International Journal of Mobile Computing and Multimedia Communications
Volume 13 • Issue 2

English Article Style Recognition and Matching by Using Web Semantics

Mi Zhou, Dalian University of Science and Technology, China*
Lina Peng, Dalian Naval Academy, China

ABSTRACT

With the explosion of internet information, people feel helpless and find it difficult to choose in the face of massive information. However, the traditional method to organize a huge set of original documents is not only time-consuming and laborious, but also not ideal. The automatic text classification can liberate users from the tedious document processing work, recognize and distinguish different document contents more conveniently, make a large number of complicated documents institutionalized and systematized, and greatly improve the utilization rate of information. This paper adopts termed-based model to extract the features in web semantics to represent documents. The extracted web semantics features are used to learn a reduced support vector machine. The experimental results show that the proposed method can correctly identify most of the writing styles.

KEYWORDS

Document Type Recognition, Reduced Support Vector Machine, Vector Space Model, Web Semantics

1. INTRODUCTION

With the continuous popularization and development of information superhighway, information technology has penetrated into every corner of our social living (He 2021, Camero 2019). It has changed people's life and work style with unprecedented speed and ability. We are in an era of information explosion (Kumari 2017). On one side, the Internet contains a vast amount of information which is far beyond people's imagination. On the other hand, people often feel helpless when they face the vast ocean of information. It is called as information overload (Schmitt 2018, Swar 2017). It is a challenging task to help people effectively to manage massive information and quickly select useful information that they are interested in.

The web information is increasing, including online news, e-magazines, online technical reports, online documents, e-mail, BBS, online announcements (Yamamoto 2018). The traditional method to handle daily huge amount of information is time-consuming and laborious. The automatic text classification can directly filter and classify the document information (Kadhim 2019, Nguyen 2018). The user can only receive minor part which they are interested in. Then, users can be liberated from tedious document processing work and can easily understand and distinguish different document

DOI: 10.4018/IJMCMC.293751 *Corresponding Author

contents. A large number of complicated documents can be regulated and systematized, and the utilization rate of information can be greatly improved.

One the other hand, while people can easily send and obtain web information, they also face with many harmful or illegal information, such as pornography, violence, and superstition. Unhealthy content such as gambling can be seen everywhere on the Internet. Even criminal text may be exist in BBS, blog, e-mail etc. to carry out reactionary propaganda, fraud, extortion, terrorist threats, drug sales and other illegal and criminal activities. Installing illegal information filtering software cannot effectively prevent the occurrence of illegal web information. Through legislative means, investigating the criminal responsibility of criminals can effectively crack down on this kind of criminal behavior. However, due to the lack of effective evidence, the criminals may be free from the evasion of legal sanction.

Text classification refers to the process of marking a free document with one or more predefined category labels according to its content information (Kowsari 2019, Mirończuk 2018). In order to correctly perform the task of text classification, it must input the useful information of the text into the computer to scientifically abstract the text and establish mathematical model to describe the text. The document expression is a key part of the text classification. The text representation refers to many representation methods and techniques of text retrieval (Wang 2020, Luo 2019). The common used text retrieval methods include: Boolean model (Lashkari 2009), vector space model (Raghavan 1986) and probabilistic model (Feng 2018). These models deal with feature weighting, category learning, and similarity calculation from different perspectives.

By closely combining with machine learning, the vector space model has been successfully used in text classification and becomes a mainstream method in the field of text classification. Vector space model (VSM) was first proposed the field of information retrieval. Then, it has been widely used in the field of text classification. In the vector space model based text classification method, the documents are converted as vector form by using term frequency and inverse document frequency. The vectors are indexed by inverted documents to calculate document similarity. Although vector space model has been solved text representation, it still needs to assume that words in the document are independent with each other to reduce the complexity of the representation. This paper adopts a termed-based model to represent the document and utilizes the extracted features to learn a reduced support vector machine to recognize the document type.

