Sustainability **2015**, *7*, 2338-2352; doi:10.3390/su7032338

OPEN ACCESS

sustainability

ISSN 2071-1050

www.mdpi.com/journal/sustainability

Article

SAW Classification Algorithm for Chinese Text Classification

Xiaoli Guo [1], Huiyu Sun [1], Tiehua Zhou [2], Ling Wang [1,*], Zhaoyang Qu [1] and Jiannan Zang [1]

[1] School of Information Engineering, Northeast Dianli University, Jilin 132012, China;
E-Mails: gxl@mail.nedu.edu.cn (X.G.); Emailforu.001@gmail.com (H.S.);
qzywww@mail.nedu.edu.cn (Z.Q.); ddbear126@126.com (J.Z.)

[2] Database/Bioinformatics Laboratory, Chungbuk National University, Chungbuk 362-763, Korea;
E-Mail: thzhou@dblab.chungbuk.ac.kr

* Author to whom correspondence should be addressed; E-Mail: smile2867ling@gmail.com;
Tel./Fax: +86-0432-6480-6367.

Academic Editor: Jason C. Hung

Received: 14 August 2014 / Accepted: 1 February 2015 / Published: 27 February 2015

Abstract: Considering the explosive growth of data, the increased amount of text data's effect on the performance of text categorization forward the need for higher requirements, such that the existing classification method cannot be satisfied. Based on the study of existing text classification technology and semantics, this paper puts forward a kind of Chinese text classification oriented SAW (Structural Auxiliary Word) algorithm. The algorithm uses the special space effect of Chinese text where words have an implied correlation between text information mining and text categorization for high-correlation matching. Experiments show that SAW classification algorithm on the premise of ensuring precision in classification, significantly improve the classification precision and recall, obviously improving the performance of information retrieval, and providing an effective means of data use in the era of big data information extraction.

Keywords: big data; SAW classification algorithm; relevance

1. Introduction

With the rapid development of information technology, all kinds of data information are growing rapidly. The research results of the International Data Corporation (IDC) show that the global data volume reached 1.2 ZB in 2010 (the number is as high as 1.82 ZB in 2011), and is expected to reach

40 ZB in 2020, of which text data accounts for about 80%, thus how to effectively manage text information, and solving problems, such as the development of automatic text classification technology, are emerging research topics. Text classification technology has been widely used in daily data management applications [1,2]; it can realize automatic document classification, such as spam filter, site index systems, literature retrieval, user intent analysis, *etc.* Automatic text classification technology can achieve effective classification and extraction of text data; at the same time, it can improve the utilization rate of text data and precision of retrieval, and so on. Data volume is growing exponentially [3–5], thus, automatic text classification technology is particularly important in data mining and web mining research fields.

At present, the core of text classification research work mainly revolves around the text representation and classification model. For a long time, the primary method of text representation directly used the VSM (Vector Space Model), or variations based on VSM. Compared with the work of text representation and classification model, there is relatively more research on VSM, mainly focused on the introduction and improvement on the related research results of the machine learning field. In VSM, the text is converted into vector or mechanical text matching to realize classification. Although the classification performance and usability are better than previous knowledge engineering methods, there are still some problems, such as slow classification speed and precision. This is because most text classification methods are a simple classification from the perspective of text matching, but neglect the practical significance of the word itself or the semantics, leading to low text matching efficiency.

In order to solve this problem, this paper proposed a novel text classification algorithm according to the characteristics of Chinese grammar—SAW (Structural Auxiliary Word) classification algorithm, based on Chinese text classification. The algorithm introduces the theory of semantics into the weight of the existing algorithm, combined with the machine understanding of natural language, based on the factors affecting text categorization at its base. When studying the features of Chinese grammar to find meaning of a text, it is important to dig deeper into the text information in order to improve the precision of text classification and correlation of retrieval. In this paper, through four text-classification experiments, we compare the proposed SAW classification algorithm experiment with three aspects of classification. We classify the SAW algorithm has having high classification efficiency, precision, recall and F1-measure evaluation. The first experiment used the traditional support vector machine (SVM) classification algorithm; the second experiment used KNN classification algorithm based on clustering; and the third experiment used a combining rough set theory and ensemble learning based semi-supervised theory of text classification algorithm. The four experiments will be described in detailed in part 4.

2. Related Work

Usually, the text classification process including pre-processing, statistics, feature extraction and classifier training steps; depending on the classifier, these can be roughly divided into three kinds of methods: word matching method, knowledge engineering method, and the method of statistical learning. Classic decision trees include K nearest neighbor (KNN) [6], support vector machine (SVM), Bayes algorithm, neural network [7], rough sets, *etc.* Now it is generally believed that K nearest neighbor (KNN) method and support vector machine (SVM) method are more suitable for text categorization, and support vector machine (SVM) method has the highest classification precision.

