國立交通大學

資訊科學與工程研究所

碩 士 論 文

基於RSS Feed之中文部落格文章分類系統

Chinese Blog Article Classification Base on RSS Feed

研 究 生：劉嘉倩

指導教授：袁賢銘　教授

中 華 民 國 九 十 九 年 六 月

基於 RSS Feed 之中文部落格文章分類系統

Chinese Blog Article Classification Base on RSS Feed

研究生：劉嘉倩　　Student：Jia-chiam Liu

指導教授：袁賢銘　　Advisor：Shyan-Ming Yuan

國立交通大學

資訊科學與工程研究所

碩士論文

A Thesis

Submitted to Institutes of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

基於 RSS Feed 之中文部落格文章分類系統

研究生：劉嘉倩　　　　　　　　指導教授：袁賢銘

國立交通大學資訊科學與工程研究所

摘要

在台灣已經有越來越多人使用 RSS (簡易資訊聚合)的方式訂閱部落格文章。然而，台灣大多的部落格供應平台僅提供支援單一 RSS 輸出，訂閱者無法利用分類 RSS 的功能選擇某個站台的某個分類。基於此出發點，本研究實作一套個人化的部落格 RSS 訂閱系統，根據文章的內文自動判斷其分類後再發布給用戶，讓使用者不僅可選擇訂閱的部落格頻道，還能進一步設定想接收的文章類型。

本研究設計了一個雙層式 SVM 分類機制，使系統可以第一層先篩選出定義類別內的文章後再進行第二層的類別判斷，也設計了一種可以隨著定義類別數目增加而重複進行資料訓練的方法。經由實驗 10 項分類而得的結果，可達 87%的招回率及 95%的精確度。

Chinese Blog Article Classification Base on RSS Feed

Student : Jia-chiam Liu Advisor : Shyan-Ming Yuan

Institutes of Computer Science Engineering

National Chaio Tung University

Abstract

In Taiwan, more and more people subscribe RSS (Really Simple Syndication) to receive update information of blog. However, mostly Taiwanese BSP (Blog Service Provider) support single export of RSS, subscriber can't utilize the capability which like category RSS to choose some specific classifications of one blogger's. At this point, this research implement a personalized RSS subscribe service, which will automatically categorize the article based on its content, then publish to subscribers. So that, users can choose not only blog channels but also article type of category they want to receive further.

This research has designed a two-layer SVM classification mechanism, let system filter the articles which was non-defined category first, then judge its real category in second layer after. Additionally we designed a training module that can train repetitiously according to defined category increasing. The experiment result of 10 categories classify can achieve 87% of recall and 95% of precision finally.

Acknowledgements

此研究主題及本篇論文能夠順利完成最主要感謝我的指導教授袁賢銘老師，老師不僅給予我很大的空間自行發揮，更提供不同實驗方法之建議，讓我在整個研究過程中學習到許多。而三位口試委員：彭文志教授、柯皓仁教授、陳興忠教授在口試當天也給予了許多指導，對於此研究的完整性有非常大的幫助。

另外，感謝中央研究院之中文斷詞系統(CKIP)研究小組與交通大學資訊系計算機中心的學術合作，使的此研究能夠藉由此服務順利達到預期目標。

此外還要特別感謝國亨學長於整個研究過程的協助，同時還有家鋒學長、永威學長，以及諸位一起互相打氣奮鬥的同學們：阿貴、銘鴻、昱凱、玟瑾，很幸運能夠在大家的伴隨下度過這兩年美好日子。

最後，感謝我摯愛的家人，多虧你們在背後默默支持我，接受我任性的選擇重返學生身分，我才能有今日這份成就。此份於我學生生涯最後一個階段的成果獻給你們，願你們與我一起享受這份榮耀。

Table of content

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Preface

RSS has been around for more than 10 years but has only recently become popular. People can receive the update information from webmasters or content providers via subscripting RSS. When a new article is posted or a change is made, RSS feeds can automatically notify the subscribers.

With the prevalence of blog (also known as weblogs), majority of BSP (Blog Service Provider) offer the RSS subscription capability. And more and more people have the habit to read the specific blogger's articles regularly from RSS. Therefore, people have no need to bookmark every bloggers' site and visit each one to check their fresh articles. Even we can find popular topic through top RSS or recommend RSS list by online RSS reader/aggregator, also can subscribe specific keyword via Google Blog Search to receive relative information. In short, RSS is a way let us obtain network information faster and more convenient.

Many blogging software or blog publishing application, like WordPress[1], can classify by adding specific category RSS for the articles, to help visitors subscribe the interesting classification. But most Taiwanese BSP don't support the functionality about category RSS, for instance: Wretch, Pixnet, Xuite, Roodo …etc. So, different from the news station or the forum, subscribing a RSS from BSP is not that by category but that by single output of one blogger's site.

[1] http://wordpress.org/

1.2 Motivation

If subscriber has just interested in some domain of articles by some bloggers, subscribing by blogger's site is unsuitable. Because the content inside RSS contains overall update information of one blogger's website, subscriber can't subscript elasticity that only accepts the specific part of the renewal information. This kind of phenomenon will cause them spend more time to pick up the articles they want indeed, especially for those people who subscribe a great quantity of RSS; the mobile user will experience more obvious.

There are some alternatives, but not very ideal.