2. PROBLEM DESCRIPTION

Text classification is a supervised learning process. It learns a classification model to represent the relation between text features and text labels by a training set which consists of massive labeled documents. The features text document are input into learnt classification model to determine the document type. The text classification is mathematical mapping which maps the document to the associated type. The mapping can be represented as:

$$f : T \rightarrow L \tag{1}$$

In Equation (1), T represents the document set, while L represents the document type set.

The mapping rule of text classification is based on the data information of samples from each class to summarize the regularity of classification and establish the rules to determine the text related categories. The classification of text is based on document's content other than the data pattern in the document. It means that the concept of which type of text is related to is subjective.

From the above description, the flowchart of text classification contains two stages: training classification model and predicting future document according to learnt classification model. In the training classification model stage, the documents in the training set are represented as a unified form

by using vector space model. The features are processed by using feature extraction method. The processed features are used to learn a classifier. In predicting future document stage, a new document is represented as a feature vector by using vector space model. The feature vector is processed by using feature extraction method and the processed feature vector is input into the learnt classification model to determine its type. An illustration of the text classification is shown in Figure 1.

In Figure 1, the text classification framework consists of four parts: text preprocessing, text representation, classifier training and classification evaluation. The feedback mechanism heuristic adjusts the parameters of feature extraction and classification algorithms to make the text recognition can achieve the best performance.

3. DOCUMENT REPRESENTATION

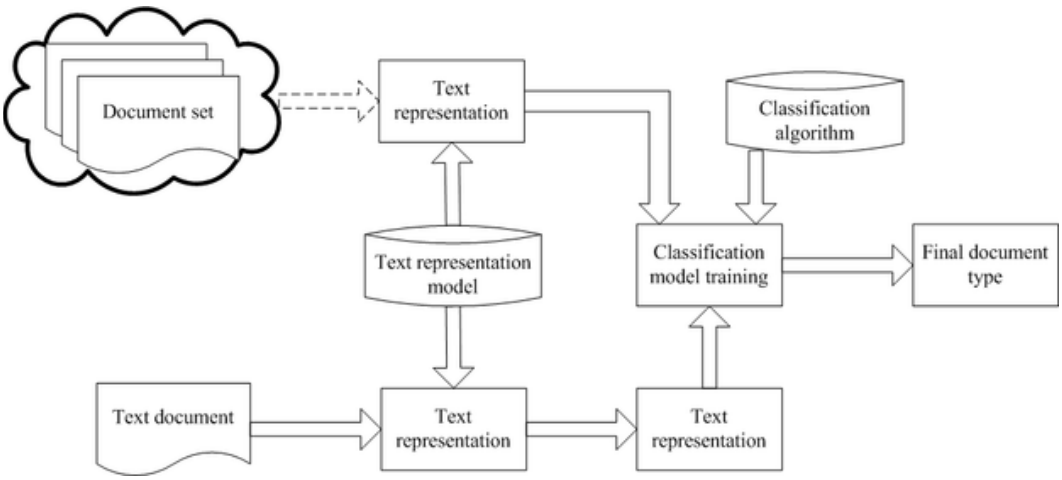
When reading an article or document, we can have a fuzzy understanding of the content according to our own understanding ability and experience. However, the computer does not have this ability. The human understanding refers to text semantics (Sinoara 2017). According to the current research level of computer technology, it is impossible for machine to read the natural text that people can understand. The machine only knows and understands 0 or 1. Thus, it is necessary to convert the document as the form that machine can understand. A unified expression is needed for the document.

With the development of information retrieval technology, several text retrieval models are developed. The current text retrieval models include Boolean model, vector space model, and probability model, etc. These models process feature weighting, category learning, and similarity computation from different perspectives. The vector space model is an effective text representation model.

