ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201711.01201

Research on Chinese News Text Classification Method Based on Denoising Autoencoder (Post-print)

Authors: Liu Hongguang, Ma Shuanggang, Liu Guifeng

Date: 2017-10-11T00:00:00+00:00

Abstract

【Objective】Drawing upon deep learning theory, this study addresses the problem that traditional feature selection methods tend to cause ambiguous feature items and degraded classification accuracy.

【Method】 For Chinese news text classification, denoising autoencoders are employed to construct a deep network that learns compressed and distributed representations of text, with an SVM algorithm applied at the final network layer to classify instances into specific categories.

【Results】 As the number of samples increases, classification accuracy, recall, and F-measure all increase, achieving superior classification performance compared to KNN, BP, and SVM algorithms, with an average classification accuracy exceeding 95%.

【Limitations】The dataset size remains relatively small, and the advantage of deep learning in parallel processing of large-scale data has not been fully exploited.

【Conclusion】 This method can improve the accuracy of feature extraction and enhance classification effectiveness.

Full Text

Classifying Chinese News Texts with Denoising Autoencoder

Liu Hongguang, Ma Shuanggang, Liu Guifeng
(Institute of Scientific & Technical Information, Jiangsu University, Zhenjiang 212013, China)

Abstract

[**Objective**] This study addresses the problem that traditional feature selection methods often lead to ambiguous feature terms and degraded classification accuracy by leveraging deep learning theory. [**Methods**] We employ a denoising autoencoder to construct a deep network that learns compressed and distributed representations of Chinese news texts, with an SVM algorithm applied at the final layer to classify texts into specific categories. [**Results**] As the sample size increases, classification accuracy, recall, and F-measure all improve, achieving superior performance compared to KNN, BP, and SVM algorithms alone, with an average classification accuracy exceeding 95%. [**Limitations**] The dataset remains relatively small and does not fully exploit the advantage of deep learning for parallel processing of large-scale data. [**Conclusions**] The proposed method enhances the accuracy of feature extraction and improves classification effectiveness.

Keywords: Denoising Autoencoder; Support Vector Machine; Feature Extraction; Text Classification

The rapid development of information technology has led to exponential growth in massive information data, heralding the era of big data. In this context, the effective organization and utilization of massive text information has become increasingly important. Text classification technology has been widely applied across various domains of social life due to its advantages in efficiently and accurately managing and locating massive information, and has achieved considerable development.

In text classification, the Vector Space Model (VSM) is generally adopted for text representation. However, the complexity of text data structure and semantics results in high-dimensional feature vector spaces even after word segmentation and stop word removal, necessitating further optimization. The most common approach is dimensionality reduction, which substantially decreases the scale of text data to be processed by the classifier and significantly reduces noise. Conventional dimensionality reduction methods include feature selection and feature extraction.

Feature selection typically employs statistical methods to obtain a subset of the original feature set, with common techniques including CHI-Square test [1], Mutual Information (MI) [2], and Information Gain (IG) [3]. Related research on feature selection methods [4] indicates that IG performs relatively well. Feature extraction methods can construct or synthesize new feature terms from the original set, thereby reducing the dimensionality of text feature space. Researchers have proposed various feature extraction approaches, such as the mutual nearest neighbor clustering algorithm [5] and maximum entropy model [6]. Although these traditional feature selection and extraction methods can identify most features, they generally suffer from poor feature discriminability. For instance, features that rarely appear in a designated category but frequently occur in others may be selected, leading to loss of feature terms. Extracted features may

contain errors and thus cannot accurately represent the original dataset, particularly when extracted from large-scale, high-dimensional datasets, ultimately resulting in decreased classification accuracy.

In 2006, Hinton et al. [7] introduced the application of deep networks constructed with autoencoders (AE) for feature dimensionality reduction in images and texts, achieving superior results compared to traditional methods. Consequently, scholars have applied AE to feature extraction and proposed various improved algorithms, including Sparse Autoencoder (SAE) [8], Denoising Autoencoder (DAE) [9], and Convolutional Autoencoder (CAE) [10]. Among these, DAE has been widely applied in feature extraction, primarily for dynamic video textures [11], audio [12], images [13], and medical diagnosis [14]. This paper focuses exclusively on the application of DAE to text feature extraction.