In recent years, research focused on how to improve classic text categorization performance classification algorithms, and a lot of research results have been achieved. Shi *et al.* [8] proposed a novel semi-supervised classification algorithm based on tolerance rough set and ensemble learning. Based on the analysis of tow parallel proximal support vector machine (PSVM) algorithm-model algorithm and Cascade SVMs algorithm, Zhang *et al.* [9] proposed an improved parallel support vector machine classification algorithm (IPSVM). Zhang *et al.* [10] introduced the Map Reduce model into the Bayesian classification algorithm to improve the classification speed and solve the suiting problem of traditional Bayesian classification algorithm for large-scale data. Nedungadi *et al.* [11] proposed a standard Principal Component Analysis (PCA) and k nearest neighbor (KNN) hybrid algorithm for text classification, which reduced the computational complexity and maintained similar classification accuracy. Zhao *et al.* [12] proposed a gLDA algorithm by introducing distribution parameter of topic category to improve performance of Latent Dirichlet Allocation (LDA).

For text classification technology applied in different fields encountering various problems, many scholars also give corresponding solutions. Gupta *et al.* [13] proposed a novel ranking mechanism to rank the web documents that consider both the HTML structure of a page and the contextual senses of keywords that are present within it and its back-links. Colace *et al.* [14] proposed a single label text classification method that performed better than baseline methods when the number of labeled examples is small. Xu *et al.* [15] presented a web page classification algorithm—Link Information Categorization (LIC)—to solve the traditional classification algorithms based on the analysis of web content that cannot implement effective classification. In order to classify a web page as being benign or malicious, Hwang *et al.* [16] designed the system of 14 basic and 16 extended features that heuristically combined two basic features into one extended feature in order to effectively distinguish benign and malicious pages.

The above-described classification algorithms improve the classification performance, but there is also a common problem: these methods are starting from the angle of how to improve the efficiency of the traditional text classification method, but did not consider the characteristics of the language itself; there is no unique role for a special word statement. These special words are usually in the statement to the language logic function, which makes the performance limits. Therefore, by the integrated use of traditional text classification algorithm and special words, this paper puts forward the SAW classification algorithm. An algorithm with text classification constraints, greatly improves the accuracy and the categorization precision of text.

3. SAW Classification Algorithm

3.1. Algorithm Idea

The studying shows that the character “的” (“of’, it is equivalent to the Chinese character “的”) often occurs in vernacular Chinese language, especially in the title of journal papers. The word “的” is a cut-off point; the title is divided into two parts, in the front of “的” is called “prefix word” and at the back of the “的” is called “suffix word”. The paper titles have one common feature, which is the prefix words are usually the same, but the suffix words are different, and between these suffix words have a certain links, such as coordinating relation or inclusion relation. An example is shown in Table 1.

Table 1. Data sample

	Chinese	**English**
Paper Tiles	一种基于KNN的文本分类算法	A text classification algorithm based on KNN
	基于KNN的文本分类模型研究	The research of the text categorization model based on KNN
	基于KNN的快速WEB文档分类	A fast document classification based on KNN
	基于KNN的不良文本过滤方法	Reactionary text filtering method based on KNN

To show the feature of those paper titles more clearly, the collate data from Table 1 is presented in Figure 1. Owing to the remarkable differences between Chinese and English, the Chinese word “的” cannot easily be expressed in English. In Figure 1, it can be seen that the prefix words are the same, however, the suffix words are different; but the suffix words all are about text categorization. It shows that those paper titles belong to the same category. Therefore, in this paper, we use the unique role of the word “的” in Chinese for text classification, thereby mining the potential relation among texts to improve the accuracy and efficiency of the text classification and information retrieval.