In vector space model, it only considers the frequency of words and neglects the order of words in the document. The words in document should be different. The document space can be regarded as a vector space composed of a set of orthogonal entry vectors. Each document is represented as one of the normalized eigenvectors which is rewritten as follows:

$$S\left(d\right)=\left\{\left(t_1,w_1\left(d\right)\right);\ldots;\left(t_i,w_i\left(d\right)\right);\ldots;\left(t_n,w_n\left(d\right)\right)\right\} \tag{2}$$

Figure 1. The architecture of document type recognition framework



In Equation (2), t_i represents word item in document d, $w_i(d)$ represent the weight of word item t_i in document d. The word item t_i can be either the word or the phrase in the document d to improve the accuracy of content feature representation. The $w_i(d)$ can be regarded as a function of the frequency of word item t_i in the document d, which can be represented as $w_i(d) = \phi(tf_i(d))$. The $tf_i(d)$ represents the frequency of word item t_i in document d. The common used function for weight include: Boolean function, square root function, logarithmic function, and TFIDF function, which are written as Equation (3), (4), (5), and (6), respectively:

$$\phi(tf_i(d))=\begin{cases}1, & tf_i(d)\geq 1;\\ 0, & tf_i(d)=0.\end{cases} \tag{3}$$

$$\phi(tf_i(d))=\sqrt{f_i(d)} \tag{4}$$

$$\phi(tf_i(d))=\log\left(tf_i(d)+1\right) \tag{5}$$

$$\phi(tf_i(d))=f_i(d)\times\log\frac{N}{n_i} \tag{6}$$

Here, N represents the number of documents, while n_i represents the number of documents which contain word item t_i.

In general, the words, phrases, concepts are adopted as feature words. The word is the smallest semantic unit. The words are separated by space. The words are extracted from the document and are further processed into phrases, concepts. The words, phrases, and concepts are described as follows.

The word is the simplest feature item. Each feature item in the feature vector is associated with a word in the document. In general, case differences are ignored. When selecting words as feature items, the stop words are discarded, such as preposition and conjunction. In order to avoid the words from the same etymology appear many times, the stem of the words is extracted to represent the words have the same stem. For instance, “teach”, “teaching”, and “teacher” are converted as “teach” to be stored in the feature vector.

Using phrase as the feature item can overcome the weakness that simple words cannot reflect grammatical structure, the order in the paragraph, sentence, and words. Thus, an amount of information in the original document is not been effectively expressed. The meaning of words is often different in different phrases. The purpose of using phrases as feature words is to retain more information to distinguish document types. There are two methods to extract phrases from document. One is statistical method, which realizes phrase discovery through the co-occurrence probability statistics of words. This method can be suitable for a wide range of fields, but it needs a lot of training samples. The other is rule-based method, which identifies phrases through tagging dictionaries and word formation rules. However, this method is not flexible in grammar, and it is difficult to solve the ambiguity of word meaning. The dictionary cannot contain all the natural language phrases. Thus, it is difficult to exhaust all the rules of words.

Using concept as the feature item maximizes the internal similarity and minimizes the similarity between classes by combining the words in the source document according to a certain relationship. And then, the words are abstracted to the concept level to generate feature items. The feature items generated in this way contain more semantic information and have lower redundancy because similar information is merged.

In addition, some forms can also be used to generate feature items, such as tuple, some regular pattern. However, words, phrases, and concepts are most widely used form in document representation. Both phrases and concepts can be regarded as the combination or synthesis of the words. This paper adopts words as feature items in document representation.

Feature selection in document representation refers to discard the words that cannot contribute or have little contribute to distinguish document type. The feature selection can reduce the computational complexity on the basis of text preprocessing. There are several principles to select feature items. First, we should select those language units that contain more semantic information and have stronger ability to express the text as feature items. Second, the distribution of the text on these feature items should have obvious statistical regularity. Third, the selection process should be easy to implement and the associated time and space complexity.

The word can express whole semantic information. However, not all words are suitable as feature items. High frequency words and low frequency words are less effective than medium frequency words. The reason is that high frequency words in all articles have similar high frequency, low frequency words appear less in the text. Both high frequency word and low frequency word are not suitable to analysis by using statistical methods. The medium frequency words are most relevant to the topic of the document.

The common used feature selection for text classification includes text frequency, information Gain, mutual information, χ^2-test, and term strength. It is important to choose the best feature selection among different feature selection algorithms. The performance of the vector space model directly depends on the selection of feature items and the calculation of weight.

