Postprint: Application of the Shuffled Frog Leaping Algorithm to Feature Selection Optimization in Text Classification

Authors: Lu Yonghe, Chen Jinghuang

Date: 2017-11-08T00:00:00+00:00

Abstract

Objective: Due to the existence of numerous redundant terms unrelated to classification in text data, a shuffled frog leaping algorithm is introduced to optimize feature selection and improve classification accuracy. Method: CHI and IG are respectively employed to pre-select feature sets of varying dimensions, followed by the introduction of an improved shuffled frog leaping algorithm to perform secondary optimization on the pre-selected feature sets. Each frog' s position represents a feature selection rule, with classification accuracy serving as the algorithm' s fitness function. SVM and KNN classifiers are utilized for calculating classification accuracy in the experiments. Results: The improved frog leaping algorithm achieves superior classification performance compared to CHI and IG, with a maximum improvement reaching 12%. Limitations: Overfitting occurs under a small subset of feature dimensions. Conclusion: The feature selection optimization method combining feature term pre-selection with the improved frog leaping algorithm can effectively eliminate interference from noisy feature terms, thereby enhancing text classification accuracy.

Full Text

Preamble

ChinaXiv Cooperative Journal, Issue 1, 2017, No. 1
Optimizing Feature Selection for Text Classification Using the Shuffled Frog Leaping Algorithm
Lu Yonghe, Chen Jinghuang
(School of Information Management, Sun Yat-sen University, Guangzhou 510006)

Abstract

[**Objective**] Due to the presence of numerous classification-irrelevant redundant terms in text data, this paper introduces the shuffled frog leaping algorithm to optimize feature selection and improve classification accuracy. [**Methods**] We first used CHI and IG methods to pre-select feature sets of varying dimensions, then applied an improved shuffled frog leaping algorithm to conduct secondary optimization on these pre-selected feature collections. Each frog' s position represents a feature selection rule, with classification accuracy serving as the algorithm' s fitness function. SVM and KNN classifiers were employed to calculate classification accuracy in experiments. [**Results**] The improved frog leaping algorithm achieved better classification performance than CHI and IG alone, with a maximum improvement of 12%. [**Limitations**] Overfitting occurred in a small portion of feature dimensions. [**Conclusion**] The combined approach of feature term pre-selection and improved frog leaping algorithm can effectively eliminate interference from noise features, thereby improving text classification accuracy.

Keywords: Feature Selection; Text Classification; Shuffled Frog Leaping Algorithm
Classification Number: TP391

In the field of text information processing, text classification has attracted considerable scholarly attention as an important foundation for information mining, natural language processing, and information retrieval. Text classification technology has evolved from traditional manual classification to machine learning-based automatic classification, achieving significant improvements in both quality and efficiency. However, text data often exhibits characteristics of high dimensionality, sparsity, and multi-labeling, which affect classification performance to some extent, making text feature selection optimization a research hotspot. In the Vector Space Model (VSM), not every feature term in the original feature set is necessary for classification learning; some noise features not only increase dimensionality but also impact overall classification effectiveness. Therefore, dimensionality reduction of the feature set is essential.

This paper employs the shuffled frog leaping algorithm (SFLA), which has seen limited application in the text domain, and improves its encoding rules and individual evolution mechanisms for application in text feature selection optimization. Experimental results demonstrate the effectiveness of this approach.

2.1 Traditional Text Feature Selection Methods

The text classification process primarily includes text preprocessing and segmentation, text representation, feature selection, weight calculation, and classification. Text representation mainly adopts the VSM model. After preprocessing, the resulting feature set has extremely high dimensionality and sparse distribution, with each text represented as a high-dimensional vector. Such high-dimensional vectors impose significant computational burdens on classi-

fiers, making feature selection crucial in text classification. Feature selection yields a representative feature term collection that reduces the spatial dimensionality of each text vector and improves classification efficiency and accuracy. Currently, mainstream feature selection methods include Document Frequency (DF), Chi-square test (CHI), Information Gain (IG), and Mutual Information (MI). Experimental evidence shows that CHI offers good classification performance but high computational overhead. In English text classification, CHI and IG perform best, DF is basically comparable, while MI is relatively poor. In Chinese text classification, CHI performs best, followed by IG, with MI relatively poor and DF in the middle.

