vector space model is Term Frequency-Inverse Document Frequency (shorted for TF-IDF) [8], which is showed in (1).

$$W(t,d)=\frac{tf(t,d)\times\log(N/n_t+0.01)}{\sqrt{\sum_{t\in d}\left[tf(t,d)\times\log(N/n_t+0.01)\right]^2}} \tag{1}$$

where $W(t,d)$ is the weighting of the term t in document d; $tf(t,d)$ is the frequency of the word t in document *d*; *N* is the total number of documents; n_t is the number of documents which include term *t*.

But TF-IDF doesn't think about the structure of documents. In reality, the same term in different places of a text may have different contributions to the text. In the web texts, as HTML tags express the structural information, so we can use HTML tags to improve TF-IDF. Some researchers introduced HTML tags into TF-IDF to improve feature weighting. Wang Changlong etc introduced a HTML tag set of web texts, which is S={TITLE, $H1$, $H2$, $H3$, $H4$, $H5$, $H6$, B, U, I}. The terms in different HTML tags are given different weighting coefficients [9]. Mingyu Lu etc. used HTML tags of the web texts to adjust the feature vector spaces [10]. Hui Liu etc. introduced a HTML tag factor (i.e., pti) and proposed a feature weighting algorithm called TF-IWF [11].

But the weighting coefficients of HTML tags are not scientific, because they just depend on given values personally. Particle Swarm Optimization (shorted for PSO) is an optimization calculation based on swarm intelligence. It is originally attributed to Kennedy etc. and was first intended for simulating social behavior in 1995 [12]. PSO can be used to improve the performance of web text categorization. Ziqiang Wang etc. used PSO as a classifier of the web text categorization. The values of F1 and ROC are both higher than Decision Tree, Naïve Bayes and KNN [13]. Song Liangtu etc. combined PSO and SVM and came up with a new classifier, namely PSORTP. Compared to SVM, the velocity of web text categorization is greatly improved [14]. Chaoxia Tang used PSO to improve KNN, not only the time of the web text categorization reduced, but also the effect improved [15]. Yonghe Lu proposed a text feature selection method based on PSO (PSOTFS) to mine the text feature selection rules. The experiment results show that it is better than that of Chi-squared, information gain, document frequency and mutual information [16]. At the same time, PSO can be used to optimize parameters, like weighting coefficients. Qi Liu etc. provided an algorithm to pick-up the body of the web pages. It used PSO to optimize the characteristic weighting [17]. By using PSO, Long Long etc. found a public parameter. It is used to calculate supporting probabilities of each subject area's characteristic attributes [18]. Based on PSO, more scientific and reasonable parameters can be gotten.

In this paper, a new feature weighting algorithm, which is called the feature tag weighting algorithm, is come up with. In the feature tag weighting algorithm, we use Particle Swarm Optimization to calculate tag weighting coefficients.

2. Feature Tag Weighting Algorithm

This task starts from a list of HTML tags, retrieving the text according to each HTML tag. The HTML tags considered are currently <title>, <h1> to <h6>, <a>, <strong>, <b>, <i>, <image> and so on. Whenever one of these tags is found, a context phrase is recorded. For example, the title is within a pair < title > </title>, the strong phrase within a pair <strong> </strong>. We consider k kinds of HTML tags here, they are provided in the following:

0 is the content of tag0;

1 is the content of tag1;

2 is the content of tag2;

…

k is the content of tagk.

P={0,1,2,⋯,k} is the set of the above figures ($k \in P$). When calculating the feature weighting of the term in tag *i*, a tag weighting coefficient a_i ($i \in P$) should be multiplied by. The tag frequency of each item *t* is calculated in (2).

$$tf_t(t,d) = \sum_{i \in P} [\mathrm{tf}_t(t_i,d) \times \frac{a_i}{\sqrt{\sum_{j=0}^{k} a_i^{2}}}] \tag{2}$$

where $tf_t(t,d)$ is the tag frequency of term *t* in document d; $tf_t(t,d)$ is the tag frequency of term *t* in tag *i*. Based on TF-IDF which is calculated in (1), the feature tag weighting is calculated in (3).

$$W_t(t,d) = \frac{tf_t(t,d) \times \log(N/n_t + 0.01)}{\sqrt{\sum_{t \in d} [tf_t(t,d) \times \log(N/n_t + 0.01)]^2}} \tag{3}$$

where $W_t(t,d)$ is the feature tag weighting of term *t* in document *d*; *N* is the total number of documents; n_t is the number of documents which contain term *t*. Feature tag weighting algorithm takes into account the influence of HTML tags on the feature weighting in web texts.

