Power Law for Text Categorization

Wuying Liu[1], Lin Wang[2], and Mianzhu Yi[1]

[1] PLA University of Foreign Languages, 471003 Luoyang, Henan, China
wyliu@nudt.edu.cn, mianzhuyi@gmail.com
[2] National University of Defense Technology, 410073 Changsha, Hunan, China
wanglin@nudt.edu.cn

Abstract. Text categorization (TC) is a challenging issue, and the corresponding algorithms can be used in many applications. This paper addresses the online multi-category TC problem abstracted from the applications of online binary TC and batch multi-category TC. Most applications are concerned about the space-time performance of TC algorithms. Through the investigation of the token frequency distribution in an email collection and a Chinese web document collection, this paper re-examines the power law and proposes a random sampling ensemble Bayesian (RSEB) TC algorithm. Supported by a token level memory to store labeled documents, the RSEB algorithm uses a text retrieval approach to solve text categorization problems. The experimental results show that the RSEB algorithm can achieve the state-of-the-art performance at greatly reduced space-time requirements both in the TREC email spam filtering task and the Chinese web document classifying task.

Keywords: Text Categorization, Power Law, Online Binary TC, Batch Multi-Category TC, TREC.

1 Introduction

Automated text categorization (TC) has been widely investigated since the early days of artificial intelligence. According to the arriving mode of documents, TC can be divided into online TC and batch TC. According to the number of predefined categories, TC includes binary TC and multi-category TC. For instance, email spam filtering is an online binary TC application and web document classifying is normally a batch multi-category TC application. The binary TC is a special case of the multi-category TC and a batch TC can be regarded as a series of online classifications, so this paper addresses the online multi-category TC problem.

Most TC applications pay more attention to the space-time complexity of TC algorithms. The power law of word frequency in a set of text documents, a famous random distribution phenomenon, was discovered for a long time. How to use the power law to reduce the space-time complexity of statistical TC algorithms is a significant research problem. In statistical TC algorithms, token frequency is a very effective feature. If we only use token frequency features in a closed text collections, the feature with once occurrence will never be used, and according to the power law,

M. Sun et al. (Eds.): CCL and NLP-NABD 2013, LNAI 8202, pp. 131–143, 2013.

we can easily remove these useless long tail features for lower space-time costs. But in an online situation, we meet a puzzle of open feature space. The ubiquitous power law may bring an opportunity to propose a novel statistical TC algorithm for the efficient online multi-category TC problem.

The rest of this paper is organized as follows. In section 2, we describe some related works about TC. In section 3, we investigate the power law of token frequency both in an email collection and a web document collection, and analyze the potential uselessness rate. In section 4, we propose a random sampling ensemble Bayesian (RSEB) algorithm. In section 5, the experiment and result are described. At last, in section 6, the conclusion and further work are given.

2 Related Work

Recently, statistical TC algorithms have been widely used in TC applications [1]. Email spam filtering is defined as an online supervised binary TC problem, which is simulated as an immediate full feedback task (IFFT) in the TREC spam track. Web document classifying is normally defined as a batch multi-category TC problem, which is simulated as a 12-category Chinese web document classifying task (WDCT) [2].

Many online binary TC algorithms have been proposed for the email spam filtering. For instance: 1) based on the vector space model (VSM), the online Bayesian algorithm uses the joint probabilities of words and categories to estimate the probabilities of categories for a given document; 2) the relaxed online support vector machines (SVMs) algorithm [3] relaxes the maximum margin requirement and produces nearly equivalent results, which has gained several best results in the TREC 2007 spam track; and 3) the online fusion of dynamic Markov compression (DMC) and logistic regression on character 4-gram algorithm [4] is the winner on the IFFT in the TREC 2007 spam track.