Text contains considerable noise that affects classification accuracy. Therefore, relevant scholars have adopted DAE for text feature extraction. For example, Liu et al. [15] proposed a feature extraction and clustering algorithm based on deep DAE for short texts, effectively addressing the high-dimensional and sparse nature of short text vector spaces. Qin et al. [16] improved DAE to achieve unsupervised sample classification, demonstrating good adaptability to highly imbalanced samples. Although research in this area remains relatively limited, these studies show that deep networks constructed with DAE can extract more accurate feature encodings representing original texts while effectively removing noise, thereby substantially improving classification accuracy when combined with classification algorithms.

Unlike studies [15-16], this paper applies DAE to feature extraction from news texts. We first use DAE to construct a deep network that automatically learns low-dimensional features from texts. At the topmost layer, we employ the linear classifier Support Vector Machine (SVM) to classify the obtained low-dimensional feature encodings and implement classification based on the output results. Finally, we compare our approach with K-Nearest Neighbors (KNN), SVM, and Error Back Propagation (BP) neural network algorithms to demonstrate its effectiveness.

2. Theoretical Foundations

2.1 Feature Extraction Based on DAE

An autoencoder (AE) [7, 17] constructs an unsupervised deep network structure. Through unsupervised layer-wise greedy pre-training and systematic parameter optimization, it obtains a multi-layer nonlinear network that extracts hierarchical features from unlabeled high-dimensional complex input data and obtains distributed feature representations of original data, enabling good reproduction of input signals. AE consists primarily of two components: an encoder and a decoder, as illustrated in [Figure 1: see original paper].

However, AE cannot eliminate noise interference in data. To address this limi-

tation and obtain more robust features, Vincent et al. [18] proposed randomly processing the original input matrix X using a probability distribution (typically binomial distribution) to corrupt the original data and obtain $\tilde{X}$. This corrupted data is then encoded, with subsequent operations identical to the standard AE process. This improved encoder is called a denoising autoencoder, with its structure illustrated in [Figure 2: see original paper].

The encoder $\hat{f}(x)$ reduces high-dimensional data. It first corrupts the input vector x to obtain $\tilde{x}$, then feeds it into the encoder $\hat{f}(x)$. Through linear transformation and activation functions, it finally obtains the hidden encoding result y. The decoder $g(y)$ reconstructs the low-dimensional encoding by mapping hidden layer data back to the reconstructed z. These are expressed as:

$$y = f(x) = S_f(Wx + b_y)$$
$$z = g(y) = S_g(W'y + b_z)$$

where S_f is the nonlinear activation function, implemented as a sigmoid function: $S_f(y) = \text{sigmoid}(y)$. S_g is the decoder' s activation function, also using the sigmoid function in this paper. W' is the transpose of W, so only W needs to be trained. b_y and b_z are bias vectors.

The DAE training process involves finding parameters $\{W, b_y, b_z\}$ that minimize the reconstruction error over the training sample set D. The reconstruction error is expressed as:

$$L(x, g(f(x)))$$

where L is the reconstruction error function. Research [19] shows that cross-entropy loss consistently outperforms squared difference loss in experiments. Therefore, this paper adopts the cross-entropy loss function:

$$L(x, z) = -\sum_{i=1}^{n}[x_i \ln z_i + (1 - x_i)\ln(1 - z_i)]$$

where n is the number of training samples, x_i is the i-th input, and z_i is the i-th reconstructed data after decoding.

The autoencoder uses the classic stochastic gradient descent algorithm for training. In each iteration, the weight matrix is updated using:

$$W \leftarrow W - \varphi \times \frac{\partial L(x, y)}{\partial W}$$

where φ is the learning rate. b_y and b_z are updated using the same approach.

2.2 Classification Based on SVM

The SVM algorithm [20] aims to find an optimal hyperplane that separates positive and negative examples, maximizing the margin between them while maintaining equal distance to the nearest examples from each class. For unknown samples, classification is determined by which side of the hyperplane they fall on.

In the linearly separable case, the classification equation is $(w \cdot x) + b = 0$. After normalization, each linearly separable sample (x_i, y_i), where $i = 1, 2, \ldots, l$, $x \in \mathbb{R}^n$, and $y \in \{-1, 1\}$, satisfies:

$$y_i[(w \cdot x_i) + b] \geq 1, \quad i = 1, 2, \ldots, l$$

where x_i is the i-th input sample, l is the number of samples, w is the adjustable weight vector, and b is the bias. $y_i \in \{-1, 1\}$ represents the expected output corresponding to x_i.