3. Particle Swarm Optimization

In Particle Swarm Optimization, each particle represents a possible solution to the optimization task. During each time of iteration, each particle accelerates in the direction of its own personal best solution found so far, as well as in the direction of the global best position discovered so far by any of the particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move to it, and explore the region more thoroughly in the process.

The position vector and the velocity vector of particle *i* in the D-dimensional search space can be represented as $X_i=(x_{i1}, x_{i2},..., x_{iD})$ and $V_i=(v_{i1}, v_{i2},..., v_{iD})$ respectively. According to the objective function, let the best position of each particle be $P_i=(p_{i1}, p_{i2},..., p_{iD})$ and the best position of all the particles be $G=(g_1, g_2, ...,g_D)$ among all P_i. The particles are used to adjust its position and velocity in (4) and (5).

$$V_i = V_i + c_1 \times rand() \times (P_i - X_i) + c_2 \times Rand() \times (G - X_i) \tag{4}$$

$$X_{i+1} = X_i + V_{i+1} \tag{5}$$

where c_1 and c_2 are the acceleration coefficients, and rand() and Rand() are random numbers uniformly distributed within [0,1]. Typically, c_1 and c_2 are both set to a value of 2.0. The value of each dimension of every velocity Vi can be clamped to the range [-Vmax, Vmax] to reduce the likelihood of particles leaving the search space.

As there are some unknown tag weighting coefficients in (2), PSO could be used to find them. It helps to get the more scientific parameters.

4. Algorithm Description

The steps of feature tag weighting algorithm of web text categorization based on Particle Swarm Optimization are described as follows:

Step 1: Progressively scan each character of training set and testing set, the content of tag k should be identified and extracted without those HTML tags which are worthless for web text categorization, like script, advertisement links, navigation bar, annotation and so on.

Step 2: Stop words are some of the most common short function words, such as the, is, at, which, and so on. They have no meaning to the text categorization. Removing them could reduce the interference of the feature selection.

Step 3: Using Chi-squared (CHI) which is one of the classical methods of feature selection to select features, *n* feature terms will be gotten in CHI descending order.

Step 4: A tag weighting coefficient is one dimension of a particle's position vector. The dimension of particles' position in the algorithm is same to the kinds of tags.

Step 5: A set of individuals (i.e., particle) is created at random. Individual i's position at iteration 0 can be represented as the vector X_i^0=(x_{i1}^0, x_{i2}^0,..., x_{ik}^0), where k is the number of tag weighting coefficients. The velocity of individual i (i.e.,V_i^0=(v_{i1}^0, v_{i2}^0,..., v_{ik}^0)) which corresponds to the tag weighting coefficient update quantity covering all tag weighting coefficient values, the velocity of each individual is also created at random. The elements of position and velocity have the same dimension.

Step 6: Similar to all evolutionary computation techniques there must be some function or method to evaluate the goodness of a position. The fitness function must take the position in the solution space and return a single number representing the value of that position. The evaluation function of PSO algorithm provides the interface between the physical problem and the optimization algorithm. In consideration of the effectiveness of web text categorization, evaluation metrics can be used as the fitness function. There are some evaluation metrics: precision, recall, F1, macro-averaged score and micro-averaged score. But F1, macro-averaged score and micro-averaged score are relatively complex in calculation, the fitness function can be defined as the precision [16], which is showed in (6).

$$fitness() = \frac{the\ number\ of\ the\ correct\ categorization\ texts}{the\ total\ number\ of\ texts} \tag{6}$$

Step 7: K-NearestNeighbor (kNN) is used to classify training set and testing set, and get the precision which is the value of fitness().