- Reduce the range through subscribing the keyword – When subscriber only want to acquire the RSS of specific themes clearly, this method is effective. But if the range that subscriber want to acquire is a fuzzy concept, for example the relevant topic of food includes cooking, eating, dish, it's still unable to improve the state.
- Subscribe from category website – In Taiwan, there exists a special website that offers categorized article links which are handpicked by users: http://funp.com/push/. Because these article links are all recommended by users, categorized accuracy is more believable. But a new recommended article does not represent a new posting article. Furthermore, user can't choose author.
- Subscribe from the BSP association- Some BSP establish the association for the popular theme, and then invite the famous bloggers to join this group. Relevant articles which these bloggers posting can be published by the official in unison. However, this way is limited to the association which BSP provide and unable choose author by oneself.

The above-mentioned three points are all unable to totally satisfy the demand: let users can not only subscribe by website but also designate content category. So, we hope to develop a subscribe system which is mainly aimed at Taiwanese BSP RSS to achieve this purpose.

1.3 Objective

The main direction of this thesis is automatic article classification. Why don't we use the <category> element inside the RSS directly? It's easy to find the category which article labels is often not correct when we browse the category page of blog sites. Most reasons are that users have not defined well before posting. So we unable to classify article only accord to this <category> element. We must try to fetch full content of article, and then judge its real category from the content.

Via this classification, we expect to reach certain accuracy on judgment for category RSS. And then help the readers with keen interest on certain categories only, rather than the entire headlines span. Readers can have one more option: category, which can be used to filter newer information of the articles they wanted to follow indeed.

In this whole research, we focus on Chinese word, and excluded the articles which comprised mainly with the pictures and videos.

1.4 Overview

This paper is organized as follows. In Chapter 2, we introduce the background knowledge of this thesis and relevant technology of text classification. In Chapter 3, we show the system architecture and explain how we develop the classification method. And then formally describe whole implement detail including three parts: classification training, server center, and client application in Chapter 4. Chapter 5 attaches the experiment result of classification and a simple System Usability Scale (SUS). Finally, concludes, discusses some issues and possible reformation of further research for this system in Chapter 6.

Chapter 2 Background

2.1 Really Simple Syndication (RSS)

RSS (most commonly expanded as Really Simple Syndication) is a standardized format used to publish frequently updated web content, something like blog articles, news headlines, audio, and video. An RSS document (which is called a "feed", "web feed", or "channel") includes full or summarized text, plus metadata such as publishing dates and authorship [1].

RSS feeds can be read using software called an "RSS reader", "feed reader", or "aggregator", which can be web-based, desktop-based, or mobile-device-based. A standardized XML file format allows the information to be published once and viewed by many different programs. The user subscribes to a feed by entering into the reader the feed's URI or by clicking an RSS icon in a web browser that initiates the subscription process. The RSS reader checks the user's subscribed feeds regularly for new work, downloads any updates that it finds, and provides a user interface to monitor and read the feeds.

2.2 Text Classification

Text classification (also known as Document Categorization) is a research topic that tries to decide the predefined classes (or category) of a text document. Nowadays, besides documents category, the classified issue is also applied in many other domain researches. For example, web page classification [**2**] [3] [**4**], spam mail filter [**5**], news events tracking [**6**]…etc.

In order to automatic classify text documents, we must reorganize the rule for which documents should be what category first, then computer can according as this rule to determine the others. However, it's difficult to obtain the effective classifying rule by artificial analysis. So, before classification process, it should perform to train; let the process can automatic study the rules which from manual classification experience and knowledge.

There are many methods for text classification, and different classifier will have different classified effect in different situation [**7**]. The famous Classifiers are SVM (Support Vector Machine), KNN (K-Nearest Neighbors), LLSF (Linear Least Square Fit), Perceptron, Neural Network, Naïve Bayes…etc [new]. This thesis attempts to use SVM to classify blog articles, which will introduce in the below section.

2.3 Chinese Word Segmentation

Word segmentation is a process of dividing written text into meaningful units. In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word delimiter. But Chinese, unlike Western languages, is written without spaces between words. So we should consider more influencing factors to segment the Chinese strings.

The method of Chinese word segmentation can divide into dictionary based approach, statistical based approach and hybrid approach [**9**] [**10**]:

- Dictionary based approach - Analysis the sentence by using the pre-defined dictionary files and finding the Chinese Words.
- Statistical based approach - Directed against the large amounts of corpus and using statistical techniques to calculating words in corpus. According to the character length which takes out, it can be divided into 2-Gram, 3-Gram and also to N-Gram.
- Hybrid approach - mixed of dictionary based and statistical approach.

In Taiwan, the Chinese word segmentation process “CKIP” in Academia Sinica [11] is the most prominent. It not only adopts hybrid approach to segment, but also uses a bottom-up merging algorithm [12] for Chinese unknown word extraction.