To obtain the optimal classification hyperplane, we need to maximize the classification margin margin $= 2/\|w\|$, which is equivalent to minimizing $\|w\|$, subject to the constraints above. This is a typical quadratic programming problem with the objective function:

$$\min L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{l} \alpha_i\{[(w \cdot x_i) + b]y_i - 1\}$$

Using Lagrangian optimization, this problem can be transformed into its dual form by adding the constraint $\sum_{i=1}^{l} \alpha_i y_i = 0$ for $i = 1, 2, \ldots, l$, and solving for the maximum of:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j \cdot y_i y_j (x_i \cdot y_j)$$

where $\alpha_i \geq 0$ are the Lagrange multipliers corresponding to each sample. Only a few Lagrange coefficients α_i^* are non-zero, and the samples corresponding to these non-zero coefficients are defined as support vectors.

By adopting appropriate kernel functions in the optimal classification surface, linear classification after some nonlinear transformation can be achieved without increasing computational complexity. The objective function then becomes:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j \cdot y_i y_j K(x_i, y_j)$$

The final trained classification function is:

$$d(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b$$

which is the support vector machine. The sign of $d(x)$ determines the category of input sample x.

3. DAS Text Classification Model

The Chinese news text classification model combining DAE and SVM (referred to as the DAS classification model) consists of six main components: NLPIR text segmentation, stop word removal, text representation, DAE feature extraction, SVM classification, and classification effectiveness evaluation, as illustrated in [Figure 3: see original paper].

(1) Unlike Western languages, Chinese lacks clear boundaries between words, requiring special word segmentation operations. This paper employs the mature NLPIR Chinese word segmentation system [21] for Chinese news text segmentation.

(2) After segmentation, numerous stop words remain, including punctuation marks and common words that do not contribute to classification. This paper compiles a comprehensive stop word list by merging multiple existing lists to eliminate these words and obtain candidate feature terms that can represent text characteristics.

(3) The candidate feature terms obtained after step (2) remain numerous and high-dimensional, requiring preliminary screening. This paper applies the information gain algorithm for initial feature selection, then uses the VSM model for text feature representation.

(4) The feature representation from step (3) is fed into a deep network constructed by DAE. After layer-wise training, a relatively low-dimensional feature encoding is obtained.

(5) At the final layer of the deep network, the SVM algorithm classifies the low-dimensional encoding obtained in step (4), with classification implemented based on the output results.

(6) The classification effectiveness is evaluated, and the text classification model is continuously optimized based on evaluation results until satisfactory performance is achieved.

The most fundamental and important task in text classification is feature term extraction in step (4). Text contains substantial redundant data and noise, which can easily cause errors and poor recognition during feature extraction, thereby affecting final classification performance. To achieve better classification results, the impact of redundant data and noise must be minimized as much as possible. DAE corrupts input data and trains feature coefficients using these

corrupted data, resulting in relatively small noise. Moreover, corrupted data can, to some extent, reduce the gap between training and test data.

Therefore, to extract and encode more robust features from texts and eliminate noise effects for better classification performance, this paper draws on relevant theories [7, 17-20] and applies DAE to Chinese news text feature extraction. We construct a deep network that obtains low-dimensional feature encoding through layer-wise training, extracting the most representative low-dimensional features and achieving dimensionality reduction of high-dimensional text data. The SVM algorithm is then used at the topmost layer of the deep network to classify and output the low-dimensional encoding, with final classification implemented based on the output results. The deep network constructed based on DAE and SVM is illustrated in [Figure 4: see original paper].

In this architecture, text candidate terms first undergo processing by hidden layers composed of multiple DAEs to obtain low-dimensional encoding. At the topmost layer, LibSVM classifies the low-dimensional encoding, and the entire text classification model is trained through fine-tuning based on the classification output.