Many batch multi-category TC algorithms have been introduced to deal with the web document classifying. For instance: 1) the k-nearest neighbor (kNN) TC algorithm decides a document according to the k nearest neighbor document categories; 2) the centroid TC algorithm [5] is based on the assumption that a given document should be assigned a particular category if the similarity of this document to the centroid of its true category is the largest; and 3) the winnow algorithm [6] uses a multiplicative weight-update scheme that allows it to perform much better when many dimensions are irrelevant.

Structured feature and token frequency distribution feature of documents are both crucial to the classification performance. Previous research shows that the multi-field structured feature of email documents supports the divide-and-conquer strategy, and can be used to improve the classification performance [7]. This multi-field learning (MFL) framework will bring the statistical, computational and representational advantages like that of ensemble learning methods [8]. Previous research also shows that the token frequency distribution follows the power law [9], which is a prevalent random phenomenon in many text documents.

The previous TC algorithms often pursue the high classification accuracy and the high overall performance of supervised learning, without more claiming their low space-time complexity. However, in practice the algorithm is space-time-cost-sensitive for many real-world large-scale applications. For instance, specified in the TREC spam track, the space-time limitation (total 1 GB RAM and 2 sec/email) is still

unpractical and horrible in a real large-scale email system, where large-scale emails will form a round-the-clock data stream and there will be more than thousands of emails arriving during 2 seconds. Especially, it is unreasonable to require an industrial TC algorithm with a time-consuming training or updating: such a requirement defeats previous complex statistical algorithms, and motivates us to explore a space-time-efficient TC algorithm.

3 Re-examination of Power Law

3.1 Corpora

The email documents corpus is the TREC07p collection, firstly designed as a public corpus for TREC 2007 spam track, which contains total 75,419 emails (25,220 hams and 50,199 spams). Each email document is stored as a plain text file, and email text is unaltered except for a few systematic substitutions of names.

The Chinese web documents corpus is the TanCorp collection, which contains total 14,150 documents and is organized in two hierarchies. The first hierarchy contains 12 big categories and the second hierarchy consists of 60 small classes. In this paper, we use TanCorp-12.

From the perspective of lingual category, above two corpora are representative. The TREC07p corpus contains multi-language, although the main language is English. The TanCorp corpus is a Chinese text documents collection.

3.2 Token Frequency Distribution

In order to re-examine token frequency distribution, we calculate the number of tokens. According to the widely-used VSM, a text document is normally represented as a feature vector, and each feature is a text token. Previous research has shown that overlapping word-level k-gram token model can achieve promising results [10]. But different languages have different appropriate k values, and the different representational granularities determine the total number of text features. Here, we consider four overlapping word-level k-gram token models (1-gram, 2-gram, 3-gram, 4-gram) to represent tokens.

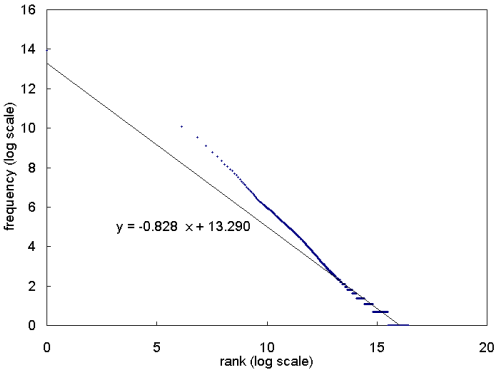

Fig. 1. Word-level 4-gram Token Frequency-Rank in the TREC07p Collection

Firstly, we regard an email message as a single plain-text document, and calculate the number of each token occurrence in the TREC07p collection. Fig. 1 shows the token frequency as the function of the token's rank with the word-level 4-gram token model. The horizontal-axis (x-axis) indicates the token's rank (log scale), and the vertical-axis (y-axis) indicates the token frequency (log scale). The trendline (y = ***a*** x + ***b***) indicates that the frequency distribution of the word-level 4-gram token approximately follows a power law.