2.4 Vector Space Model

Vector space model (VSM) is a model for representing text documents (and any objects, in general) as vectors of identifiers. Hereinafter represent the vector space:

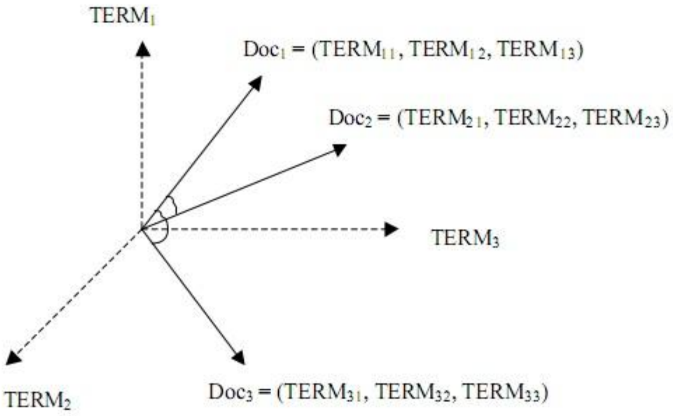
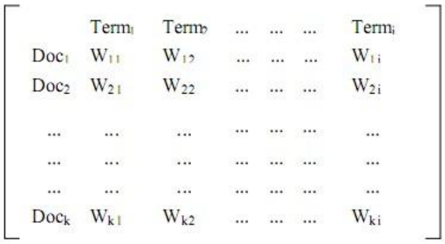

TERM1
Doc1 = (TERM11, TERM12, TERM13)
Doc2 = (TERM21, TERM22, TERM23)
TERM3
TERM2
Doc3 = (TERM31, TERM32, TERM33)

Figure 1. The Vector Space (From Salton & McGill [13]**)**

Gerard Salton proposed that expresses a document by the document index vector [**13**]. The content of a document can denoted a linear combination: D = (T_1, T_2,…,T_n) by the feature terms (T), and this representation can regarded as a vector in this vector space. Additional, we can use weighted feature (W) value to modify the term significances, then D = ([T_1, W_1], [T_2, W_2],…,[T_n, W_n]).

In this model, each document can be expressed as one vector in the space. By the vector concept, not only may facilitate express the relations between various documents, simultaneously may also use for to calculate the similarity. Even, it can be represented by a term-document matrix, shown in Figure 2 , let it more easily to calculate or apply in the computer.



$$\left[\begin{array}{ccccccc} & \text{Term}_1 & \text{Term}_2 & \cdots & \cdots & \cdots & \text{Term}_i \\ \text{Doc}_1 & W_{11} & W_{12} & \cdots & \cdots & \cdots & W_{1i} \\ \text{Doc}_2 & W_{21} & W_{22} & \cdots & \cdots & \cdots & W_{2i} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \text{Doc}_k & W_{k1} & W_{k2} & \cdots & \cdots & \cdots & W_{ki} \end{array}\right]$$

Figure 2. Term-Document Matrix (From Salton & McGill [13])

2.5 Support Vector Machine (SVM)

Support vector machine (SVM) is a kind of related supervised learning method used for classification and regression. Its main concept is aims at a set of training examples, each sets of instance-label as one category, and then SVM training algorithm builds a model that we can used to predict whether a new unknown example falls into which category.

The SVM basic methodology is in constructs a linear separation hyper-plane in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Consider the following example (see Figure 3), here training an algorithm to separate between black and white values:

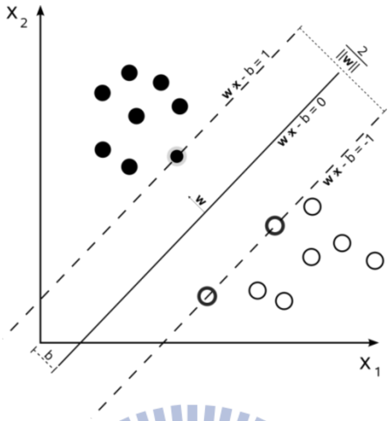


Figure 3. The linear SVM

The vector ***w*** is a normal vector what is perpendicular to the hyper-plane. The parameter $\frac{\boldsymbol{b}}{||\boldsymbol{w}||}$ determines the offset of the hyper-plane from the origin along the normal vector ***w***. By using geometry, we find the distance between these two hyper-planes is $\frac{\mathbf{2}}{||\boldsymbol{w}||}$. The SVM algorithm relies on making a choice as the best hyper-plane is the one that represents the largest separation, or margin, between the two classes. It means we want to minimize **||*w*||** so that the distance from the support vector to the nearest data point on each side is maximized.

The SVM may discover the independent variable and the dependent variable corresponding relationships the optimization classification hyper-plane. Therefore the SVM has the extremely good result in various types' classified question, at present has widely utilized in domains and so on biotechnology, text classification, phantom identification.

2.6 Performance measures

Classification effectiveness is usually measured in terms of the classic IR notions of recall and precision, adapted to the case of text categorization. The precision is defined as the fraction of retrieved documents that are relevant. Recall is defined as the fraction of relevant documents that are retrieved. The goal is attain both high recall and precision. Their definitions are as below:

- $recall =$

$$\frac{|\{rlevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

- $precision = \dfrac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$

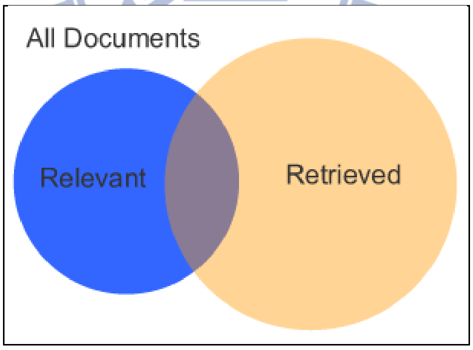


Figure 4. Relationship between Relevant and Retrieved

There is a trade-off between precision and recall. Greater precision decreases recall and greater recall leads to decreased precision. The F-measure (also called F-score or F_1 score) is the harmonic-mean of precision and recall and takes account of both measures [14].