However, traditional methods like CHI and IG select feature sets with good discriminative ability and text representativeness through mathematical models, without considering from a textual perspective the mutual influence among feature terms or the overall impact of redundant terms on classification effectiveness. Therefore, based on traditional feature selection methods, this paper introduces an improved SFLA that leverages its strong optimization capability to conduct secondary optimization on pre-selected feature sets, obtaining relatively low-dimensional yet high-precision feature collections that ultimately improve classification results.

2.2 Feature Selection Optimization with Swarm Intelligence Algorithms

In recent years, scholars have increasingly applied swarm intelligence algorithms to text feature selection with notable results. The general approaches fall into two categories: (1) Directly using swarm intelligence algorithms for text feature selection without traditional methods. Representative research includes Tabakhi et al.' s UFSACO method, which introduces Ant Colony Optimization (ACO) into unsupervised feature selection, considering feature correlations to remove redundant terms and achieve dimensionality reduction, demonstrating better performance than traditional methods. Liu Yanan applied genetic algorithm (GA)-based feature selection to KNN classification with dynamic K-value acquisition. Liu Kui constructed a text feature selection model based on invasive weed optimization, which gives low-weight terms opportunities for selection while preserving advantages for high-weight terms, improving comprehensiveness and accuracy. (2) Combining swarm intelligence algorithms with traditional feature selection methods, where traditional methods first produce pre-selected feature sets that are then refined by swarm intelligence algorithms to obtain high-precision collections and improve classification. Representative work includes Uguz' s use of IG with GA and Principal Component Analysis (PCA) for secondary selection and extraction to remove irrelevant terms. Javed et al. used BNS and IG for pre-selection followed by Markov Blanket Filter (MBF) for secondary screening. Lu et al. used CHI for pre-selection and six improved particle swarm optimization (PSO) algorithms for refinement, showing asynchronous improved PSO achieved the best results.

This paper combines SFLA with traditional feature selection methods, first conducting feature term pre-selection, then introducing an improved binary SFLA for feature refinement to obtain high-precision feature collections and ultimately improve text classification performance.

2.3 Shuffled Frog Leaping Algorithm

The shuffled frog leaping algorithm, proposed by Eusuff et al., is a collaborative search swarm intelligence algorithm that combines characteristics of memetic algorithms (MA) and particle swarm optimization, featuring both genetic properties of MA and social information sharing of PSO. The algorithm has a simple and reasonable flow, few parameters, fast convergence, and strong global optimization capability.

SFLA was originally inspired by frog foraging behavior. A frog population P of N frogs searches for limited and optimal food sources in an S-dimensional constrained space. Each frog i' s position is represented by $X_i = (x_{i1}, x_{i2}, ..., x_{iS})$, where S represents the spatial dimension. In optimization problems, X_i represents a feasible solution vector. The fitness $F(X_i)$ of each frog's current position is calculated, and frogs are sorted in descending order by fitness while recording the global best position X_g. The entire population is divided into n memes, each containing m frogs, following the grouping rule: frog 1 to meme 1, frog 2 to meme 2, ···, frog m to meme m, frog m+1 to meme 1, and so on. Each meme' s local best solution X_b and worst solution X_w are recorded. Each meme then undergoes internal evolution according to:

$$D = \text{rand}() \times (X_b - X_w) \quad (1)$$

where D represents the step size for each jump, rand() is a random number between 0 and 1, and X'_w represents the position after jumping:

$$X'_w = X_w + D \quad (2)$$

The new position X'_w is calculated using formulas (1) and (2). If the fitness $F(X'_w)$ of X'_w is better than $F(X_w)$, X'_w replaces X_w for the next internal evolution. Otherwise, X_g replaces X_b in formula (1) to recalculate X'_w. If $F(X'_w)$ is still not better than $F(X_w)$, a random X'_w is generated to replace X_w. When all memes reach the maximum internal evolution count L, all frogs are remixed, re-sorted by fitness, the global best solution X_g is updated, and the next generation population is constructed until reaching maximum total iterations T or satisfying termination conditions.

SFLA has been applied to water resource network optimization, bridge deck repair, dynamic optimal power flow calculation in wind power systems, distributed wind generator planning models, and speech recognition. However, literature

shows limited research on SFLA in text information processing. Xu Fang improved traditional SFLA and combined it with K-means and FCM for text clustering, improving Web text clustering precision. Wei Jianxing et al. also combined SFLA with K-means to improve clustering performance. In text classification, Sun et al. used SFLA directly as a classification algorithm with LDA for feature selection, improving Web text classification accuracy. To date, SFLA has seen limited application in text information processing. This paper attempts to improve SFLA and combine it with traditional feature selection methods, verifying its effectiveness and feasibility through experiments.