The classical weight of feature item must considers term frequency and inverse document frequency (Havrlant 2017). The term frequency refers to the number of the words appearing in the document. The inverse document frequency refers to the quantitation of the distribution of words in the document. The common method to calculate inverse document frequency is represented as $log_2(\frac{N}{n_k}+0.01)$. The N represents the number of documents in the document set, while n_k represents the number of the documents that contain the word.

The weight can be represented by combining term frequency and inverse document frequency as follows:

$$w_{i,k}=tf_{i,k}\times\log_2\left(\frac{N}{n_i}+0.1\right) \tag{7}$$

In Equation (7), $tf_{i,k}$ represents the term frequency that word W_k occurs in document D_i, $w_{i,k}$ represents the weight of word W_k in document D_i ($k=1,\ldots,m$, m is the number of words). The vector is normalized by using the following equation:

$$w_{i,k}=\frac{tf\left(W_k,D_i\right)\times\log_2\left(\frac{N}{n_i}+0.1\right)}{\sqrt{\sum_{W_k\in D_i}\left(tf\left(W_k,D_i\right)\times\log_2\left(\frac{N}{n_i}+0.1\right)\right)^2}} \tag{8}$$

The above formula is based on the assumption that the most meaningful feature words to distinguish documents should be those words that appear frequently enough in documents and less frequently enough in other documents in the document set. An improvement of the weight is represented as follows:

$$w_{i,k}=N\left(W_k,D_i\right)*\left(\log_2\frac{N\left(W_k\right)}{N}\right)^2 \tag{9}$$

In Equation (9), $N(W_k,D_i)$ is the times that word W_k occurs in the document D_i, $N(W_k)$ is the times that word W_k occurs in the training corpus, N the sum of the times that all words occur in the training corpus. The extracted features are used to represent the document. Then, the training corpus consisting of extracted features are used to learn a classification model. The common used classification model include: k nearest neighbor (Gou 2019), Gaussian processing (Manogaran 2018), logistic regression (Ranganathan 2017), and support vector machine (Cervantes 2020) etc. Since the processing to learn support vector machine is time-consuming, this paper adopts reduced support vector machine (Zhu 2017) as the classification model.

4. DOCUMENT TYPE RECOGNITION BY USING CLASSIFIER

Let $X\times Y$ represent the training set. The X is the feature set and Y is the label set. The $x_i\in\mathbb{R}^n$ is the feature of the i^{th} document, y_i is the associated label. In classical support vector machine $y_i\in\{+1,-1\}$. The aim of support vector machine is to find an optimal hyperplane $f(x)=w^T\varphi\left(x\right)+b$ via maximize the minimum margin between positive class and negative class. The optimal hyperplane is obtained by the following programming:

$$\begin{aligned}\min_{w,\rho}\quad & \frac{1}{2}w^Tw+C\sum_{i=1}^{l}\xi_i\\ s.t\quad & y_i\left(w^T\varphi\left(x_i\right)+b\right)\geq 1-\xi_i,\quad i=1,2,\ldots,l\\ & \xi_i\geq 0,\quad i=1,2,\ldots,l.\end{aligned} \tag{10}$$

In Equation (10), l represents the number of samples in the training set, C is the panel factor to balance the risk error and structural error, $\phi(x_i)$ is the image of sample x_i in the kernel reproducing Hilbert space to treat non-linear separable classification problem. The $\phi(x_i)$ is an implicit function, whose inner product can be computed via $<\phi(x_i),\phi(x_j)>=k(x_i,x_j)$. The kernel function is an explicit function which can be computed directly. The common used kernel functions include: linear kernel, polynomial kernel, Gaussian kernel, exponential kernel etc. When the dataset is linear separable, the linear function is adopted.