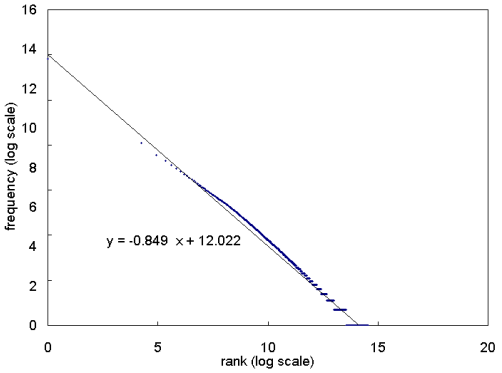

Fig. 2. Word-level 2-gram Token Frequency-Rank in the TanCorp Collection

Secondly, we use the same method to calculate the number of each token occurrence in the TanCorp collection. Fig. 2 shows the token frequency as the function of the token's rank with the word-level 2-gram token model. The trendline also shows a power law distribution.

Table 1. Trendline Coefficients in TREC07p Collection and TanCorp Collection

		1-gram	2-gram	3-gram	4-gram
TREC07p	***a***	-1.118	-1.103	-0.923	-0.828
	b	15.047	16.454	14.501	13.290
TanCorp	***a***	-1.766	-0.849	-0.460	-0.280
	b	20.129	12.022	6.879	4.242

Finally, the statistical results show that not only the 4-gram token frequency distribution in the TREC07p collection and the 2-gram token frequency distribution in the TanCorp collection follow the power law, but the others k-gram token frequency distribution also follow the power law. Table 1 shows the detailed trendline (y = ***a*** x + ***b***) coefficients ***a*** and ***b***.

Above re-examination shows that the token frequency distribution follows the power law in the multilingual email documents, the Chinese web documents, and the field sub-documents of email [11]. The ubiquitous power law indicates that the weightiness of each token feature is not equivalent, which suggests a feature selection method to remove those useless features for lower space-time costs.

3.3 Potential Useless Feature

In statistical TC algorithms, the text feature selection is a widely-used method against the high dimensional problem and has a crucial influence on the classification performance. The iteration, the cross-validation and the multi-pass scans are all effective methods to the text feature selection. But these methods bring the high space-time complexity. If we can detect and remove those useless features, we will save more time and space. However, what is the useless feature and how to find it?

In a whole text documents set, a token feature with a less frequency (≤2) is a potential useless feature. As an extreme instance, if a token feature occurs only once all the time, it is useless because it will never be used in the future. So the useful features will not decrease after removing the useless features. We further define the uselessness rate R_u as the ratio of the number of token features with less frequency to the total number of token features. Here, we only consider the word-level 4-gram token in the TREC07p collection and the word-level 2-gram token in the TanCorp collection.

Table 2. Feature Number and Uselessness Rate

	Feature Number (num)			R_u(%)	
	$N(1)$	$N(2)$	$N(*)$	$R_u(\leq 1)$	$R_u(\leq 2)$
TREC07p	9,985,998	2,885,917	15,754,699	63	82
TanCorp	1,316,422	325,834	2,087,815	63	79

The $N(1)$ and $N(2)$ separately denote the number of token features which only occurs once and twice in the related documents set. The $N(*)$ denotes the total number of token features. The $R_u(\leq 1)$ is defined as $N(1)/N(*)$, and the $R_u(\leq 2)$ is defined as $(N(1)+N(2))/N(*)$. Table 2 shows that the uselessness rate in the TREC07p collection is between 63% and 82%, and the uselessness rate in the TanCorp collection is between 63% and 79%. The uselessness rates are all higher in the five natural text fields of the TREC07p corpus [11]. If we can get the whole text documents before TC predicting, we will easily find these useless token features and cut this long tail. However, the online TC application faces an open text space problem, and we can not foreknow a token feature's occurrence in the future. Though an online text stream makes it impossible to find these posteriori useless token features, the higher uselessness rate indicates that there are lots of useless token features. Supported by the priori and ubiquitous power law, this paper proposes a random sampling method to remove these useless token features at the time of online training. The range of uselessness rate indicates the theoretical tolerant range of training feature loss rate.