3. Text Feature Selection Optimization Based on Shuffled Frog Leaping Algorithm

3.1 Algorithm Improvements

(1) Encoding Rules

Since text feature selection optimization is essentially a combinatorial optimization problem, SFLA is improved with binary encoding rules. Each frog's position represents a feature selection rule, where each dimension corresponds to a feature term with two possible outcomes: selected (1) or not selected (0). Each solution vector (frog position) can be represented as $X_i = \{x_{i1}, x_{i2}, ..., x_{iS}\} \in \{0,1\}^S$, where x_{ij} represents the j-th component of the i-th solution vector, taking only 0 or 1. If $x_{ij} = 1$, the j-th feature term is selected; if $x_{ij} = 0$, it is not selected.

(2) Individual Evolution Mechanism Improvement

Since this paper uses binary-encoded SFLA, the standard evolution mechanism (formulas (1) and (2)) is no longer applicable. Therefore, we improve the individual evolution mechanism as follows to better suit text feature selection optimization. The improved process is shown in Figure 1 [Figure 1: see original paper].

First, we calculate the intersection G of selected feature terms between the best solution X_b and worst solution X_w in a meme (i.e., for the j-th component/feature term, all dimensions where both X_b and X_w equal 1). Treating X_b and X_w as sets, G is their intersection:

$$G = X_b \cap X_w \quad (3)$$

Then we calculate each frog' s jumping step size D_{new} using:

$$D_{new} = R_1 \cup R_2 \quad (4)$$

where $(X_b - X_w)$ and $(X_w - X_b)$ represent set difference operations. r_1 and r_2 are random integers between 0 and 100. R_1 takes the top r_1 percent of feature elements from $(X_b - X_w)$, while R_2 takes the top r_2 percent from $(X_w - X_b)$. The union of R_1 and R_2 forms set D_{new}, representing a frog' s jumping step

size. For example, when $r_1 = 20$, $r_2 = 40$, $(X_b - X_w)$ has 100 elements, and $(X_w - X_b)$ has 200 elements, we take $100 \times 20\% = 20$ features from the first set and $200 \times 40\% = 80$ from the second, forming D_{new} with $20 + 80 = 100$ features. The position update after a frog' s jump within a meme is:

$$X'_w = G \cup D_{new} \quad (5)$$

This improvement is based on the following rationale: First, we find intersection G between X_b and X_w to preserve their "common features," allowing new individuals to "inherit" these common features while continuing to evolve toward better positions. When calculating the jumping step, we select top features from each solution' s "unique" features to form D_{new}. This allows new individuals to randomly "inherit" certain proportions of unique features from both X_b and X_w, enabling evolution in certain directions. Moreover, since candidate feature sets are pre-selected by CHI or IG and sorted by their scores (higher scores indicating better representativeness), we select top-ranked features.

(3) Maximum Step Size D_{max} Improvement
With the improved step size calculation for binary SFLA, we must also redefine the maximum step size D_{max_new}.

We first define a new variable: difference degree ($diff$), representing the proportion of differing solution components between the new individual X'_w and original X_w across corresponding dimensions. D_{max_new} refers to the maximum allowed $diff$ between X'_w and X_w. For example, if $X'_w = \{0, 1, 1, 1, 0, 0\}$ and $X_w = \{1, 0, 1, 1, 0, 1\}$ differ at dimensions 1, 2, and 6, then $diff = (3/6) \times 100\% = 50\%$. The variable $diff$ is introduced to calculate the difference proportion between binary-encoded frog individuals, equivalent to step size in standard SFLA. However, since the step size calculation formula is modified, step size no longer represents the difference degree. Therefore, D_{max_new} in the improved algorithm refers to the maximum allowed difference degree between the new individual X'_w and original X_w.

3.2 Parameter Settings

The improved binary SFLA requires five parameters: frog population size N, number of memes n, maximum step size D_{max}, internal evolution count L, and total iterations T. Parameter settings significantly impact algorithm performance.

Population size N refers to the total number of frogs, representing the number of initial solution vectors for combinatorial optimization problems. Generally, N relates to problem complexity, but due to high computational overhead in fitness calculation, we set $N = 20$. The number of memes n depends on the size m of each meme; we set $n = 5$, resulting in 4 frogs per meme. The maximum step size D_{max} controls global search capability. We set $D_{max} = 45$, meaning the difference degree between new and original individuals cannot exceed 45%.