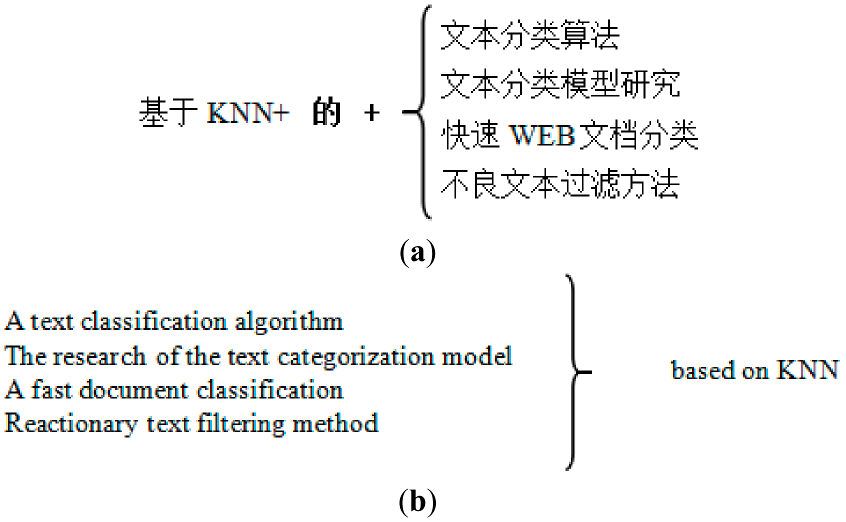
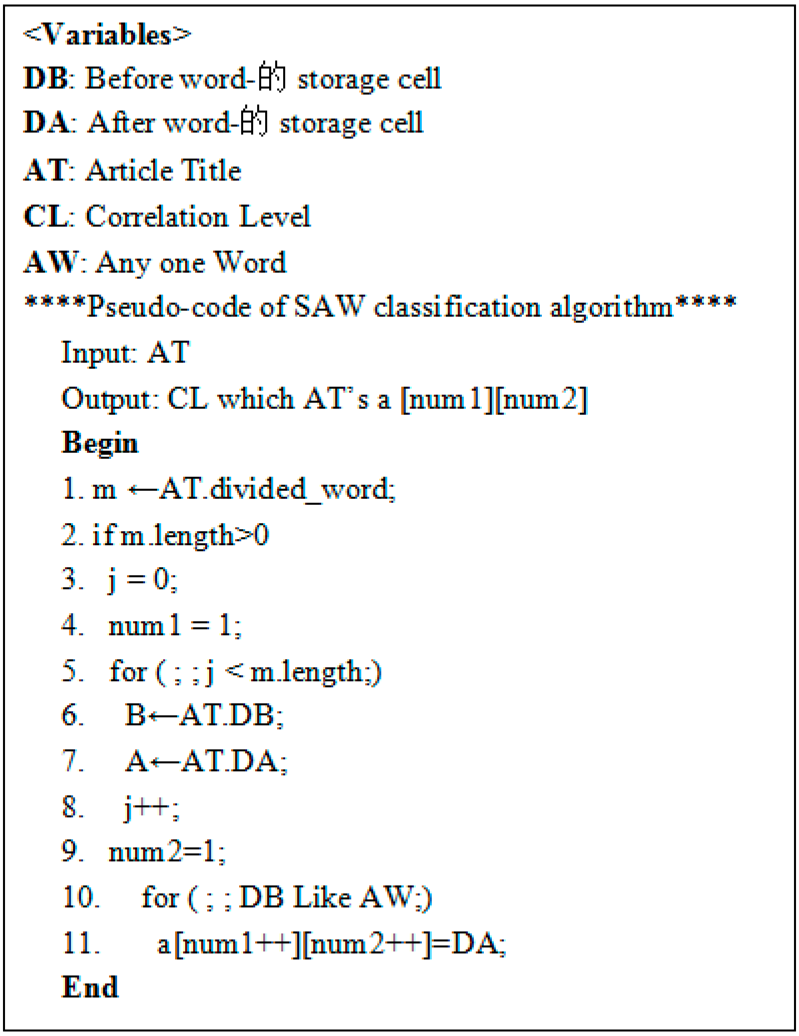

基于 KNN+ 的 +
文本分类算法
文本分类模型研究
快速 WEB文档分类
不良文本过滤方法
(a)
A text classification algorithm
The research of the text categorization model
A fast document classification
Reactionary text filtering method
based on KNN
(b)

Figure 1. The data sample of collating.

3.2. Process of Classification

In this section, the proposed SAW classification algorithm is presented. The algorithm consists of three key steps: Pre-processing, Calculating entry weighting, and designating relevance weighting to the SAW-Model.

Pseudo-code of SAW classification algorithm is shown below.



```
<Variables>
DB: Before word-的 storage cell
DA: After word-的 storage cell
AT: Article Title
CL: Correlation Level
AW: Any one Word
****Pseudo-code of SAW classification algorithm****
  Input: AT
  Output: CL which AT's a [num1][num2]
  Begin
  m ←AT.divided_word;
  if m.length>0
    j = 0;
    num1 = 1;
    for ( ; ; j < m.length;)
      B←AT.DB;
      A←AT.DA;
      j++;
    num2=1;
      for ( ; ; DB Like AW;)
        a[num1++][num2++]=DA;
  End
```

Figure 2. Pseudo-code of SAW classification algorithm

3.2.1. Pre-Processing

Pre-processing is a common first stage of text classification, but is performed differently from other classification algorithms. Pre-processing of the proposed SAW classification algorithm is done as follows:

Step 1. The first step is to assign ID for entries of the data sets.

Step 2. Using the corresponding word segmentation algorithm to separate the Chinese data sets. In this paper, using a dictionary mechanism for Chinese word segmentation based on Hash, which consists of the first character hash indexing, word indexing and dictionary. The structure of the dictionary is in Figure 1. The hash function is configured as follows:

$$Hash = (HB - 176) \times 100 + (LB - 160) \quad (1)$$

where *HB* is the high byte of the corresponding machine code and the *LB* is the low byte.