- $F - score = 2 * \frac{precision * recall}{precision + recall}$

Chapter 3 Classification Design and System Architecture

3.1 Classification Method Development

Up to the present, text classification research mostly stands on the finite definition categorizations. However, categories of each BSP in Taiwan have not been unified, so there may have the exceptional condition that some articles do not belong to the category which we pre-defined. In order to avoid this condition, we design 2-layer classification scheme, one for flitting the non-defined category, classified to “other-category”; and another for real categorizing.

3.1.1 Dataset Collection

In text categorization system, generally divide the corpus into two parts: the training set and the test set. The training set is made up of many pro-classified documents, and is used to learn the category attribute to structure classifier. Each category includes some pre-classified documents. The test set is used to evaluate the classifier by assigning a category to each unclassified document in it.

We define some popular topic of categories by referring three Taiwan mainstream BSP [15]: Wretch[2], Pixnet[3], and Xuite[4]. Our dataset are mostly obtained from these three sites. Among them, the content of articles must be very clear and

[2] http://www.wretch.cc/blog/

[3] http://www.pixnet.net/blog

[4] http://blog.xuite.net/

positive, because these contents representative the judge basis of article category. And, we fetched 100 documents for each category in calculating conveniently, then partitioned 75% of the dataset to Training Set and 25% of the dataset to Test Set.

Moreover, we fetch articles of “other-category”, used for differentiating those articles that has already defined category. Among them, we possibly let authors are not duplicate, and try fetch various kinds which distributing equally.

3.1.2 Feature Selection

Before feature selecting, we delivered the training data to segmentation engine. In this stage, we obtained the individual token terms which are normal nouns (*Na*) and place names (*Nc*)[5] by CKIP [11] at first. Then we judged these token terms by stop word process and synonymous word process. The stop word list includes the insignificant neutral word which length is 1, e.g.ㄅ.ㄆ.ㄇ.們.事.點.者…etc, and the word term that appear in various category of articles frequently, e.g.東西.感覺.時候.方式.地方.問題.重點.顏色.關係.事情…etc. This kind of words of the latter usually has both high document frequencies in each different classification. In another process, the synonymy list includes the word groups of same meaning. Among them, we have consulted certain Chinese thesaurus that provider by some on-line synonym inquiry service[6]. These two processes are defined depending on the experience artificially in order to filter out the insignificant word and reduce the feature dimension. Eventually, we got the features in the form of segregate words basally.

[5] You can refer the Appendix 1 to see Speech Tag.

[6] http://www.kwuntung.net/synonym/

Even after above processing, most of the features are irrelevant or introduce noise which misleads the classifiers. Thus, feature selection is often performed in order to increase the efficiency and effectiveness of the classification. We believe that if a meaningful noun appears frequently in one category of article, this word should have the high classified value. So, we first constructs a local feature set for each category by selecting a set of features based on document frequency (df) threshold, and then constructs a global feature set based on the local feature set.

3.1.3 Classification by SVM

SVM method is suitable for the classification, especially for text classification [16] [17] [18] [19] [**20**]. And it based on the analysis can solve the small sample collection machinery study problem effectively [21]. We review and refer many research reports and literature [22] [**23**] [**24**] [**25**] [**26**], then discover SVM apply in text categorization have higher accuracy than the common categorization methods in machine learning generally. It's a major motive for us to use it be our classifier.

There exist numbers of implementations of SVM, varying in the speed and the quality of solutions. In this research, we used a library for support vector machines: LIBSVM[7]. Due to our categorized data were unable totally by cutting apart linearly on realistic space, we must draw support from the kernel function: $\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \emptyset(\mathrm{x_i})^T \emptyset(\mathrm{x_j})$ to deal these data. Here we adopted the C-SVC type of SVM, and the reasonable choice - radial basis function (RBF): $\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \exp\left(\gamma * \|\mathrm{x_i} - \mathrm{x_j}\|^2\right), \gamma > 0$ of kernel function [27]. Then, the data could be non-linearly mapped from input space to feature space.

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/. Copyright (c) 2000-2010 Chih-Chung Chang and Chih-Jen Lin

In the initial training stage, we had collected three categories: food, travel, and makeup of articles, and use ubiquitous *tf-idf* (term frequency – inverse document frequency) representation to be the feature attribute for SVM. The term frequency defined as $tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$, where value $n_{i,j}$ is the number of occurrences of the considered term $t_i$in document d_j; the inverse document frequency defined as $idf_i = log\frac{|D|}{|\{d: t_i \in d\}|}$, where $|D|$ is total number of documents in the corpus, $|\{d: t_i \in d\}|$ is number of documents where the term t_i appears.

However, there appeared a result which we couldn't understand at this state. After adjusting content of dataset, stop word, synonymous word and threshold repeatedly, the result shown as Table 1, where df-value threshold is 0.1, and there were 343 features. Observed the prediction result of test set, we found that all of wrong results were predicted into travel-category (see Table 1). Even we adjusted the parameter by each category individually, such strange situation still existed

Table 1. Result of SVM using tf-idf

Measur Category	**Recall of Test Set**	**Precision of Test Set**
food	92% (23/25)	100% (23/23)
travel	100% (25/25)	65.79% (25/38)
makeup	56% (14/25)	100% (14/14)

So, we tried to adopt another representation - word occurrences weighting (shown in Table 2), and linearly scaled them to the range [-1, +1] to be feature value for SVM. It's a simple approach to determine the weight $f_{i,k}$ of word *i* in document *k*.

Then, we got the average F-measures by tf-idf and occurrences respectively are 89.44% vs. 85.52%. And this new result of precision was more reasonable. Because of this result, we decided to use this weighting representation to continue the following experiment.