The Equation (10) is a convex programming whose solution is the same as that of its dual programming. After introducing the Lagrange multiplier α_i for the constraints $y_i\left(w^T\varphi\left(x_i\right)+b\right)\geq 1-\xi_i$ in the Equation (10), the dual form of Equation (10) is written as follows:

$$\begin{aligned} \min_{\alpha} \quad & \alpha^T Q\alpha - \alpha \\ s.t. \quad & \sum_{i=1}^{l} y_i\alpha_i = 0, \quad i = 1,\ldots,l \\ & 0 \le \alpha_i \le C, \quad i = 1,\ldots,l \end{aligned} \tag{11}$$

In Equation (11), α is the vector form of Lagrange multipliers, Q is the kernel matrix whose element is $Q(i,j) = y_i y_j k(x_i, x_j)$. The Equation (11) can be solve be decomposition algorithm or chucking algorithm. The weight and bias of the optimal hyperplane are rewritten as follows:

$$w = \sum_{x_i \in SVs} \alpha_i \varphi\left(x_i\right) \tag{12}$$

$$b = y_i - \sum_{x_j \in SVs} \alpha_i k\left(x_i, x_j\right), \quad 0 < \alpha_i < C \tag{13}$$

In Equation (12) and (13), the SVs represents the support vector set. The support vector is the sample which is with non-zero Lagrange multiplier $\left(0 < \alpha_i \le C\right)$. Then, the optimal hyperplane is rewritten as follows:

$$f(x) = \sum_{x_i \in SVs} \alpha_i k\left(x, x_i\right) + b \tag{14}$$

The Equation (14) is only determined by the support vectors. The complexity of Equation (14) is determined by the number of support vectors. The scale of programming (13) and the number of support vectors are related with the size of the training set. When the size of training set is huge, the number of support vectors is also very large. Thus, it is time-consuming to solve programming (13). In order to solve this issue, the sample reduction strategy is introduced into support vector machine model learning.

The optimal hyperplane $f(x)$ is only determined by minor support vectors. If removing the samples that would not become support vectors, the result of support vector machine will not change. Let $\mathbf{X}'\times \mathbf{Y}'$ be the set consisting of all potential support vectors. The reduced support vector machine learning on reduced training set is formulated as follows:

$$\begin{aligned} \min_{w,\rho} \quad & \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i \\ s.t \quad & y_i\left(w^T\varphi\left(x_i\right) + b\right) \ge 1 - \xi_i, \quad \left\{x_i, y_i\right\} \in X' \times Y' \\ & \xi_i \ge 0, \quad i = 1,2,\ldots,l. \end{aligned} \tag{15}$$

The dual form of programming (15) is the same as that of programming (11). The scale of programming (15) is related with the size of X'. Let l' represent the size of set X'. In general, $l' << l$. The potential support vectors generally locate in the boundary region of each class or near the optimal hyperplane. This paper adopt the distribution of nearest neighbors (Zhu 2014) to detect the samples in the boundary region of each class as the candidate support vectors for reduced support vector machine learning. The distribution of nearest neighbors is described by the following equation:

$$d_{x_i} = \sum_{x_j \in NN_k\left(x_i, X_{y_i}\right)} \frac{\left\langle x_i, x_i \right\rangle - \left\langle x_i, x_j \right\rangle - \frac{1}{k} \sum_{x_p \in NN_k\left(x_i, X_{y_i}\right)} \left\langle x_i, x_p \right\rangle + \frac{1}{k} \sum_{x_p \in NN_k\left(x_i, X_{y_i}\right)} \left\langle x_j, x_p \right\rangle}{\sqrt{\left\langle x_i, x_i \right\rangle - \frac{2}{k} \sum_{x_p \in NN_k\left(x_i, X_{y_i}\right)} \left\langle x_i, x_p \right\rangle + \frac{1}{k} \sum_{x_p, x_q \in NN_k\left(x_i, X_{y_i}\right)} \left\langle x_p, x_q \right\rangle} \sqrt{\left\langle x_i, x_i \right\rangle - 2\left\langle x_i, x_j \right\rangle + 2\left\langle x_j, x_j \right\rangle}} \tag{16}$$

In Equation (16), $NN_k\left(x_i, X_{y_i}\right)$ represents nearest neighbor set of sample x_i in set X_{y_i}, the set X_{y_i} represents the set consists of all samples with same class of sample x_i. In general, the Equation (16) is close to 1 if the sample locates in the boundary region of a class, and close to 0 if the sample locates in the interior of a class. Then, we only need to retain the samples with high values as the candidate training set for reduced support vector learning. The whole procedure is summarized as shown in Algorithm 1.