4 Random Sampling Ensemble Bayesian Algorithm

4.1 Token Level Memory

In this paper, the object categories of the online multi-category TC problem are represented as a set in the form ($\boldsymbol{C}=\{C_i\}$, i=1, 2, ..., n), and a document $\boldsymbol{D}$ is represented as a sequence of tokens in the form ($\boldsymbol{D}=T_1T_2...T_j...$). Here, we use the overlapping word-level k-gram model to define a token. The token frequency within

historical labeled documents, the key feature of online supervised machine learning, implies rich classification information and must be stored effectively. The token level memory (TLM) is a data structure to store the token frequency information of labeled documents, from which we can conveniently calculate the Bayesian conditional probability $\boldsymbol{P}(C_i|T_j)$ for the object category C_i and the token T_j. We straightforwardly combine the Bayesian conditional probabilities of tokens and choose the category of the biggest probability as the document's final category prediction.

Fig. 3 shows the TLM structure, including two indexes organized as two hash tables. The table entry of the DF index is a key-value pair <key_C, *value*>, where each key C_i denotes the *i*th category and each value $DF(C_i)$ denotes the total number of documents with C_i category labels. The hash function $hash_{DF}(C_i)$ maps the category C_i to the address of the $DF(C_i)$. The table entry of the TF index is also a key-value pair <key_T, *value*>, where each key T_j denotes a token and each value consists of n integers. The integer $TF_i(T_j)$ denotes the occurrence times of the token T_j in labeled C_i category documents. The hash function $hash_{TF}(T_j)$ maps the token T_j to the address of the n integers.

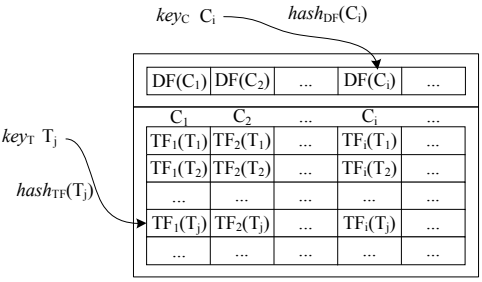

Fig. 3. Token Level Memory

The TLM stores labeled tokens, the tiny granularity labeled examples, while other memory-based algorithms, such as kNN, store document-level labeled examples. This index structure has a native compressible property of raw texts. Each incremental updating or retrieving of index has a constant time complexity. The power law can help us to remove lots of long tail tokens through random sampling learning.

4.2 Random Sampling Learning

Supported by the TLM, the RSEB algorithm takes the online supervised training process as an incremental updating process of indexes, and takes the online predicting process as a retrieving process of indexes.

According to the power law, we add a random sampling learning into the online supervised training process. The random sampling idea is based on the assumption that some tokens selected randomly according to equiprobability trend to be higher frequency tokens. If only the relative frequency features are concerned among tokens, we can use partial tokens of a labeled document to update the TLM after random sampling. As a result, lots of long tail tokens will be online removed, and the relative frequency will not change among tokens. We define the random sampling rate R_{rs} as the ratio of the number of tokens added into the TLM to the total number of tokens of each labeled document, which is a real number ($R_{rs} \in [0, 1]$).