Table 2. Result of SVM using occur times

Measur Category	Recall of Test Set	Precision of Test Set
food	88% (22/25)	91.67% (22/24)
travel	96% (24/25)	85.71 % (24/28)
makeup	84% (21/25)	91.30% (21/23)

3.1.4 Classifiers Strategy

Before increasing the amount of category, we explain first how we classify multiply categories. Basic support vector machine (SVM) is a binary classifier, but only one classifier can't handle more than two categories. Therefore, we need to use other methods or many binary classifiers to combine classifier strategy model. At present, there are common two classifier model: one-against rest (OAR) classifiers strategy and one-against-one (OAO) classifiers strategy.

OAR strategy structure constructs (N-1) SVM models where N is the number of classes. The i-th SVM is trained with all of the examples in the i-th class with positive labels, and all other examples with negative labels. Another method: OAO constructs $\frac{N(N-1)}{2}$ classifiers where each one is trained on data from two classes. Then it uses a voting strategy: each binary classification is considered to be a voting where votes can be cast for all data points x - in the end point is designated to be in a class with maximum number of votes [28]. See the Figure 5, respectively shows the process of OAR and OAO that a multi-class classification problem can be viewed as a series of binary classification problem, one for each category.

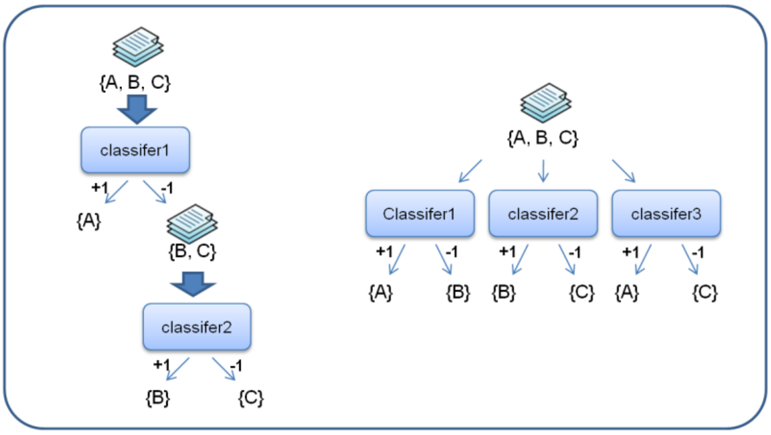


Figure 5. OAR vs. OAO

The LIBSVM use the OAO approach [**29**], so we can compare with Table 2 and Table 3 to see which approach is more fitting in our case. Both of these approaches we used word occurrences weighting be feature value, then also got the average F-measures respectively are 89.44% vs. 80.54%. So, we decided to adopt OAO approach by LIBSVM to handle multi-class problem.

Table 3. multi-class classification by OAR approach

Measur / Category	**Recall of Test Set**	**Precision of Test Set**
food	72% (18/25)	100% (18/18)
travel	64% (16/25)	80 % (16/20)
makeup	100% (25/25)	67.57% (25/37)

3.1.5 Article Categorization & Filtering

When collected certain amounts of dataset, we increased the number of classified categories and trained again. In the phased result shown in Table 4, SVM could still keep the accuracy up than 91%~95% (see Table **4**m). But with the increase of category, we found a fatal problem soon: in point of face, the blog articles are very diversified that unable to define its category completely. So it needs to judge whether the incoming article is belong in our defined categories or not before the classification process.

Table 4. F-measure of phased result

Category Number	**Feature Number**	**F-measure of Training set**	**F-measure of Test set**
5	422	95.2%	91.3%
6	495	94.86%	91.12%
7	564	95.59%	91.43%

At beginning, we regarded these articles as a new classification, collected 100 of amount and then add to SVM training, the result shown in Table **5**. We found that the average precision is less than 60% only by this way, and the difference between recall and precision of “other-category” there is over 30%. Apparently, this notion is bed reflected in the result. So we tried to use the originally classification algorithm to implement a filter which could pick out the articles beside our defined categories. Therefore, we translated this problem into binary classification problem. Likewise, selected each article features from pre-defined category and non-defined category as positive sample and negative sample, then solved by using the “filter feature set” to classified. By this way, we can get good result, which will show in Chapter 5 later.

Table 5. Result of phased result – other-category as normal category

Measur Category	Recall		Precision	
	Training Set	Test Set	Training Set	Test Set
food	92.0% (69/75)	84.0% (21/25)	95.84% (69/72)	91.31% (21/23)
travel	81.34% (61/75)	76.0% (19/25)	96.83% (61/63)	90.48% (19/21)
makeup	89.34% (67/75)	88.0% (22/25)	97.11% (67/69)	91.67% (22/24)
beauty salon	93.34% (70/75)	80.0% (20/25)	97.23% (70/72)	90.91% (20/22)
fashion	90.67% (68/75)	72.0% (18/25)	95.78% (68/71)	90.0% (18/20)
sentiment	89.34% (67/75)	72.0% (18/25)	95.72% (67/70)	100.0% (18/18)
3C	85.34% (64/75)	88.0% (22/25)	98.47% (64/65)	95.66% (22/23)
Avg.	88.77%	80.0%	96.72%	92.87%
other	98.67% (74/75)	96.0% (24/25)	62.72% (74/118)	48.98% (24/49)

Here we should specially note that the processes of feature selecting and adjusting between these two stages: categorizing and filtering were independent. Stated in another way, there were generated two different model files after training, respectively is *filter.model* and *category.model*, provided for predict procedure use.