In Step 3 of reduced support vector machine, l_p represents the number of samples in positive class. In Step 6 of reduced support vector machine, l_n represents the number of samples in negative class. The parameter δ represents the size of retained subset of the training set. The larger δ is, the more samples are retained in reduced support vector machine.

5. EXPERIMENTS AND SIMULATIONS

In this section, we use the framework in Figure 1 to recognize the document type. We adopts the documents from Reuter's news. The Reuter's news is a standard dataset which is widely used in the research of text classification. The documents are denoted manually and processed as the fixed form. The Reuter's news include four types: corporate/industrial (CCAT), government/social (GCAT), markets (MCAT), and economics (ECAT). The corporate/industrial includes strategy/plans, legal/judicial, and share listing. The government/social includes sports, environment and natural word. The markets include equity markets, bond markets. The economics includes economics performance, monetary/economic. Each document type contains 600 documents and there are 2,400 documents in total for training. Each document is represented as the extracted features. The classical support vector machine only can deal with binary class classification problem. In order to address multi-class classification problem, the one versus one strategy is adopted. In one versus one strategy, 6 binary class classification models are learnt, include CCAT versus GCAT, CCAT versus MCAT, CCAT versus ECAT, GCAT versus MCAT, GCAT versus ECAT, and MCAT versus ECAT. The learnt model is evaluated on a test set which consist of 800 documents in total, 200 document per type.

The document type recognition framework is evaluated from validity, computational complexity and simplicity. The validity measures the ability whether the framework can classify document type

Algorithm 1. Reduced support vector machine

Step 1: calculating values of Equation (16) for all samples in positive class;
Step 2: sorting the samples in positive class according to the values of Equation (16) in descending order;
Step 3: retaining the top $\delta * l_p$ samples in positive class to construct X'_p;
Step 4: calculating values of Equation (16) for all samples in negative class;
Step 5: sorting the samples in negative class according to the values of Equation (16) in descending order;
Step 6: retaining the top $\delta * l_n$ samples in positive class to construct X'_n;
Step 7: combing X'_p and X'_n as X' and learning support vector machine model on $X' \times Y'$.

correctly. The computational complexity includes time complexity and space complexity. The time complexity includes training time and test time. The simplicity requires the classification model to be as simple as possible. The validity is the most important index. If a document type recognition model has simple structure, is easy to be learnt, but cannot classify document correctly, the document type recognition is useless. The validity is evaluated by precision, recall, and F1-measure.

Let TP, FP, TN, and FN represent true positive, false positive, true positive, and true negative, respectively. Then, precision, recall, and F1-measure are defined as following equations:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$F_1 = \frac{2Precision * Recall}{Precision + Recall} \tag{19}$$

The experimental results are reported in Table 1 in terms of precision, recall, and F1-measure. The features are selected by Boolean model, vector space model, probability model, and term based model. The classification algorithm adopts support vector machine (SVM) and reduced support vector machine (RSVM).

From the results in Table 1, when support vector machine is used as classification algorithm, the term-based model achieves 80.73%, 82.35%, 81.89% for precision, recall, and F1-measure. The results is higher than Boolean model 9.5%, 8.83%, and 9.22%; is higher than vector space model 6.2%, 6.99%, 7.96%; and is higher than 6.34%, 6.94%, 7.02% for precision, recall, and F1-measure. The reduced support vector machine retains 15% of the training set. The difference between support vector machine and reduced support vector machine is less than 0.15% for precision, recall, and F1-measure.