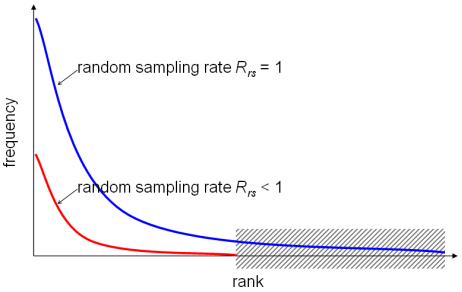

Fig. 4. Random Sampling Sketch

Fig. 4 shows the random sampling sketch. The horizontal-axis (x-axis) indicates the token's rank, and the vertical-axis (y-axis) indicates the token frequency. If R_{rs}=1, all the tokens of a labeled document will be added into the TLM at the time of online training. While if R_{rs}<1, there will be some tokens absent in the TLM. Along the online incremental updating, these above two cases will form two power law curves in Fig. 4, where the shadow range denotes removed tokens. These two power law curves also indicate that the random sampling will not change the total distribution of the relative frequency among tokens. Of course, if the random sampling rate approximates zero, the classification ability of the TLM will also be damaged. However, what is the optimal random sampling rate? Theoretically, a promising random sampling rate is the (R_{rs}=1-R_u). But the exact R_u is also not a priori value. Fortunately, the ubiquitous power law gives an approximate heuristic, such as the 20|80 rule of the $R_{rs}|R_u$.

```
//OTP: Online Training Procedure
OTP(Document d; Gram k; Category c; TLM t; Rrs r)
(1) String[] T := Tokenizer(d; k);
(2) String[] Trs := RandomSampler(T; r);
(3) t.DF(c) := t.DF(c)+1;
(4) Loop: For Each Tj∈Trs Do:
   (4.1) If t.containKeyT(Tj) Then: t.TF(c, Tj) := t.TF(c, Tj)+1;
   (4.2) Else:
      (4.2.1) t.TF(c, Tj) := 1;
      (4.2.2) t.TF(~c, Tj) := 0; //~c means all other categories
      (4.2.3) t.putKeyT(Tj);
(5) Output: t.

//Extract tokens based on overlapping word-level k-gram model
Tokenizer(Document d; Gram k)

//Sample tokens based on the random sampling rate Rrs
RandomSampler(String[] T; Rrs r)
```

Fig. 5. Pseudo-Code for Online Training

Fig. 5 gives the pseudo-code for the online training procedure of the RSEB algorithm. When a new labeled document arrives, the online training procedure only needs to add the document's tokens into the TLM. This procedure firstly analyzes the

document text and extracts tokens based on an overlapping word-level k-gram model, and then randomly samples the tokens based on a preset random sampling rate, and finally updates the token frequency or adds a new index entry to the TLM according to the tokens after the random sampling.

4.3 Ensemble Bayesian Predicting

The Bayesian conditional probability predicting is a very classical method. According to each observed token of a document, the Bayesian method can obtain an array of probabilities, reflecting the likelihood that the classified document belongs to each category. The ensemble method uses arithmetical average to combine the multi-array of probabilities predicting from all tokens to form a final array. And then, the category of the maximal probability in the final array is predicted as the document's category.

```
//OPP: Online Predicting Procedure
OPP(Document d; Gram k; TLM t)
(1) String[] T := Tokenizer(d; k);
(2) Float[] ep := new Float[n];
(3) Loop: For Each T_j ∈ T Do:
   (3.1) Float[] p := BayesianPredictor(T_j; t);
   (3.2) Loop: For Each i ∈ [1, n] Do:
      (3.2.1) ep[i] := ep[i]+p[i];
(4) Float sum := Sum(ep); //Add the n floats to a sum
(5) Loop: For Each i ∈ [1, n] Do:
   (5.1) ep[i] := ep[i]/sum;
(6) Integer index := Math.max(ep).getIndex;
(7) Output: C_index, ep[index].

//Compute conditional probability P(C_i|T_j) for each category C_i
BayesianPredictor(String token; TLM t)
(1) Float[] p := new Float[n];
(2) Loop: For Each i ∈ [1, n] Do:
   (2.1) p[i] := t.TF(C_i, token)/t.DF(C_i);
(3) Float sum := Sum(p); //Add the n floats to a sum
(4) Loop: For Each i ∈ [1, n] Do:
   (4.1) p[i] := p[i]/sum;
(5) Output: p.

//Extract tokens based on overlapping word-level k-gram model
Tokenizer(Document d; Gram k)
```