4. Experiments

4.1 Simulation Experiment Steps

(1) Experiment 1: Classic Experiments

To test the superiority of the DAS classification model, we conduct comparative simulation experiments on the same dataset using classic training algorithms (KNN, SVM, and BP neural network with one hidden layer) after feature selection via the information gain method. We compare their classification recall, precision, and F-measure with those of the DAS model. The specific steps for comparative experiments using classic algorithms are as follows:

Dataset Selection: The news text dataset for simulation experiments [22] was provided by Li Ronglu from the Department of Computer Information and Technology at Fudan University. This dataset is well-annotated, moderately sized, and suitable for small-to-medium scale classification experiments. The "answer" subset serves as test corpus with 9,833 documents, while the "train" subset serves as training corpus with 9,804 documents across 20 categories. We randomly select 6 categories with 1,000 documents each, creating 4 training sets of 200, 400, 600, and 800 documents respectively, with 200 documents in each set reserved as test data. Specific category information and experimental grouping design are shown in .

Text Dataset Preprocessing: This includes text segmentation and stop word removal. The NLPIR Chinese word segmentation system is used for text segmentation, offering functions including Chinese word segmentation, part-of-speech tagging, named entity recognition, user dictionary functionality, microblog segmentation, new word discovery, and keyword extraction. It is a

relatively mature and widely-used Chinese text segmentation system in China. This paper analyzes text semantic features and compiles a comprehensive stop word list by integrating online resources, as shown in .

Text Representation: After preprocessing, text dimensionality remains too high, requiring preliminary dimensionality reduction. We calculate the information gain value for each feature term using the formula:

$$IG(t) = -\sum_{i=1}^{m} P(c_i) \log P(c_i) + P(t) \sum_{i=1}^{m} P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^{m} P(c_i|\bar{t}) \log P(c_i|\bar{t})$$

where m is the total number of categories, c_i represents a category, $P(c_i)$ is the probability of category c_i occurring, $P(t)$ is the probability of documents containing feature term t, $P(\bar{t})$ is the probability of documents not containing t, $P(c_i|t)$ is the probability of belonging to c_i given t, and $P(c_i|\bar{t})$ is the probability of belonging to c_i given the absence of t.

After calculating information gain values, we sort them and retain the top 5,000 feature terms for vector space model representation, as shown in . In this feature word matrix, n represents the total number of feature terms across all documents, with each feature term corresponding to one dimension in the feature space; m represents the number of texts to be classified. Each document is represented as a point in N-dimensional space: $V(d_i) = ((t_1, w_{i1}), (t_2, w_{i2}), \ldots, (t_n, w_{in}))$, where the feature weight w_{ij} is the TF-IDF value of each feature term, calculated as:

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

where $W(t,d)$ is the weight of feature term t in text d, $tf(t,d)$ is the frequency of term t in document d, n_t is the number of texts containing feature t in the corpus, $\log(N/n_t + a)$ is the inverse document frequency function (larger n_t yields smaller values), a is a constant (set to 0.01 in this paper), and the denominator is a normalization factor.

Classification Training: Supervised classification training is performed using classification algorithms to obtain classification parameters, with test datasets used for classification testing. The selected algorithms are: KNN algorithm (a relatively simple C language implementation designed for this study), SVM algorithm (using the mature LibSVM), and BP algorithm (using MATLAB' s built-in neural network toolbox).

Classification Effectiveness Evaluation and Comparison: We evaluate final classification results using recall R, precision P, and F-measure:

$$R = \frac{M}{M+T}, \quad P = \frac{M}{M+N}, \quad F = \frac{2RP}{R+P}$$

where M is the number of correctly classified texts in the category, N is the number of texts incorrectly assigned to the category, and T is the number of texts belonging to the category but misclassified into other categories.

(2) Experiment 2: Optimization Experiment

The DAS classification model for news text classification follows steps – and identically to the classic experiments, differing only in step (classification training).

In the DAS classification model, after obtaining matrix representation from training text datasets through steps –, we feed it into a deep network constructed by DAE. Using unsupervised learning, we perform layer-wise dimensionality reduction on the 5,000-dimensional features, with the linear classifier SVM algorithm applied at the final layer for classification output. Parameters are adjusted based on output results to obtain final classification parameters for testing dataset evaluation.

Feature Dimensionality Reduction: The vectorized matrix representation X of texts is first corrupted to obtain matrix $\tilde{X}$, which is then fed into the encoder to obtain encoding. This passes through a decoder to reconstruct a matrix, which is compared with the original matrix to obtain reconstruction error. Encoder and decoder parameters are adjusted to minimize reconstruction error, yielding final encoding. The encoding features from the upper layer serve as input for the lower layer, with the same method applied to obtain lower-layer encoding. This process continues until encodings for the specified number of layers are obtained.