The complex of classification model of support vector machine and reduced support vector machine is reported in Table 2 in terms of training time and the number of support vectors. The complex of support vector machine model depends on the number of support vectors. The number of support vectors in Table 2 is the average of the number of the support vectors of 6 binary classification models. The reduced support vector machine retains 15% of the training set.

From the results in Table 2, it can be found that the reduced support vector consumes 6.9% of the support vector machine and the support vectors of reduced support vector machine is only 16.15% of that of support vector machine.

In Figure 2, we report the results of reduced support vector machine with different percentage of the training set. The size of retained subset ranges 5%, 10%, 15%, 20%, and 25%. The document is represented by termed based model.

From the result of Figure 2, it can be found that when 15% of the training set is retained, the reduced support vector machine can maintain the performance which is very close to that of support vector machine.

Figure 3 reports the confusion matrix when 15% of the training set is retained.

From the result of Figure 3, it can be found that the accuracy can achieve 78%, 80.5%, 84.5%, 79.5% for CCAT, GCAT, MCAT, and ECAT, respectively.

Table 1. The performance comparison of document type recognition

		SVM	RSVM
Boolean model	Precision (%)	71.23	71.09
	Recall (%)	73.52	73.39
	F1-measure (%)	72.67	72.53
Vector space model	Precision (%)	74.53	74.47
	Recall (%)	75.36	75.42
	F1-measure (%)	73.93	74.03
Probability model	Precision (%)	74.39	74.41
	Recall (%)	75.41	75.32
	F1-measure (%)	74.87	74.73
Term-based model	Precision (%)	80.73	80.59
	Recall (%)	82.35	82.42
	F1-measure (%)	81.89	81.83

Table 2. The model complexity comparison between support vector machine and reduced support vector machine

	SVM	RSVM
Training time (sec.)	252.31	17.42
No. of SVs	87	14

Figure 2. The relation between the size of the retained subset in reduced support vector machine and precision, recall, and F1-measure

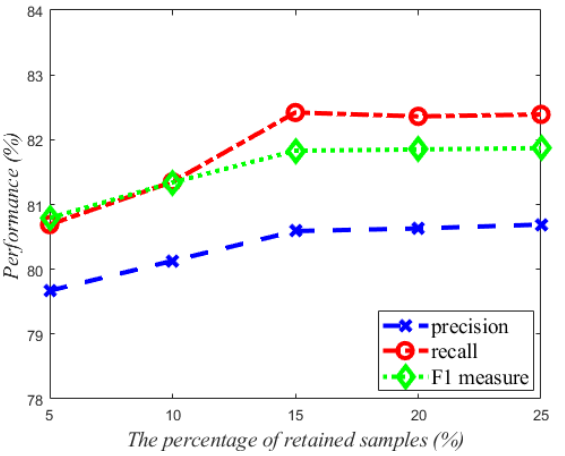


Figure 3. The confusion matrix of document type recognition of reduced support vector machine

	CCAT	GCAT	MCAT	ECAT	Accuracy (%)
CCAT	156	23	23	29	67.53
GCAT	13	161	5	7	86.56
MCAT	27	7	169	5	81.25
ECAT	4	9	3	159	90.86
Accuracy (%)	78.00	80.50	84.50	79.50	80.63

6. CONCLUSION

The automatical document type recognition and analysis plays an important role in the massive emerging web information. From, the web documents are collected and denoted the type by domain experts. The collected documents are constructed as training set to learn a classification model. Then, the future document is represented as term-based features and the features are input into classification model to determine the document type. In the proposed document type recognition framework, the term-based model is adopted as feature representation method. The reduced support vector machine is adopted as classification algorithm. In order to solve multi-class classification problem, the one versus one strategy is adopted. The results of the experiments on Reuter's news dataset show that most of the document type can be correctly identified by the proposed document type recognition framework.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Camero, A., & Alba, E. (2019). Smart City and information technology: A review. *Cities (London, England)*, *93*, 84–94. doi:10.1016/j.cities.2019.04.014

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. doi:10.1016/j.neucom.2019.10.118