Fig. 6. Pseudo-Code for Online Predicting

Fig. 6 gives the pseudo-code for the online predicting procedure of the RSEB algorithm. When a new document arriving, the online predicting procedure is triggered: 1) the procedure also analyzes the document text and extracts tokens based on an overlapping word-level k-gram model; 2) the procedure retrieves the current TLM and calculates each token's probabilities array according to the Bayesian conditional probability; 3) the procedure assumes that each token's contribution is equivalent to the final probabilities array and uses the arithmetical average method to

calculate a final ensemble probabilities array; and 4) the procedure chooses the maximal probability in the final ensemble probabilities array, and outputs the document's category predication and this maximal probability.

4.4 Space-Time Complexity

The RSEB algorithm mainly makes up of the online training and the online predicting procedures, whose space-time complexity depends on the TLM storage space and the loops in the two procedures.

The TLM storage space is efficient owing to two reasons: the native compressible property of index files [12] and the random-sampling-based compressible property at the time of online incremental updating. Hash list structure, prevailingly employed in information retrieval, has a lower compression ratio of raw texts. Though the training documents will mount in the wake of the increasing of online feedbacks, the TLM storage space will only increase slowly. The native compressible property of index files ensures that the TLM storage space is theoretically proportional to the total number of tokens, and not limited to the total number of training documents. The random-sampling-based compressible property of TLM is caused by the power law of token frequency distribution and the only requirement of relative frequency. The random-sampling-based feature selection can cut the long tail useless features in the online situation. The above two compressible properties make that the online labeled document stream can be incrementally space-efficiently stored.

The incremental updating or retrieving of TLM has a constant time complexity according to hash functions. The online training procedure is lazy, requiring no retraining when a new labeled document added. Fig. 5 shows that the time cost of per updating is only proportional to the total number of tokens in the document. Except the loop (see (4) of Fig. 5) according to the number of tokens, there are no time-consuming operations. The major time cost of the online predicting procedure is related to the number of categories. The straightforward calculating makes that the time complexity is acceptable in the practical online application.

5 Experiment

5.1 Implementation

We implement an email spam filter (*esf*) and a web document classifier (*wdc*) according to the proposed RSEB algorithm. In the filter and classifier, we do nothing about text pre-processing, such as stemming, stop word elimination, etc.

Using *cs4* combining strategy [7], the *esf* filter is combined from seven field classifiers within the seven-field MFL framework, five natural fields (Header, From, ToCcBcc, Subject, and Body) and two artificial fields (H.IP, H.EmailBox), and each field classifier is an implementation of the RSEB algorithm with binary categories. In each field classifier, the overlapping word-level 4-gram token model is applied.

Applying the overlapping word-level 2-gram token model, the *wdc* classifier is an implementation of the RSEB algorithm with 12 categories in Chinese texts. In order to extract word-level tokens, we build a Chinese segmenter in the *wdc* classifier.

5.2 Task and Evaluation

We run an IFFT of email spam filtering and a WDCT of 12-category Chinese web document classifying to evaluate the performance of the RSEB algorithm.

On the email spam filtering, we report the overall performance measurement 1-ROCA, the area above the receiver operating characteristic (ROC) curve percentage, where 0 is optimal, to evaluate the filter's performance. We compare the *esf* to the *bogo* filter (bogo-0.93.4), the *tftS3F* filter, and the *wat3* filter on the IFFT, defined in the TREC spam track. The *bogo* filter is a classical implementation of online Bayesian algorithm, the *tftS3F* filter is based on the relaxed online SVMs algorithm and has gained several best results in the TREC 2007 spam track, and the *wat3* filter is based on the online fusion of DMC and logistic regression algorithm, which is the winner on the IFFT in the TREC 2007 spam track. In this experiment, we use the TREC07p corpus, the TREC spam filter evaluation toolkit, and the associated evaluation methodology.