Parameter L determines internal evolution times within memes, while total iterations T relates to problem complexity (higher complexity requires larger T to increase optimal solution probability). Due to computational costs, we set $L = 10$ and $T = 10$.

3.3 Fitness Function

The fitness function in swarm intelligence algorithms calculates individual fitness based on optimization objectives. This paper introduces SFLA for feature selection optimization, aiming to reduce feature set dimensionality while improving classification accuracy. Therefore, we use text classification accuracy to measure each frog' s position quality, guiding frogs to "leap" toward higher accuracy positions:

$$\text{Fitness} = \frac{\text{Number of correctly classified test texts}}{\text{Total number of texts in test set}}$$

3.4 Algorithm Design

The text feature selection optimization algorithm based on improved SFLA proceeds as follows:

Input: Training text set TR, test text set A, pre-selected feature word count (feature space dimension) S to be obtained through CHI or IG, initialized frog count N, meme count n, maximum step size D_{max}, maximum internal evolution count L, total iterations T.

Output: Feature set after SFLA secondary optimization.

1. Segment training set TR using segmentation software, then apply CHI and IG for feature pre-selection to obtain candidate feature sets.
2. Randomly assign values from $\{0, 1\}$ to each dimension of every frog' s position. A value of 1 selects the corresponding feature term, while 0 deselects it, serving as initial positions.
3. Calculate each frog' s fitness (classification accuracy). Use features with value 1 to construct test set A's feature vectors, then calculate classification accuracy using a classifier as the fitness value.
4. Follow the improved SFLA process until iterations reach T or other termination conditions are met. Output the optimal solution X_g and features with value 1 in X_g as the secondary-optimized feature set.

The algorithm flow is shown in Figure 2 [Figure 2: see original paper].

4. Experiments

4.1 Experimental Design

The experiments consist of two parts: (1) Direct use of traditional CHI or IG selected feature sets for classification; (2) Introduction of SFLA for secondary

optimization to obtain high-precision feature sets for classification, as shown in Figure 3 [Figure 3: see original paper].

In the direct classification process using CHI or IG, the dataset comprises training set TR and test set B to calculate classification accuracy for original feature sets. In the SFLA optimization process, since fitness (classification accuracy) must be calculated, the dataset requires both training and test sets, so TR and test set A serve as the training set for model building. After obtaining high-precision feature sets, test set B evaluates their performance to verify whether they generalize well to other test sets. Additionally, since SFLA requires multiple accuracy calculations, using a large test set would increase time overhead, so test set A is smaller, composed by randomly extracting 15% of texts from each category of test set B.

To demonstrate algorithm effectiveness, experiments use both English and Chinese datasets. Experiment 1 uses a subset of the Reuters-21578 corpus; Experiment 2 uses a subset of the Intelligent Information Processing Laboratory corpus at Sun Yat-sen University' s School of Information Management (referred to as the Laboratory corpus).

The experimental environment was 32-bit Windows 10, 4GB RAM, i5-2400 processor, with Java programming. Text preprocessing used the Lucene open-source package; segmentation used the Institute of Computing Technology, Chinese Academy of Sciences' ICTCLAS system. CHI and IG were used for pre-selection; TF-IDF for feature weighting; SVM and KNN for classification.

Specific steps: 1. Use TR and test set B with CHI to pre-select 12 feature sets of dimensions 100-1200 (CHI_{100}-CHI_{1200}), calculating accuracies P_{CHI}. 2. Apply improved binary SFLA for secondary optimization on these 12 sets. Use TR and test set A as training data to calculate fitness (accuracy). Output SFLA' s optimal solutions: high-precision feature sets after secondary optimization. 3. Calculate accuracies P_{CHI_SFLA} for these optimized sets using TR and test set B. 4. Repeat steps 1-3 using IG method to obtain P_{IG} and P_{IG_SFLA}. 5. Compare P_{CHI}, P_{IG} with P_{CHI_SFLA}, P_{IG_SFLA} across 12 dimensions. 6. Conduct paired sample T-test on all accuracy data divided into P_{old} (before SFLA) and P_{new} (after SFLA) groups.

4.2 Experimental Results

Experiment 1: Reuters-21578 Corpus
The Reuters-21578 corpus contains 8 categories: acq, crude, earn, grain, interest, money-fx, ship, trade. The large test set and training set were split at a 1:2.5 ratio, with specific quantities shown in Table 1 .