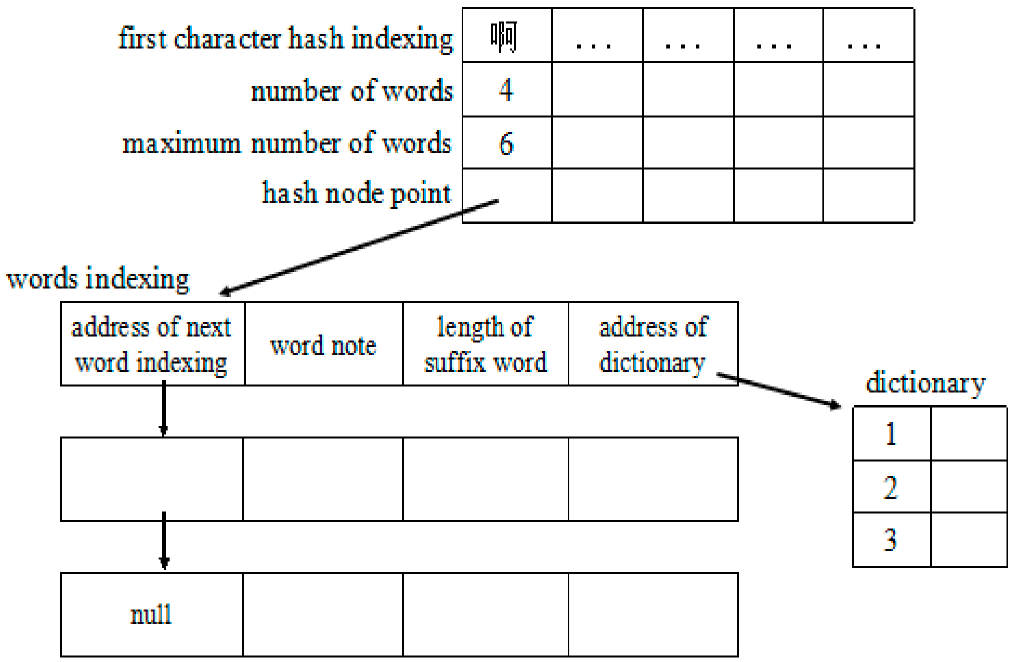


Figure 3. Structure of dictionary.

The character hash indexing consists of the number of words, hash node point and the maximum number of words permitted. The number of words is the number of the words designated the prefix; hash node point is used to store the address of the next node; the maximum number of words permitted is the maximum length of a string. The word indexing consists of the word node, the length of the suffix, the address of the dictionary and the address of next word indexing. The length of dictionary is equal to the number of suffix words that have same length and the dictionary is stored as the sequence list.

Step 3. In practice, the text is useful to classify accounts for only a small part of the word and most of the word has nothing to do with the class we are going to discriminate, and belongs to the "noise" word classification. Noise words will largely influence the text vector, including angle between the numerical and will flood useful information, leading to low precision of classification method. So, it is necessary to remove noise words. In this paper we redefine stop word to omit the most common words, such as prepositions and conjunctions to reduce the data dimension and the amount of calculation.

3.2.2. Calculating Entries Weighting

This stage calculates entries weighting, which are sorted by size, and it is done as follows:

Step 1. Applying the TFIDF (term frequency–inverse document frequency) weighting scheme to calculate data texts' vector. TFIDF is a statistical method for assessing the entry for a set of files or a corpus of a document by the degree of importance. The TFIDF algorithm calculates each class text vocabulary weights, sort these words, and gets the weight sort table.

TF-IDF computation formula is defined as follows.

$$tf_{i,j}idf_i = tf_{i,j} \times \left(\log |L| / |\{j : i \in L\}|\right) \tag{2}$$

where $tf_{i,j}$ is the number of times i appears in j; $|L|$ is corpus files in total; and $|\{j : i \in L\}|$ is the number of documents L in which i appears.

Step 2. After the calculations are complete, each word is re-sorted according to its weight to get the weight sort table. Also, since excessive dimension feature vectors will bring a huge amount of computation and other issues, reduction of the text feature vector dimension is necessary. The maximum weighting value K words is selected before the word is characterized, such as text feature vector. Using these feature vectors words can be searched with the training set and is represented as a vector of entry, such that it is easy to calculate semantic relatedness between them.

Step 3. Use the semantic relevancy algorithm to calculate and store entries by entries relatedness degrees. Assuming *A* and *B* are different entries, using the above formula, the weight of *A* and *B* can be calculated, and is represented in the form of *A* and *B* vector; x_i is a weight of any word in the vector space. *A* and *B* are expressed as:

$$A = \{A_1,\ A_2,\ \dots,\ A_K\}$$
$$B = \{B_1,\ B_2,\ \dots,\ B_K\}$$

Cosine relevancy between the two vectors, using the formula of cosine similarity is calculated as follows.

$$\cos(A,B) = \frac{A \times B}{|A| \times |B|} \tag{3}$$

Cosine similarity formula is a mathematical method to show the relevancy between the different entries. When the value is close to 1, the two entries have greater relevancy and when the value is close to 0, the two entries have lesser relevancy.

3.2.3. Relevance Weighting Model—SAW-Model

In this paper, the relevance-weighting SAW-Model is the core, and the paper titles are named entries. In this stage, the relevance-weighting SAW-Model, based on relevancy between words in different conditions, was readjusted to improve the relevancy of the weight-ranking table, to get the final classification results.

Using the above basic calculations, the text vector's size, angle, and numerical difference from other vectors can be calculated. The fundamental reason is that general text categorization algorithm does not mine deeper relations between texts, so corresponding revisions in the relevance-ranking table must be made.