3.1.6 2-Layer SVM Predict Scheme

In this section, we will show how to filter and categorize articles by the 2-layer SVM predict schemes. The flow of this scheme is shown in Figure 6.

At begining, the article went through segmentation engine to become the token term. First layer is for article filter. SVM predict procedure would turn the token term to feature vector in accordance with filter feature set by feature translation process. Using this feature vector and *filter.model*, SVM could predict positive class or negative class. If result was negative, it denoted this article was not categorical, so we classified as “other category”; that meant it was non-defined category. Contrarily, if result was positive, it should be deal with the second layer to predict the real category which we defined. By the same way, procedure turned the token term to feature vector in accordance with category feature set, and then used this feature vector and *category.model* to get the category number.

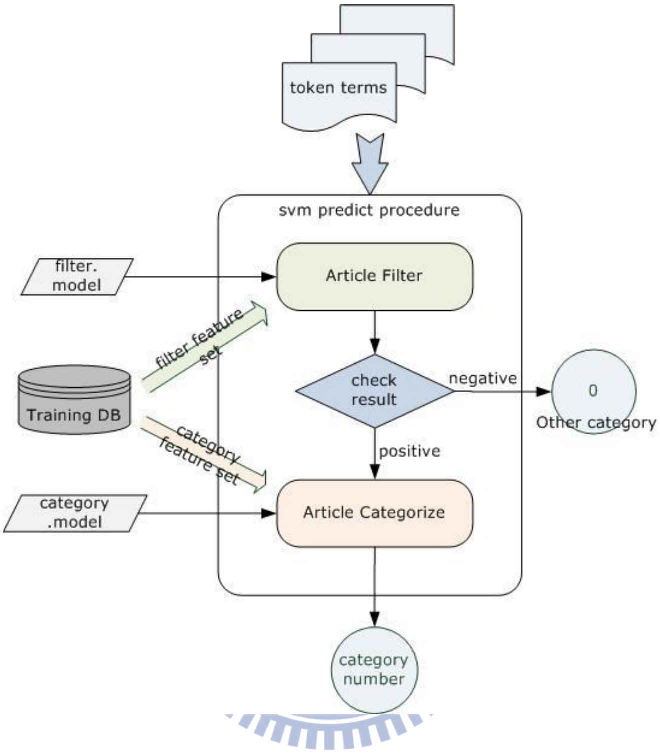


Figure 6. 2 – Layer SVM Predict Scheme

3.2 System Architecture Overview

3.2.1 Overall Architecture

Besides the classifier, we design several components for this system. Following are description:

- RSS Feed Handling – Difference with the RSS subscription usage nowadays, this system has abilities to provide classified articles from different RSS channels, just like a feed aggregation. So we will parse the RSS feed and divide them according to the article for the unit, then merges again based on the category.
- Article Full Text Extract – Some RSS grant summary feed; that means there is only partial-content of articles you encounter. Due to categorization, the system should own a process to extract the full text of articles.
- Segmentation Engine – We use the CKIP service [11] to implement a tokenizer, whose input is one article with its topic, and the output is a set of <token, times>, called token term.
- Feature Translation - Depend on this process, the feature term can be translated to one feature vector for SVM algorithm.
- Training Module - Before classifying, we should utilize the artificial pre-collecting training data to build an accurate predict rule. Along with the defined categories increases gradually, the system need a module design that is convenience to be able train repeatedly.
- Personalized RSS Reader – It's a simple RSS viewer which we could use to browse article by choosing different category. There existed a subscription

mechanism that reader could only receive the information which user selected.

Further implement detail will descriptive in the next chapter.

3.2.2 System Work Flow

After introducing architecture design, then narrate the overall flow, shown in Figure 7:

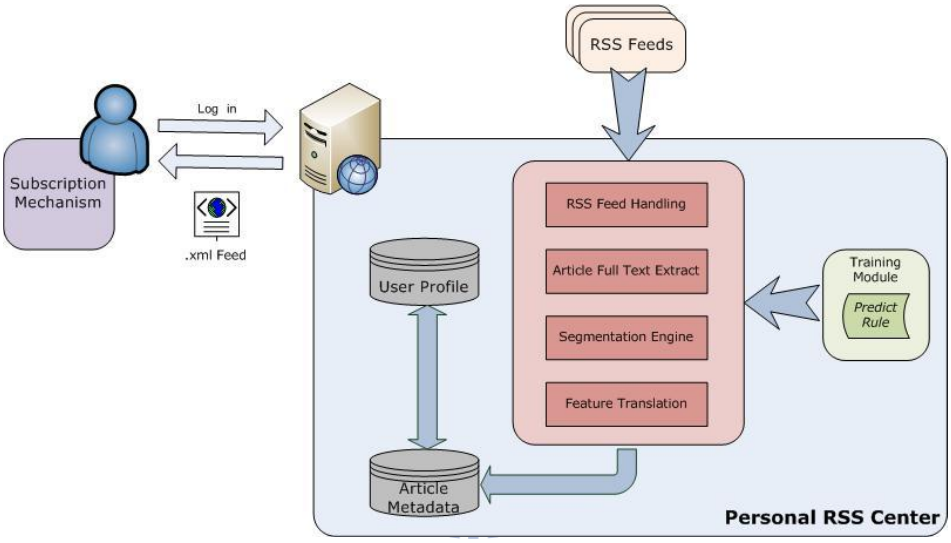


Figure 7. System Work Flow

Personalized RSS Center would access the muster of RSS feeds that all subscribers' channels per hours. First, these feeds after full text extracting process and so on segmenting process, then translating to a feature term. Second, system would predict the category of each articles according the predict rule that produced by training module. Finally, RSS feed handling process combine each RSS items (include title, URL link, and publish date…etc) according to the article for the unit, and then deposit these article entry to the RSS database.