Step 8: Each particle i memorizes its own fitness()'s value and chooses the maximum one, which has been better so far as personal best position P_i^t. The particle with the best fitness()'s value among P_i^t is denoted as global position G, where t is the iteration number.

Step 9: Modifying the velocity of each particle according to (4). If $V_i^{(t+1)} > V_i^{\max}$, then $V_i^{(t+1)} = V_i^{\max}$; If $V_i^{(t+1)} < V_i^{\min}$, then $V_i^{(t+1)} = V_i^{\min}$.

Step 10: Modifying the position of each particle according to equation (5).

Step 11: If the best evaluation value G is not obviously improved or the iteration number t reaches the given maximum, then go to Step 12. Otherwise, go to Step 6.

Step 12: The particle generates the best evaluation value *G* (i.e., the precision of the web text categorization) and the tag weighting coefficient (i.e., α_i) according to each tag (i.e., *i*).

The process of feature tag weighting algorithm of web text categorization based on Particle Swarm Optimization is shown in Fig. 1.

Table 1. The Source URLs of the Dataset

Class	URLs	Number of Webpage
finance	http://finance.sina.com.cn/	500
sports	http://sports.sina.com.cn/	500
games	http://games.sina.com.cn/⍰	500
entertainment	http://ent.sina.com.cn/⍰	500
constellation	http://astro.sina.com.cn/⍰	500
job	http://www.zhaopin.com/,http://58.com/job/⍰	500

5. Experiment Process and Results

As there is not a widely used tagged web corpus in the world, 7 URLs (Table 1) are randomly selected to crawl the data in our paper. The data crawled are divided into 6 classes. The total number of data was 3000, they are divided into a training set (1500) and a testing set (1500) at random. As four HTML tags which include <title></title>, <strong></strong>, <span></span> and <p></p> are the most common HTML tags in the web texts of these 7 URLs. In this paper, they are chosen for experiment.