The relevancy between the words that appear in the same entry is named the direct relevancy; direct relevancy is usually very high. This is defined as the first level of relevancy in this paper. The relevancy between words that present as common prefix or suffix word is named indirect relevancy, which is defined as the second level of relevancy. Therefore, in the proposed SAW classification algorithm, the relevancy were adjusted to meet the conditions for relevancy between entries, but with consideration to space and time costs, third level and above relevancy is ignored. In order to reflect the importance of the relevance between entries in the same class, this research uses clustering theory to improve the relevance between entries in the same class.

3.2.3.1. The First Level of relevancy

To the words that meet the first level of relevancy, add direct relevancy correction factor *α* on the relevancy value between those words. For any word *t* in the data sets, *t₁* is any word that is contained with *t* in the same entry, then the direct relevancy correction factor *α* is defined as follows:

$$\alpha = \frac{N_{\mathrm{tt1}}}{N} \cdot \max\{relevancy\} \tag{4}$$

where N_{tt1} is the number of entries that contain *t* and t_1 at the same time; *N* is the total number of entries in the data sets; and max relevancy is the maximum relevancy level in the ranking table.

The direct relevancy correction factor *α* reflects the relevancy between a word and words with it in direct relevancy. This paper, using *α* as the standard of correction relevance ranking tables, by the adjustment *α* realizes a high performance of text classification.

3.2.3.2. The Second Level of Relevancy

To the words that meet the second level of relevancy, add the second level of relevancy correction value IC_t to the relevancy value between those words. For the any word t in the data sets, the second level relevancy correction value IC_t is defined as follows:

$$IC_t = \frac{N_{\mathrm{t}}^{'}}{N_{\mathrm{t}} + N_{\mathrm{t}}^{'}} \times \alpha \tag{5}$$

where N_t is the number of entries that contained t; and $N_{\mathrm{t}}^{'}$ is the number of entries that exist in the same entry with the prefix word or suffix word of word *t* in the data sets.

3.2.3.3. Same Class Relevancy Correction Value V_s

For the entries in the same class, add the same class relevancy correction value V_s to the relevancy value between those entries in the same class. For any class *S* in the data sets, *m* is the total number of entries in the category *S*, and the same class relevancy correction value V_s is defined as follows:

$$V_{\mathrm{S}} = \alpha \frac{m_{\mathrm{t}}}{m} \tag{6}$$

where m_t is the number of word t in the category *S*.

The Same class relevancy correction value V_s reflects word relevancy with each class by clustering analysis to obtain high performance text classification.

The biggest difference between SAW classification algorithm and the previous classification algorithm is that the algorithm is weighted by relevance weighting SAW-Model sort of text-related degrees, so that when the text is retrieved, it is easy to find the greatest relevancy with the search entry text.

4. Experiment and Analyses

The experiments used titles of published papers as the experimental data, which are generally specialized terminology and have less noise data. Paper titles are highly correlated in the same professional field and the boundary between different fields is obvious, so to test the algorithm in the experimental section, titles of the papers were selected as the experimental data.

The experimental data is from China National Knowledge Infrastructure. From a total of 50,000 paper titles, the training sets have 7 categories with about 40,000 titles, and the test sets include 10,000 titles, 7 categories, including art (A), biology (B), computer sciences (C), economics (E), geography (G), history (H) and literature (L). Table 3 shows the samples in each category.

Table 2. Experimental data.

Category	Training Sets	Testing Sets
art (A)	6000	1500
biology (B)	6000	1200
computer (C)	5500	1500
economy (E)	6000	1500
geography (G)	5000	1500
history(H)	6000	1600
literature(L)	5500	1200
sum	40,000	10,000

4.1. Performance Measure

Generally accepted measures to evaluate the performance of text classification were used in this research to test the algorithim, namely precision and recall.

Assuming:

CC is the number of documents that were classified to the correct category;

RC is the number of documents that were rejected from the correct category;

RI is the number of documents that were rejected from the incorrect category; and

CI is the number of documents that were classified to the incorrect category,

then the precision and recall are defined as follows:

$$precision = \frac{CC}{CC + RI} \tag{7}$$

$$recall = \frac{CC}{CC + CI} \tag{8}$$

From the above formulas, it can be seen that the *precision* and *recall* reflect the quality of text classification algorithm from the two different aspects, and that *precision* and *recall* have a ttrade off relationship; both are very important, therefore, there is a composite index—*F1-measure*. The formula is defined as follows:

$$F1\text{-}measure = 2 \times \frac{precision \times recall}{\left(precision + recall\right)} \tag{9}$$

To evaluate the effectiveness of the SAW classification algorithm, three experiments were designed to compare with the SAW classification algorithm. The first experiment used the traditional support vector machine (SVM) classification algorithm, the second experiment used KNN classification algorithm based on clustering and the third experiment used a combining rough set theory and ensemble learning based semi-supervised theory of text classification algorithm. The above three experiments used the same testing data, and, respectively, calculated their categorization *precision*, *recall,* and *F1-measure*.