On the web document classifying, we use three-fold cross validation in the experiments by evenly splitting the TanCorp-12 dataset into three parts and use two parts for training and the remaining third for testing. We perform the training-testing procedure three times and use the average of the three performances as the final result. Here reports classical MacroF1 and MicroF1 measures. We run the *wdc* classifier on the 12-category Chinese WDCT, and compare the results of the *wdc* classifier as well as to that of the *kNN* classifier, the *centroid* classifier, and the *winnow* classifier.

The hardware environment for running experiments is a PC with 1 GB memory and 2.80 GHz Pentium D CPU.

5.3 Results and Discussions

There are four experiments. On the email spam filtering, the experiment A tries to evaluate that the RSEB algorithm is time-efficient and can achieve the best overall performance, and the experiment B wants to verify that the TLM data structure has the random-sampling-based compressible property and the proposed random sampling method is space-efficient. On the web document classifying, the experiment C tries to evaluate the effectiveness of the RSEB algorithm, and the experiment D wants to verify the random-sampling-based compressible property of the TLM data structure in the multi-category situation.

In the experiment A, the *bogo*, *tftS3F*, and the *esf* filter run on the IFFT on the TREC07p corpus separately, and the *esf* filter sets its random sampling rate R_{rs}=1. The detailed experimental results are showed in Table 3. The results show that the *esf* filter can complete filtering task in high speed (2,834 sec), whose overall performance 1-ROCA is comparable to the best *wat3* filter's (0.0055) among the participators at the TREC 2007 spam filtering evaluation. The time and 1-ROCA performance of the *esf* filter exceed the *bogo*'s and the *tftS3F*'s more.

Table 3. Performance Statistics of Email Spam Filtering

	Time (sec)	1-ROCA (%)	TREC 2007 Rank
esf	2,834	0.0055	
wat3		0.0055	1
tftS3F	62,554	0.0093	2
bogo	25,100	0.1558	

In the experiment B, we run the *esf* filter under different random sampling rate R_{rs} from the 90% down to the 10%. The *esf* filter repeatedly runs 30 times for each random sampling rate, and here reports the mean performance among the 30 results for each random sampling rate. The detailed random sampling rate (R_{rs}), final indexed token compressing rate (R_{tc}), and performances are showed in Table 4. Where, the R_{rs} is a predefined priori value, while the R_{tc} is a posteriori value after the filtering task, and is defined as the ratio of the number of tokens in the final TLM to the total number of processed tokens during the filtering task. The space is the number of tokens in the final TLM storage.

Table 4. Random Sampling Rate, Token Compressing Rate and Performances

R_{rs}	R_{tc}	Time (sec)	Space (num)	1-ROCA (%)
100	100	2,834	15,754,699	0.0055
90	94	2,715	14,763,087	0.0055
80	87	2,607	13,660,951	0.0054
70	79	2,481	12,511,131	0.0053
60	71	2,139	11,257,499	0.0053
50	63	2,130	9,895,697	0.0055
40	54	2,094	8,467,245	0.0053
30	44	2,066	6,860,210	0.0055
20	32	2,028	5,071,819	0.0064
10	19	2,006	2,984,139	0.0066

We find that the 1-ROCA is almost a constant (≈0.0055) while the R_{rs} varying from the 100% down to the 30%, which indicates if we randomly remove up to 70% tokens at the time of online training, the 1-ROCA will not be influenced obviously. On average of the 30 results, there are four 1-ROCA values exceed the best one (0.0055). Table 4 shows that the final indexed token compressing rate approximates a direct ratio of the random sampling rate, which proves that random-sampling-based token feature selection according to the theoretical uselessness rate heuristic between 63% and 82% is effective in the online situation.