The deep network designed in this paper has node configurations of 5000-2500-1200-600-300-100-50-20, plus the final linear classifier SVM, totaling 9 layers. Each layer' s matrix representation undergoes a random zero-masking process before training. After each layer' s training completes, the next layer is trained until the dimensionality reduction process of the denoising autoencoder is finished.

Supervised Fine-tuning: The 20-dimensional feature encoding obtained in step is classified and output using the SVM algorithm, followed by fine-tuning of coefficients across all layers based on the output results. After completing this supervised training, the system is used to classify texts in the test set to evaluate the classification system' s effectiveness. This paper uses the LibSVM algorithm to classify feature encodings obtained after dimensionality reduction for each text, then employs the BP algorithm to fine-tune coefficients from the top down across all layers. The final adjusted coefficients can then be used to classify the test dataset.

4.2 Experimental Results and Analysis

Experiments are conducted under the four training sets, with classification recall, precision, and F-measure shown in [Figure 5: see original paper]. The DAS classification model demonstrates increasing recall, precision, and F-measure as the training set expands. This is because deep networks require relatively large datasets; overly small datasets lead to overfitting and suboptimal classification performance. Therefore, for deep networks, dataset size determines training effectiveness—larger datasets yield better classification results.

Comparative experiments with different classification algorithms yield recall, precision, and F-measure values under the four training sets, as shown in through .

KNN algorithm generates feature vectors for all training texts during training and compares similarity between test text feature vectors and all training text feature vectors during testing, achieving decent results in small-to-medium scale experiments. However, this method heavily depends on selected feature terms; if representative feature terms are not selected, classification performance degrades significantly. As shown in the tables, KNN performs worse than other algorithms.

BP algorithm is a typical shallow neural network that backpropagates errors, distributing them across layers to adjust weight coefficients and complete training. However, BP typically employs only three network layers, yielding poor classification performance with small training sets. As training set size increases, recall, precision, and F-measure all improve to varying degrees.

SVM algorithm aims to find an optimal hyperplane that separates positive and negative examples with maximum margin and equal distance to the nearest examples from each class. For unknown texts, classification is determined by which side of the hyperplane they fall on. While SVM performs well on small-scale sample datasets, its performance on large-sample datasets is slightly inferior, with no significant improvement as dataset size increases.

The DAS classification model uses denoising autoencoders to unsupervisedly learn feature encodings of news texts, aligning with the human brain' s hierarchical analysis approach (word → sentence → paragraph → meaning) for text comprehension. This more accurately simulates the human brain' s understanding of text meaning, yielding better classification performance. As shown in through , the DAS classification model achieves superior classification results.

Deep learning has sparked significant research interest in academia and industry, achieving considerable success. Drawing on DAE and SVM theories, this paper designs the DAS classification model, applying DAE-constructed deep networks to Chinese news text dimensionality reduction and using SVM for classification at the topmost layer. Coefficients across layers are continuously fine-tuned based on classification results, with final testing conducted on the test dataset. Results demonstrate that the DAS classification model reduces noise impact in news

text data and achieves better classification performance than KNN, SVM, and BP algorithms. However, although four datasets of different sizes were tested, the data volume remains relatively small, not fully exploiting deep learning' s advantage in parallel processing of large-scale data. Future research will focus on experiments with large-scale datasets.

References

[1] Pei Yingbo, Liu Xiaoxia. Study on Improved CHI for Feature Selection in Chinese Text Categorization [J]. Computer Engineering and Applications, 2011, 47(4): 128-130.

[2] Xin Zhu, Zhou Yajian. Study and Improvement of Mutual Information for Feature Selection in Text Categorization [J]. Journal of Computer Applications, 2013, 33(S2): 116-118, 152.

[3] Guo Song, Ma Fei. Improving the Algorithm of Information Gain Feature Selection in Text Classification [J]. Computer Applications and Software, 2013, 30(8): 139-142.

[4] Peters C, Koster C H. Uncertainty-based Noise Reduction and Term Selection in Text Categorization [C]. In: Proceedings of the 24th BCS-IRSG European Colloquium on IR Research, Glasgow, UK. Springer, 2002: 248-267.

[5] Lewis D D. Representation and Learning in Information Retrieval [D]. University of Massachusetts, 1992.