4.2. The Experimental Results

4.2.1. The First Experiment

Generally, the SVM classification algorithm had the best classification performance over any other traditional classification algorithm. Therefore, the first experiment is based on SVM classification algorithm and the results are shown in Table 3. It can be seen that the *precision* is below 0.86, the *recall* is below 0.84 and the *F1-measure* are all bellow 0.84.

Table 3. The first experimental results.

Category	*Precision*	*Recall*	*F1-Measure*
art (A)	0.853	0.809	0.830
biology (B)	0.821	0.797	0.809
computer (C)	0.776	0.825	0.800
economy (E)	0.804	0.813	0.808
geography (G)	0.812	0.796	0.804
history(H)	0.796	0.833	0.814
literature(L)	0.753	0.796	0.773
average	0.802	0.810	0.806

4.2.2. The Second Experiment

In this experiment, using a KNN classification algorithm based on clustering, which clustered by K-means clustering, and introduced weight value to indicate the different importance of each data set [17]. The results are shown in Table 5. It can be seen that the average *precision* is 0.849, the average *recall* is 0.842 and the average *F1-measure* is 0.846. Compared to the results in Tables 4 and 5, it is clear that the KNN classification algorithm based on clustering is better than the traditional SVM classification algorithm on the classification performance.

Table 4. The second experimental results.

Category	*Precision*	*Recall*	*F1-Measure*
art (A)	0.891	0.851	0.871
biology (B)	0.863	0.839	0.851
computer (C)	0.832	0.841	0.836
economy (E)	0.846	0.822	0.834
geography (G)	0.876	0.813	0.843
history(H)	0.824	0.871	0.847
literature(L)	0.811	0.858	0.834
average	0.849	0.842	0.846

4.2.3. The Third Experiment

In this experiment, using a combining rough set theory and ensemble learning based semi-supervised theory of text classification algorithm [8], which first generated the reliable negative data set, and then used an ensemble classifier by adopting SVM, Rocchio, NB as base classifiers to implement labeling the negative data set from the unlabeled data set. The *precision*, *recall* and *F1-measure* of the text classification algorithm are shown in Table 6. It can be see that *precision* scores are all below 0.88, *recall* are all below 0.90, and *F1-measure* are all below 0.88. Comparing the results in Table 5 with those Tables 3 and 4, it is clear that the recall of the combining rough set theory and ensemble learning based semi-supervised theory of text classification algorithm is best of all.

Table 5. The third experimental results.

Category	*Precision*	*Recall*	*F1-Measure*
art (A)	0.875	0.883	0.879
biology (B)	0.847	0.875	0.861
computer (C)	0.809	0.884	0.845
economy (E)	0.822	0.871	0.846
geography (G)	0.843	0.867	0.855
history(H)	0.811	0.894	0.850
literature(L)	0.792	0.887	0.837
average	0.828	0.880	0.854

4.2.4. The Experiment of SAW Classification Algorithm

The results from the tests using the SAW classification algorithm this paper proposed, is shown in Table 6.

Table 6. The experimental results of SAW classification algorithm.

Category	*Precision*	*Recall*	*F1-Measure*
art (A)	0.946	0.923	0.934
biology (B)	0.928	0.921	0.924
computer (C)	0.926	0.907	0.916
economy (E)	0.900	0.909	0.904
geography (G)	0.916	0.931	0.923
history(H)	0.918	0.928	0.923
literature(L)	0.902	0.911	0.906
average	0.919	0.918	0.919

Figure 4 shows the relationship of precision with every category in the experiment of SAW classification algorithm with the other three experiments. It shows clearly that the precision in SAW classification algorithm experiment is higher than other three experiment.

Figure 5 shows the relationship of recall with every category in the experiment of SAW classification algorithm with the other three experiments. It shows clearly the SAW classification algorithm has a high recall, which had a stable level.

F1-measure could reach a compromise between precision and recall. Figure 6 shows the *F1-measure* of SAW classification algorithm compared with the other three experiments. It shows clearly that the F1-measure in experiment of SAW classification algorithm is higher than the other three experiments. It indicates that the proposed algorithm has a good performance of classification.

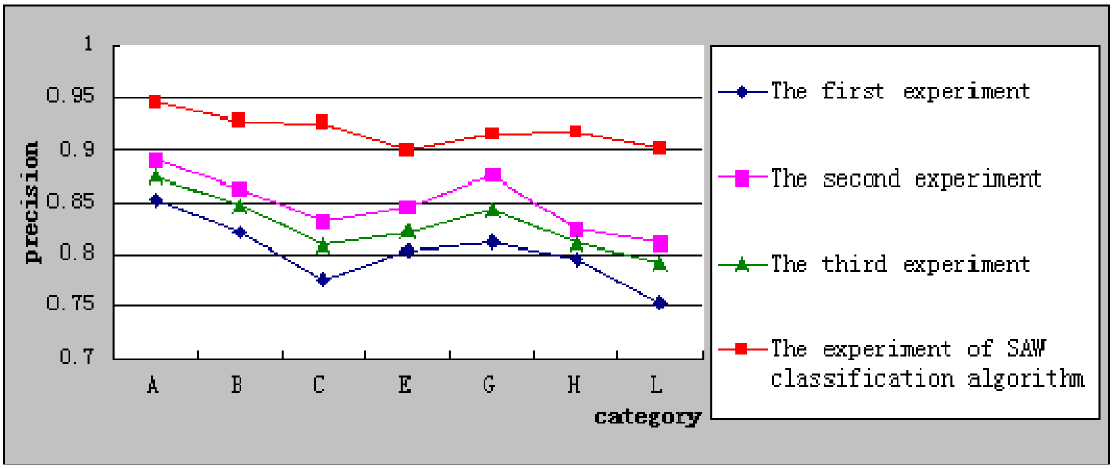


Figure 4. Comparison of precision.