With the SVM classifier, secondary optimization results for CHI or IG pre-selected feature sets are shown in Table 2 . Line charts for CHI and IG groups are shown in Figures 4 [Figure 4: see original paper] and 5 [Figure 5: see original paper], where the x-axis represents pre-selected feature counts. CHI_{SFLA}

and IG_{SFLA} denote methods using CHI or IG pre-selection followed by SFLA optimization. On Reuters-21578 with SVM, the improved SFLA secondary optimization clearly outperformed traditional CHI and IG, with accuracy improvements increasing with dimensionality.

With the KNN classifier, results are shown in Table 3 and Figures 6 [Figure 6: see original paper] and 7 [Figure 7: see original paper]. On Reuters-21578 with KNN, the improved SFLA achieved higher accuracy than CHI and IG in most dimensions. At dimension 400, IG_{SFLA} matched IG' s accuracy but with reduced dimensionality, indicating that IG' s 400-dimensional set contained classification-irrelevant terms that could be removed.

Experiment 2: Laboratory Corpus
The Laboratory corpus, collected by Sun Yat-sen University' s Intelligent Information Processing Laboratory, contains 13 categories. We selected 8 categories with more texts: education, entertainment, event, finance, game, occultism, sport, technology. From each category, 200 texts were randomly selected as training set TR (1,600 total), another 200 per category as test set B (1,600 total), and 20 per category as test set A (160 total). After preprocessing, segmentation, deduplication, and stopword removal, the training set yielded 52,794 features.

With SVM classifier, results are shown in Table 4 and Figures 8 [Figure 8: see original paper] and 9 [Figure 9: see original paper]. The improved SFLA secondary optimization outperformed CHI and IG, with both achieving highest accuracy at 1,000 dimensions. CHI_{SFLA} improved about 7% over CHI at 400 dimensions, while IG_{SFLA} improved about 9% over IG at 300 dimensions. The optimization effect was more pronounced when traditional methods produced lower baseline accuracies.

With KNN classifier, results are shown in Table 5 and Figures 10 [Figure 10: see original paper] and 11 [Figure 11: see original paper]. CHI_{SFLA} outperformed CHI, though improvements were less significant at 100 and 1,000 dimensions while still achieving dimensionality reduction. IG_{SFLA} clearly outperformed IG, with improvements reaching 12% at 1,000 and 1,100 dimensions.

4.3 Paired Sample T-Test

All accuracy data were divided into two groups: P_{old} (before SFLA) and P_{new} (after SFLA). Paired sample T-test in SPSS yielded results shown in Table 6 . With Sig. $= .000 < 0.01$, at 99% significance level, P_{old} and P_{new} show significant differences, confirming that SFLA feature optimization significantly improves text classification accuracy.

5. Conclusion

Both experiments demonstrate that the improved SFLA-based text feature selection optimization algorithm achieves better classification performance than traditional CHI and IG, validating its feasibility and effectiveness. Traditional methods like CHI and IG select features through mathematical models from a statistical perspective without considering feature interactions or redundant terms' overall impact. Consequently, their candidate sets contain many noise features that affect classifier performance. The improved SFLA conducts secondary optimization, leveraging its iterative optimization and good convergence properties to retain discriminative features while removing noise terms, substantially improving classification accuracy.

This paper introduces SFLA, rarely applied in text processing, to text feature selection optimization from a holistic perspective. Comparative experiments show the improved SFLA method achieves higher accuracy than CHI and IG by removing noise features and reducing their impact. However, current parameter settings were determined through small-scale tests. Future work will optimize SFLA parameters to find optimal value ranges, yielding better high-precision feature sets and classification performance.

References

[1] Pang Guansong, Jiang Shengyi. Text Automatic Classification Technology Research [J]. Information Studies: Theory & Application, 2012, 35(2): 123-128.

[2] Wu Ke. A Study on Text Categorization Based on Machine Learning [D]. Shanghai: Shanghai Jiaotong University, 2008.

[3] Wu Jianjun, Kang Yaohong. Comparison and Improvement of Feature Selection for Text Categorization [J]. Journal of Zhengzhou University: Natural Science Edition, 2007, 39(2): 110-113.

[4] Yang Y, Pedersen J O. A Comparative Study on Feature Selection in Text Categorization[C]//Proceedings of the 14th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc., 1997: 412-420.

[5] Fu Fa. Comparison of Feature Selection in Chinese Text Categorization [J]. Modern Computer, 2008(6): 43-45.