After Personalized RSS Center processes, when one user synchronizes his subscription, the center would dynamic produce a personalized RSS feed. The content of this was aggregated article update information according as category the user chosen and published from the channels the user subscribed.

3.3 Personalized RSS Subscribe Service

The goal of personalized RSS subscribe service is split nowadays tradition RSS feeds then system automatically judged the category of article, and finally stored in system RSS database for the user subscribing according to the category primarily. There are architecture descriptions of server site and client application in below paragraph.

3.3.1 Personalized RSS Server Architecture

In the Personalized RSS Server, there contained primary subscribe service center, proxy server, segmentation engine, and article metadata database server.

Seeing the Figure 8, we might plainly see various machines are responsible for different task. As the flow we mention in Section 3.2.2, when subscribe service center accepting the new feeds information, it would first obtain the full text by proxy server depending on article URL link. The reason of proxy designed is some sort of bandwidth limiting system that BSP (e.g. Wretch of Yahoo) have put in place on their servers; so system switched off the proxy server to access these article pages during short sequential time via a different IP addresses. The segmentation engine implemented by CKIP [11] would give back one set of data which included a list format of <token term, token tag>. In whole server architecture, the subscribe service center was responsible for the tasks which were parsing the originally RSS feeds for proxy server, splitting full text of article to limit length string for segmentation engine, executing classification algorithm to get the category number, and store/access article information to/from RSS database server.

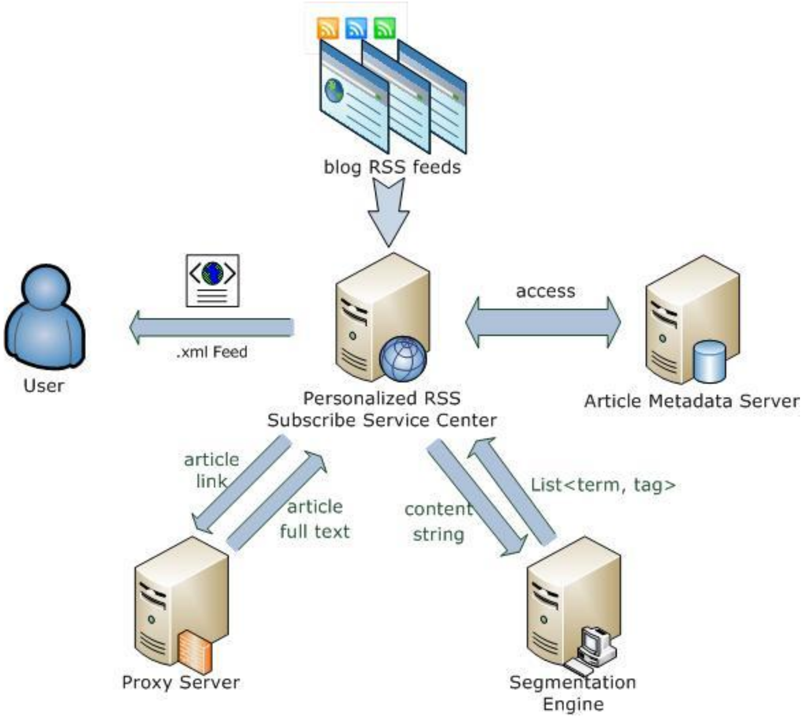


Figure 8. Personalized RSS Server Architecture

3.3.2 Personalized RSS Client Architecture

Relatively, now we see the architecture of client site. In the Personalized RSS Client Application, the reader could synchronize the subscription requirement by making a request to the subscribe service center. User first should define his option of channels he wants to subscribe and which article type of these channels. When server got the sync request, it would produce a feed which was xml form, and this feed was freash information after the user synchronized last time. During the synchronization process, the client application – personalized RSS reader would parse that feed which was from server, and then store in the local database. Different with common RSS reader in existence, the personalized RSS reader could show feeds by each article categories.

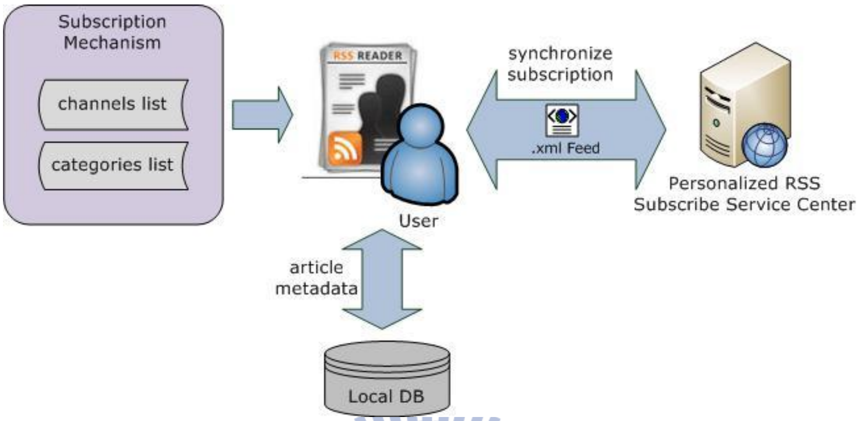


Figure 9. Personalized RSS Client Architecture

Chapter 4 Implement Detail

The system primarily consists of three parts, respectively are classification training, server center, and client application. Here will descriptive the work flow in various part, implement detail of major components which the previous chapter mentioned, and the tools this development used.