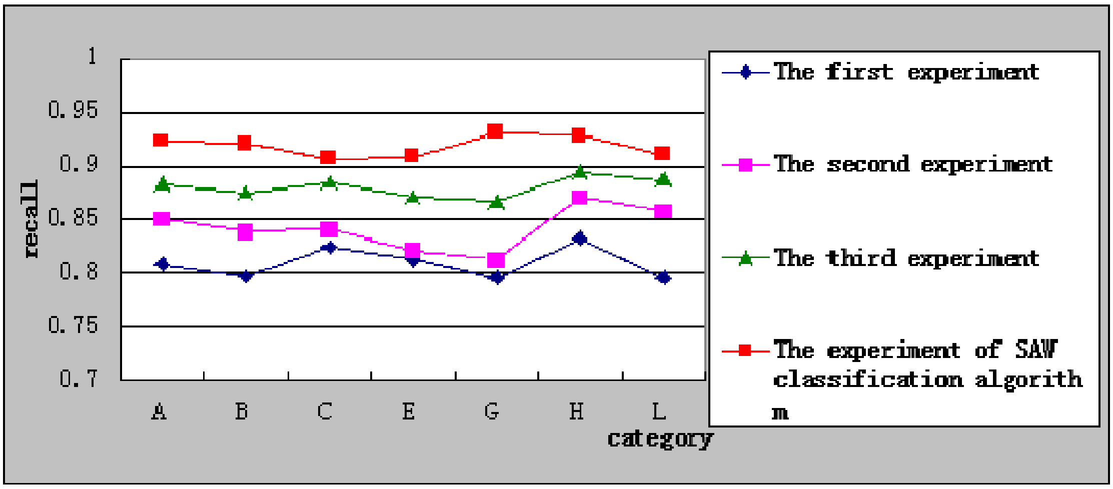


Figure 5. Comparison of recall.

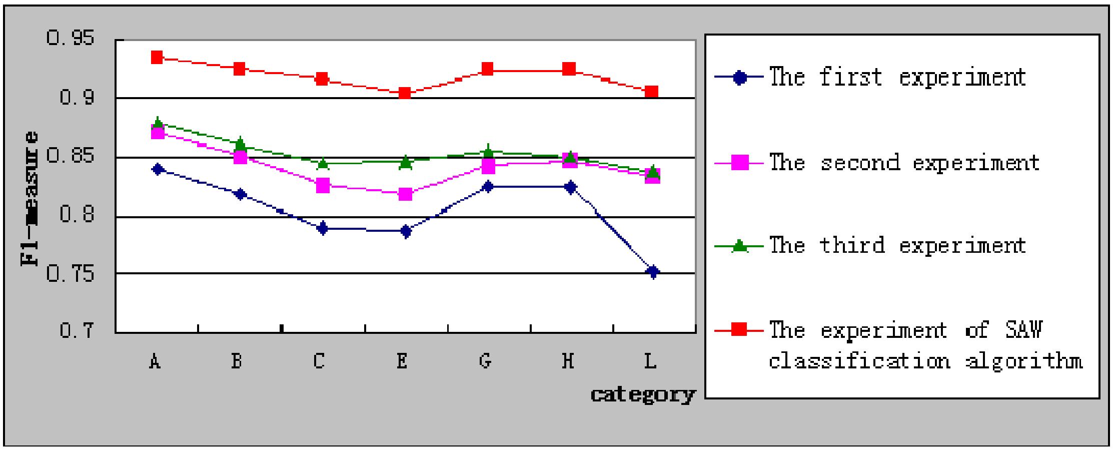


Figure 6. Comparison of F1-measure.

4.3. Analysis of Experimental Results and Thinking

(1) SAW classification algorithm used a dictionary mechanism for Chinese words segmentation based on Hash to segment entries, and used the paper title from China National Knowledge Infrastructure as the experimental data sets in the experiments; the boundaries between different categories are distinct. The experimental results shows the average precision of classification is above 0.90, verifying the proportion of documents that are classified to the correct category is high by SAW classification algorithm.

(2) SAW classification algorithm mine implicit information between texts via special words, which effectively improves the value of recall, which shows text recognition ability of classification algorithm. In the experiments, the recall of SAW classification algorithm is above 91%, and is maintains a stable value, which shows the performance of SAW classification algorithm is not affected by different data sets, indicating that SAW classification algorithm has a higher performance of classification than the other three classification algorithms.