Feng, G., Li, S., Sun, T., & Zhang, B. (2018). A probabilistic model derived term weighting scheme for text classification. *Pattern Recognition Letters*, *110*, 23–29. doi:10.1016/j.patrec.2018.03.003

Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, *115*, 356–372. doi:10.1016/j.eswa.2018.08.021

Havrlant, L., & Kreinovich, V. (2017). A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, *46*(1), 27–36. doi:10.1080/03081079.2017.1291635

He, W., Zhang, Z. J., & Li, W. (2021). Information technology solutions, challenges, and suggestions for tackling the COVID-19 pandemic. *International Journal of Information Management*, *57*, 102287. doi:10.1016/j.ijinfomgt.2020.102287 PMID:33318721

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, *52*(1), 273–292. doi:10.1007/s10462-018-09677-1

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information (Basel)*, *10*(4), 150. doi:10.3390/info10040150

Kumari, K. (2017). Information explosion, information anxiety and libraries: Strategies for Intervention. *World Wide Journal of Multidisciplinary Research and Development.*, *3*(12), 166–169.

Lashkari, A. H., Mahdavi, F., & Ghomi, V. (2009, April). A Boolean model in information retrieval for search engines. In *2009 International Conference on Information Management and Engineering* (pp. 385-389). IEEE. doi:10.1109/ICIME.2009.101

Luo, L. X. (2019). Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Personal and Ubiquitous Computing*, *23*(3), 405–412. doi:10.1007/s00779-018-1183-9

Manogaran, G., & Lopez, D. (2018). A Gaussian process based big data processing framework in cluster computing environment. *Cluster Computing*, *21*(1), 189–204. doi:10.1007/s10586-017-0982-5

Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*, 36–54. doi:10.1016/j.eswa.2018.03.058

Nguyen, D. (2018, June). Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1069-1078). Academic Press.

Raghavan, V. V., & Wong, S. M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, *37*(5), 279–287. doi:10.1002/(SICI)1097-4571(198609)37:5<279::AID-ASI1>3.0.CO;2-Q

Ranganathan, P., Pramesh, C. S., & Aggarwal, R. (2017). Common pitfalls in statistical analysis: Logistic regression. *Perspectives in Clinical Research*, *8*(3), 148. PMID:28828311

Schmitt, J. B., Debbelt, C. A., & Schneider, F. M. (2018). Too much information? Predictors of information overload in the context of online news exposure. *Information Communication and Society*, *21*(8), 1151–1167. doi:10.1080/1369118X.2017.1305427

Sinoara, R. A., Antunes, J., & Rezende, S. O. (2017). Text mining and semantics: A systematic mapping study. *Journal of the Brazilian Computer Society*, *23*(1), 1–20. doi:10.1186/s13173-017-0058-7

Swar, B., Hameed, T., & Reychav, I. (2017). Information overload, psychological ill-being, and behavioral intention to continue online healthcare information search. *Computers in Human Behavior*, *70*, 416–425. doi:10.1016/j.chb.2016.12.068

Wang, Y., Yang, Y., Chen, Y., Bai, J., Zhang, C., Su, G., Kou, X., Tong, Y., Yang, M., & Zhou, L. (2020, April). Textnas: A neural architecture search space tailored for text representation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 9242–9249. doi:10.1609/aaai.v34i05.6462

Yamamoto, Y., Yamamoto, T., Ohshima, H., & Kawakami, H. (2018, May). Web access literacy scale to evaluate how critically users can browse and search for web information. In *Proceedings of the 10th ACM Conference on Web Science* (pp. 97-106). doi:10.1145/3201064.3201072

Zhu, F., Yang, J., Gao, J., Xu, C., Xu, S., & Gao, C. (2017). Finding the samples near the decision plane for support vector learning. *Information Sciences*, *382*, 292–307. doi:10.1016/j.ins.2016.12.019

Zhu, F., Yang, J., Ye, N., Gao, C., Li, G., & Yin, T. (2014). Neighbors' distribution property and sample reduction for support vector machines. *Applied Soft Computing*, *16*, 201–209. doi:10.1016/j.asoc.2013.12.009