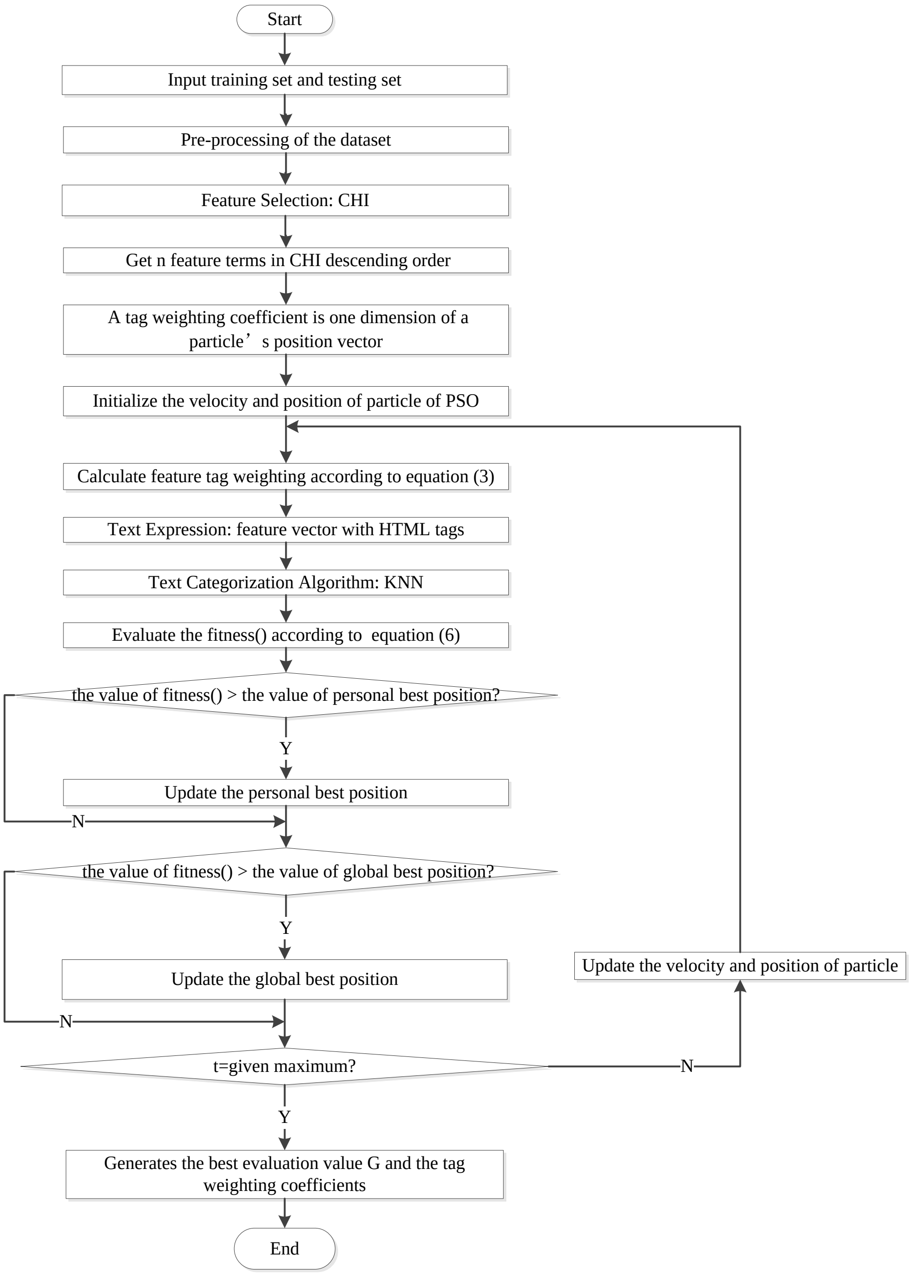
Start
Input training set and testing set
Pre-processing of the dataset
Feature Selection: CHI
Get n feature terms in CHI descending order
A tag weighting coefficient is one dimension of a particle’ s position vector
Initialize the velocity and position of particle of PSO
Calculate feature tag weighting according to equation (3)
Text Expression: feature vector with HTML tags
Text Categorization Algorithm: KNN
Evaluate the fitness() according to equation (6)
the value of fitness() > the value of personal best position?
Y
Update the personal best position
N
the value of fitness() > the value of global best position?
Y
Update the global best position
N
Update the velocity and position of particle
t=given maximum?
N
Y
Generates the best evaluation value G and the tag weighting coefficients
End

Fig. 1. The process of feature tag weighting algorithm of web text categorization based on PSO.

The experimental environment is Windows 7, 2GB, Java and eclipse-standard. The maximum iteration is set as 50, c_1 and c_2 are both 2, and w is 0.8. In KNN, k is set as 100. The experiment results based on TF-IDF and feature tag weighting are showed in Table 2 and Fig. 2.

Table 2. Precisions of Different Feature Weighting Algorithms

Feature Dimension	50	100	200	300	400	500	600
TF-IDF	88.33%	90.47%	94.47%	96.33%	93.87%	90.87%	89.47%
Feature Tag Weighting	92.80%	94.47%	95.80%	96.80%	95.53%	93.93%	93.13%

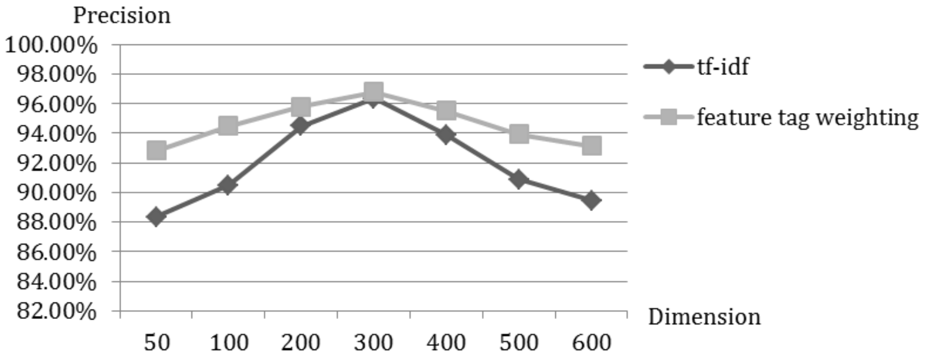

Fig. 2. Precisions of different feature weighting algorithms.