(3) Using the results of the classification precision and recall to calculate F1-measure of the algorithm, the experimental results show that the SAW classification algorithm is better than the other three classification algorithms in all respects.

In summary, the proposed SAW classification algorithm combines text classification based on statistical learning methods and semantics research features, gives full play to the advantages of a combination of both. Experiments show, SAW classification algorithm has a good performance of classification.

5. Conclusions

Research on the traditional classification algorithms and some improved classification algorithms, this paper proposed a novel classification algorithm—SAW classification algorithm that used the novel designing ideas and combined the traditional classification algorithm, by omitting the most common words to reduce the data dimension, adjusting the relevancy between the entries with different conditions and using clustering theory to improve the relevance between entries in the same class. In the experiments, the proposed SAW classification algorithm experiment is compared with the other three text classification experiments using data sets from China National Knowledge Infrastructure using three aspects of classification precision, recall and F1-measure evaluation. The experimental results show the SAW classification algorithm has better classification efficiency than the other three classification algorithms. This is mainly because the SAW classification algorithm mined deeper correlations between the text, and used some words in particular to implement higher performance and efficiency text classification.

The research results of the proposed algorithm can be applied to the classification of all kinds of document libraries, improving the precision in classification and retrieval. Text resources to facilitate efficient management of large data can effectively solve the problem of big data text resources retrieval performance. In future research, we plan to explore the factors that deeper influence the performance of text classification through more experiments to improve the efficiency of text classification algorithm.

Acknowledgments

Foundation item: This work was supported by the National Natural Science Foundation of China (No. 51077010) and by the science and technology plan projects of Jilin City (No. 201464059).

Author Contributions

Xiaoli Guo is responsible for designing algorithm, Huiyu Sun is responsible for implementation of algorithm, TieHua Zhou and Ling Wang are responsible for experiment, Zhaoyang Qu and Jiannan Zang are responsible for editing the language.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Elsayed, E.; Eldahshan, K.; Tawfeek, S. Automatic evaluation technique for certain types of open questions in semantic learning systems. *Hum.-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-19.
2. Sarkar, K. Automatic single document text summarization using key concepts in documents. *J. Inf. Process. Syst.* **2013**, *9*, 602–620.
3. Kim, J.S.; Byun, J.; Jeong, H. Cloud AEHS: Advanced learning system using user preferences. *J. Converg.* **2013**, *4*, 31–36.
4. Cho, Y.S.; Moon, S.C. Weighted Mining Frequent Pattern based Customer's RFM Score for Personalized u-Commerce Recommendation System. *J. Converg.* **2013**, *4*, 36–40.
5. Malkawi, M.I. The art of software systems development: Reliability, Availability, Maintainability, Performance (RAMP). *Hum.-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-22.
6. Hong, S.; Chang, J. A new k-NN query processing algorithm based on multicasting-based cell expansion in location-based services. *J. Converg.* **2013**, *4*, 5–10.
7. Gopalakrishnan, A.K. A subjective job scheduler based on a backpropagation neural network. *Hum.-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-17.
8. Shi, L.; Ma, X.; Xi, L.; Duan, Q.; Zhao, J. Rough set and ensemble learning based semi-supervised algorithm for text classification. *Expert Syst. App.* **2011**, doi:10.1016/j.eswa.2010.11.069.
9. Zhang, S. An Improved Parallel SVM Algorithm for Chinese Text Classification. In Proceedings of the Second International Conference on Electric Technology and Civil Engineering, Washington, DC, USA, 18 May 2012.
10. Zhang, L.; Shao, T. Improved bayesian text classification algorithm in cloud computing environment. *Comput. Sci.* **2014**, *S1*, 339–342.
11. Nedungadi, P.; Harikumar, H.; Ramesh, M. A high performance hybrid algorithm for text classification. In Proceedings of the Fifth International Conference on Applications of Digital Information and Web Technologies (ICADIWT), Bangalore, India, 17–19 February 2014.

12. Zhao, D.; He, J.; Liu, J. An improved LDA algorithm for text classification. In Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering (ISEEE), Hokkaido, Japan, 26–28 April 2014.
13. Gupta, P.; Singh, S.K.; Yadav, D.; Sharma, A.K. An Improved Approach to Ranking Web Documents. *J. Inf. Process. Syst.* **2013**, *9*, 217–236.
14. Colace, F.; de Santo, M.; Greco, L.; Napoletano, P. Text classification using a few labeled examples. *Comput. Hum. Behav.* **2014**, doi:10.1016/j.chb.2013.07.043.
15. Xu, Z.; Yan, F.; Qin, J.; Zhu, H. A Web Page Classification Algorithm Based on Link Information. In Proceedings of the Tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), Wuxi, China, 14–17 October 2011.
16. Hwang, Y.S.; Moon, J.C. Classifying malicious web pages by using an adaptive support vector machine. *J. Inf. Process. Syst.* **2013**, *9*, 395–404.
17. Yong, Z.; Youwen, L.; Shixiong, X. An improved KNN text classification algorithm based on clustering. *J. Comput.* **2009**, *4*, 230–237.