In the experiment C, the *wdc* classifier runs on the 12-category Chinese WDCT, and sets its random sampling rate R_{rs}=1. Through evenly splitting the TanCorp-12 dataset, we make the three-fold cross validation. The mean MacroF1 and the mean MicroF1 are showed in Table 5, where the results of other four classifiers are cited from existing researches [2]. The results show that the *wdc* classifier can complete classifying task in high MacroF1 (0.8696) and high MicroF1 (0.9126), whose performance exceeds the *centroid*'s, the *kNN*'s, the *winnow*'s, and approaches to the best *SVM* classifier's MacroF1 (0.9172) and MicroF1 (0.9483).

Table 5. MacroF1 and MicroF1 Results

	MacroF1	MicroF1
SVM	0.9172	0.9483
wdc	0.8696	0.9126
centroid	0.8632	0.9053
kNN	0.8478	0.9035
winnow	0.7587	0.8645

In the experiment D, we run the *wdc* classifier under different random sampling rate R_{rs} from the 90% down to the 10%. The *wdc* classifier repeatedly runs 30 times for each random sampling rate, and here reports the mean performance among the 30 results for each random sampling rate. The detailed random sampling rate (R_{rs}), training indexed token compressing rate (R_{tc}), and performances are showed in Table 6. Where, the R_{tc} is a posteriori value after the training, and is defined as the ratio of the token number in the TLM after the training to the total number of processed tokens during the training. On average of the 30 results, Table 6 shows that the training indexed token compressing rate approximates a direct ratio of the random sampling rate, which proves that random-sampling-based token feature selection according to the theoretical uselessness rate heuristic between 63% and 79% is also effective in the multi-category situation.

Table 6. Random Sampling Rate, Token Compressing Rate and Performances

R_{rs}	R_{tc}	MacroF1	MicroF1
100	100	0.8696	0.9126
90	93	0.8715	0.9136
80	86	0.8677	0.9119
70	79	0.8663	0.9113
60	71	0.8657	0.9114
50	63	0.8653	0.9103
40	53	0.8609	0.9083
30	43	0.8570	0.9051
20	32	0.8517	0.9010
10	19	0.8345	0.8921

6 Conclusion

This paper investigates the power law distribution and proposes a RSEB algorithm for the online multi-category TC problem. The experimental results show that the RSEB algorithm can obtain a comparable performance compared with other advanced machine learning TC algorithms in email spam filtering application and Chinese web document classifying application. The RSEB algorithm can achieve the state-of-the-art performance at greatly reduced space-time complexity, which can satisfy the restriction of limited space and time for many practical large-scale applications.

With the development of mobile computing and network communicating, the spam concept extends from email spam to instant messaging spam, short message service spam, and so on. A web document may belong to the hierarchical category or may

have multiple category labels. Further research will concern the short message TC problem, the multi-hierarchy TC problem, and the multi-category multi-label TC problem. According to the ubiquitous power law, the RSEB algorithm is more general and can be easily transferred to other TC applications.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
2. Tan, S., Cheng, X., Ghanem, M., Wang, B., Xu, H.: A novel refinement approach for text categorization. In: CIKM 2005: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 469–476 (2005)
3. Sculley, D., Wachman, G.M.: Relaxed online SVMs for spam filtering. In: SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 415–422 (2007)
4. Cormack, G.V.: Email spam filtering: a systematic review. Foundations and Trends in Information Retrieval 1(4), 335–455 (2008)
5. Han, E.-H(S.), Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
6. Zhang, T.: Regularized winnow methods. In: Advances in Neural Information Processing Systems, vol. 13, pp. 703–709 (2000)
7. Liu, W., Wang, T.: Online active multi-field learning for efficient email spam filtering. Knowledge and Information Systems 33(1), 117–136 (2012)
8. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
9. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Review 51, 661–703 (2009)
10. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Transactions on Neural Networks 10(5), 1048–1054 (1999)
11. Liu, W., Wang, T.: Utilizing multi-field text features for efficient email spam filtering. International Journal of Computational Intelligence Systems 5(3), 505–518 (2012)
12. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Computing Surveys 38(2), Article 6 (2006)