By Fig. 2 and Table 2, the precisions of feature tag weighting are all higher than TF-IDF, namely feature tag weighting algorithm have a better effectiveness on web text categorization than TF-IDF. The feature dimensions can't influence feature tag weighting algorithm too much, it ranges within [92.80%, 96.80%]. However, according to TF-IDF, after dimension 300, with the increase of dimensions, the precisions of TF-IDF decrease much more than feature tag weighting algorithm. The structural information of HTML tags can be considered in the feature weighting of web text categorization.

In this paper, <title></title>, <strong></strong>, <span></span> and <p></p> are chosen for experiment. So there are tag0 (i.e., <p></p>), tag1 (i.e., <title></title>), tag2 (i.e., <strong></strong>), tag3 (i.e., <span></span>). Therefore, *P*={0,1,2,3} and there are four tag weighting coefficients (i.e., α_0, α_1, α_2 and α_3). Using PSO to get the tag weighting coefficients, the experiment results are showed in Table 3.

Table 3. Tag Weighting Coefficients of Feature Tag Weighting Algorithm

Feature Dimension	50	100	200	300	400	500	600
α_0	0.473	0.150	1.000	1.000	1.000	1.000	1.000
α_1	1.000	0.422	0.852	1.000	1.000	0.582	0.402
α_2	1.000	0.060	0.001	0.781	0.841	1.000	1.000
α_3	0.561	0.288	0.694	0.968	0.774	0.647	0.999

By Table 3, we can know that even the numbers of the feature dimensions are different, some tag weighting coefficients still have regularity.

α_0 corresponds to tag0 (i.e., <p></p>). In most feature dimensions, the value of α_0 is 1. <p></p> is the tag that means paragraph. This element encloses text in a paragraph. As more words are in the paragraphs relative to other three tags, maybe as the dimension of feature selection rising, more features in the paragraphs are chose by CHI.

α_1 corresponds to tag1 (i.e., <title></title>). In three feature dimensions, the value of α_1 is 1. <title></title> is the tag that means document title. This element encloses the title of a document. It is the same as the common thought, which is that title has a higher expression to the subject of the texts than other structure information. The results show that <title></title> has a certain influence on Web text categorization.

α_2 corresponds to tag2 (i.e., <strong></strong>). In three feature dimensions, the values of α_2 is 1. <strong></strong> is the tag that means strong emphasis. This element brackets text which should be strongly emphasized. It shows that <strong></strong> still has some influence to Web text categorization. Thinking about the reason, in a webpage, there may be little terms marked as a strong terms, which may result in their little appearance in the terms of feature selection.

α_3 corresponds to tag3 (i.e., <span></span>). The values of α_3 doesn't have obvious regularity. <span></span> is the tag that means generic language or style container. The SPAN element allows authors to add structure, like images, hyperlinks, div and so on, to documents in a flow of text inline. It is a part of the content of the text body. Maybe that it is the reason <span></span> doesn't have obvious regularity.

Except our speculation, the total number of experiments is not so much, that is, every dimension had litter experiments, maybe some regular tag weighting coefficients could not be found.

6. Conclusion

Pure-text categorization has not think about the change of the web text, which has HTML tags to carry more structure information of the text. If we just use TF-IDF which suits to the pure-text categorization, the characteristic of web texts may be ignored. The feature tag weighting algorithm takes into account the influence of HTML tags (i.e., the structural information of web texts) on the web text categorization. It performs better than TF-IDF in the effect of tagged web text categorization. According to our experiment result, feature tag weighting algorithm gets the highest precisions (i.e., 96.80%) and it may not be influenced by the dimensions of feature selection so much compared to TF-IDF. The feature tag weighting algorithm can not only get the best categorization results, but also performed well in many dimensions. At the same time, our experiment reminds us that when classifying the web texts, we can consider HTML tags in order to improve the effect of web text categorization.