[6] Li Xuexiang. Research of Text Categorization Based on Improved Maximum Entropy Algorithm [J]. Computer Science, 2012, 39(6): 210-212.

[7] Hinton G E, Salakhutdinov R R. Reducing the Dimensionality of Data with Neural Networks [J]. Science, 2006, 313(5786): 504-507.

[8] Bengio Y, Lamblin P, Popovici D, et al. Greedy Layer-wise Training of Deep Networks [C]. In: Proceedings of the 20th Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada. 2007, 19: 153-160.

[9] Vincent P, Larochelle H, Bengio Y, et al. Extracting and Composing Robust Features with Denoising Autoencoders [C]. In: Proceedings of the 25th International Conference on Machine Learning. ACM, 2008: 1096-1103.

[10] Masci J, Meier U, Cireşan D, et al. Stacked Convolutional Auto-encoders for Hierarchical Feature Extraction [C]. In: Proceedings of the 21st International Conference on Artificial Neural Networks. Springer Berlin Heidelberg, 2011: 52-59.

[11] Wang Caixia, Wei Xueyun, Wang Biao. Dynamic Texture Classification Method Based on Stacked Denoising Autoencoding Model [J]. Modern Electronics Technique, 2015, 38(6): 20-24.

[12] Wu Z, Takaki S, Yamagishi J. Deep Denoising Auto-encoder for Statistical Speech Synthesis [OL]. arXiv: 1506.05268, 2015.

[13] Li J, Struzik Z, Zhang L, et al. Feature Learning from Incomplete EEG with Denoising Autoencoder [J]. Neurocomputing, 2015, 165: 23-31.

[14] Hu Shuai, Yuan Zhiyong, Xiao Ling, et al. Stacked Denoising Autoencoders Applied to Clinical Diagnose and Classification [J]. Application Research of Computers, 2015, 32(5): 1417-1420.

[15] Liu Kan, Yuan Yunying. Short Texts Feature Extraction and Clustering Based on Auto-Encoder [J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2015, 51(2): 282-288.

[16] Qin Shengjun, Lu Zhiping. Research of Unbalance Sentiment Classification Based on Denoising Autoencoders [J]. Science Technology and Engineering, 2014, 14(12): 232-235.

[17] Bengio Y, Delalleau O. On the Expressive Power of Deep Architectures [C]. In: Proceedings of the 22nd International Conference on Algorithmic Learning Theory. Springer Berlin Heidelberg, 2011: 18-36.

[18] Vincent P, Larochelle H, Bengio Y, et al. Extracting and Composing Robust Features with Denoising Autoencoders [C]. In: Proceedings of the 25th International Conference on Machine Learning. ACM, 2008: 1096-1103.

[19] Neural Networks and Deep Learning [EB/OL]. [2015-12-23]. http://neuralnetworksanddeeplearning.com/ch

[20] Vapnik V N. The Nature of Statistical Learning Theory [J]. IEEE Transactions on Neural Networks, 1995, 10(5): 988-999.

[21] NLPIR Chinese Word Segmentation System [EB/OL]. [2015-09-22]. http://ictclas.nlpir.org/.

[22] Text Categorization Corpus (Fudan) Test Corpus [EB/OL]. [2015-12-24]. http://www.nlpir.org/?action-viewnews-itemid-103.

Author Contributions

Liu Hongguang: Conceived the research idea and designed the study protocol.
Ma Shuanggang: Collected, cleaned, and analyzed data; selected appropriate algorithms for experiments.
Liu Guifeng: Verified experimental feasibility, organized experimental data, and summarized the experiments.

Conflict of Interest Statement

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] Liu Hongguang, Ma Shuanggang, Liu Guifeng. Experimental dataset used.rar. Dataset of 6 categories selected from the text categorization corpus

test corpus.
[2] Liu Hongguang, Ma Shuanggang, Liu Guifeng. InfoGain.txt. Feature terms selected by the information gain algorithm.
[3] Liu Hongguang, Ma Shuanggang, Liu Guifeng. TF-IDF values.rar. Calculated TF-IDF values for each text.
[4] Liu Hongguang, Ma Shuanggang, Liu Guifeng. Classification results.xlsx. Classification results computed by each algorithm.

Manuscript Received: 2016-01-13
Revised Manuscript Received: 2016-02-19

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

Source: ChinaXiv —Machine translation. Verify with original.