[6] Tabakhi S, Moradi P, Akhlaghian F. An Unsupervised Feature Selection Algorithm Based on Ant Colony Optimization [J]. Engineering Applications of Artificial Intelligence, 2014, 32: 112-123.

[7] Liu Yanan. Research of Feature Extraction Technology in KNN Text Classification Based on the Genetic Algorithm [D]. Beijing: China University of Petroleum, 2011.

[8] Liu Kui. An Invasive Weed Optimization Algorithm for Text Feature Selection [D]. Chongqing: Southwest University, 2013.

[9] Uguz H. A Two-stage Feature Selection Method for Text Categorization by Using Information Gain, Principal Component Analysis and Genetic Algorithm [J]. Knowledge-Based Systems, 2011, 24(7): 1024-1032.

[10] Javed K, Maruf S, Babri H A. A Two-stage Markov Blanket Based Feature Selection Algorithm for Text Classification [J]. Neurocomputing, 2015, 157: 91-104.

[11] Lu Y, Liang M, Ye Z, et al. Improved Particle Swarm Optimization Algorithm and Its Application in Text Feature Selection [J]. Applied Soft Computing, 2015, 35(C): 629-636.

[12] Eusuff M M, Lansey K E. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm [J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.

[13] Cui Wenhua, Liu Xiaobing, Wang Wei, et al. Survey on Shuffled Frog Leaping Algorithm[J]. Control and Decision, 2012, 27(4): 481-486, 493.

[14] Elbehairy H, Elbeltagi E, Hegazy T, et al. Comparison of Two Evolutionary Algorithms for Optimization of Bridge Deck Repairs [J]. Computer-Aided Civil and Infrastructure Engineering, 2006, 21(8): 561-572.

[15] Chen Gonggui, Li Zhihuan, Chen Jinfu, et al. SFLA Algorithm Based Dynamic Optimal Power Flow in Wind Power Integrated System [J]. Automation of Electric Power Systems, 2009, 33(4): 25-30.

[16] Zhang Shenxi, Chen Kai, Long Yu, et al. Distributed Wind Generator Planning Based Shuffled Frog Leaping Algorithm [J]. Automation of Electric Power Systems, 2013, 37(13): 76-82.

[17] Yu Hua, Huang Chengwei, Jin Yun, et al. Speech Emotion Recognition Based on Modified Shuffled Frog Leaping Algorithm Neural Network [J]. Signal Processing, 2010, 26(9): 1294-1299.

[18] Xu Fang. Research on Web Text Cluster Algorithm Based on Shuffled Frog-leaping Algorithm [D]. Wuxi: Jiangnan University, 2013.

[19] Yu Jianxing, Cui Donghua, Ning Xiaoqing. Application of Shuffled Frog-leaping Algorithm to Web's Text Cluster Technology [J]. Computer Development & Applications, 2011, 24(5): 35-37.

[20] Sun X, Wang Z. An Efficient Document Categorization Algorithm Based on LDA and SFL [C]//Proceedings of the 2008 International Seminar on Business and Information Management. IEEE, 2008: 113-115.

[21] NLPIR Chinese Word Segmentation System [EB/OL]. [2016-03-17]. http://ictclas.nlpir.org.

[22] Lu Yonghe, Peng Yanhong. The Classification System Construction for Internet Information both Practical and Scientific[J]. Library and Information, 2015(3): 118-124.

Author Contributions

Lu Yonghe: Proposed research ideas and experimental suggestions, revised the paper.
Chen Jinghuang: Analyzed data, designed and implemented algorithms, conducted experiments, wrote and revised the final manuscript.

Conflict of Interest

All authors declare no conflict of interest.

Supporting Data

Supporting data is available in the journal's online version at http://www.infotech.ac.cn:
[1] Lu Yonghe, Chen Jinghua. Experimental datasets.rar. Text subsets selected from Reuters-21578 corpus and Sun Yat-sen University School of Information Management Jitian Intelligent Laboratory corpus.
[2] Lu Yonghe, Chen Jinghua. Pre-selected feature sets for experiment input.rar. Feature word sets pre-selected through CHI and IG.
[3] Lu Yonghe, Chen Jinghua. Feature sets from experiment output.rar. Feature word sets refined by improved SFLA.
[4] Lu Yonghe, Chen Jinghua. Experimental classification results.xlsx. Text classification accuracies calculated using feature sets refined by SFLA.

Received: 2016-09-30
Revised: 2016-12-12

Note: Figure translations are in progress. See original paper for figures.