However, there are still many deficiencies in this paper. Just four HTML tags are chosen in our paper; the influence of HTML tags on the recall and F-Measure; the experiment was tested on a specific data set; whether the number of HTML tags will influence the results; the values of each tag weighting coefficients. All problems above need much more research. In our later research, we will consider whether the amount of the HTML tags could influence the effect of web text categorization and try to find the values of the tag weighting coefficients.

Acknowledgment

This research is supported by National Natural Science Foundation of China (Grant No. 71373291).

References

[1] Miao, D., Duan, Q., Zhang, H., & Jiao, N. (2009). Rough set based hybrid algorithm for text classification. *Expert Systems with Applications, 36(5)*, 9168-9174.

[2] Attardi, G., Gulli, A., & Sebastiani, F. (1999). Automatic web page categorization by link and context analysis. *Proceedings of THAI: Vol. 99* (pp. 105-119).

[3] Shih, L. K., & Karger, D. R. (2004). Using urls and table layout for web classification tasks. *Proceedings of the 13th international Conference on World Wide Web* (pp. 193-202).

[4] Shen, D., Yang, Q., & Chen, Z. (2007). Noise reduction through summarization for Web-page classification. *Information Processing & Management, 43(6)*, 1735-1747.

[5] Huang, J. C. (2009). A new method to weight web pages based on authority changing. *Proceedings of 2009 International Conference on Signal Processing Systems* (pp. 686-687).

[6] Wei, T., Lu, Y., Chang, H., Zhou, Q., & Bao, X. (2015). A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications, 42(4)*, 2264-2275.

[7] Li, F., Lin, A., & Chen, G. (2005). A Chinese text categorization system based on the improved VSM. *Journal of Huazhong University of Science and Technology, 33(3)*, 53-55.

[8] Lu, Y., & Li. Y. (2013). Improvement of text feature weighting method based on TF-IDF algorithm.

Library and Information Service, 57(3), 90-95.

[9] Lu, M., Guo, C., Sun, J., & Lu, Y. (2005). A SVM method for web page categorization based on weight adjustment and boosting mechanism. *Fuzzy Systems and Knowledge Discovery*, 801-810.

[10] Chang, L. W., & Yan, M. Q. (2009). Variable precision rough set weight calculation based on web text classification. *Proceedings of 5th International Conference on Wireless Communications, Networking and Mobile Computing* (pp. 1-4).

[11] Liu, H., & Shao, L. (2010). The study on feature items weight of web text classification. *Science Technology and Industry, 10(2)*, 122-124.

[12] Kennedy, J. (2010). Particle swarm optimization. *Encyclopedia of Machine Learning*, 760-766.

[13] Wang, Z., Zhang, Q., & Zhang, D. (2007). A PSO-based web document classification algorithm. *Proceedings of Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing: Vol. 3* (pp. 659-664).

[14] Liang. T. S., & Xiao, M. Z. (2007). Web text feature extraction with particle swarm optimization. *IJCSNS International Journal of Computer Science and Network Security, 7(6)*, 132-136.

[15] Zhaoxia, T. (2010). Web intelligent classification algorithm based PSO and KNN. *Journal of Taiyuan Normal University, 9(4)*, 55-58.

[16] Lichao, L. Y. C. (2011). Text feature selection method based on particle swarm optimization. *New Technology of Library and Information Service, Z1,*76-81*.*

[17] Wu, Q., Chen, X. & Tan, S. (2011). Content extraction algorithm of HTML pages based on optimized weight. *Journal of South China University of Technology, 39(4)*, 33-37.

[18] Long, L. & Deng, W. (2013). Text content extraction algorithm for green network webpage. *Computer Engineering, 39(7)*, 252-256.

Yonghe Lu is an associate professor of the School of Information Management at Sun Yat-sen University. He is committed to the research on text information analysis and processing, including five directions: text mining, intelligent information processing, custom text classification and clustering, semantic analysis, and public opinion analysis. His research has been published in the Expert Systems with Applications, Applied Mechanics and Materials, New Technology of Library and Information Service, Library and Information Service, Information Studies: Theory & Application, and Journal of Library Science among other outlets.

Yanhong Peng is a graduate of School of Information Management at Sun Yat-sen University. Her current research interests include text classification and intelligent information processing.