4.1 Classification Design

4.1.1 Training Module

Due to collect documents is consuming time, in addition, it's hard to define the whole categories of all kinds of articles at once, and we adopted one module that could train repeatedly to reach the purpose of classification. Through judging artificially, when the articles of some category were collected to certain amounts of, system added the training data part of these new articles to procedure trained again. Because of having more classified information, the original filter feature set and category feature set would extend. Furthermore, in order to keep the accuracy, we adjusted not only the arguments of our SVM classifier, but also the df-value threshold of feature selection and content of stop-word list and synonymy list.

In this way, whenever increased a definition category to classify, the system would produce the new prediction model file. After repeated adjustment, we expect the accuracy of classification would not get down because the number of category increases.

4.1.2 Article Categorization Flow

Figure 10 shows the flowchart of article categorization in this training module.

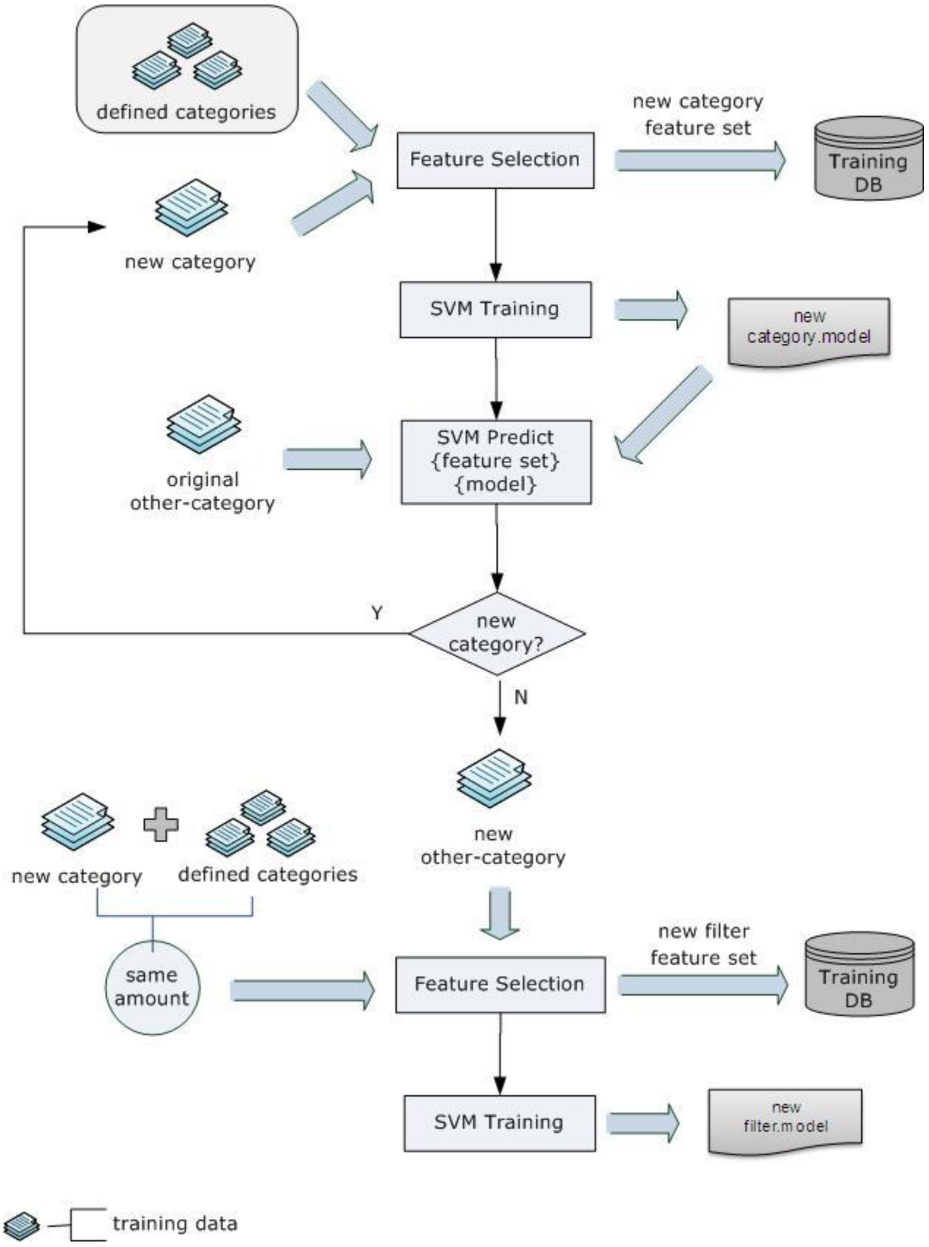


Figure 10. Flow of Article Categorization

When incoming new documents of new category, we first extracted new category feature set, and trained these articles which belong to original plus new one categories.

Thus we would get the new *category.model*. Then utilized it to predict again these already-classified articles belong to original other-category. Here we usually double checked by judging artificially. Purpose of this step is to eliminate those articles are in face should be belonged to the new category from original other-category. After that, we chosen the documents of the same amount from this new category and each pre-defined category; and then could produce one set of new filter feature along same procedure, trained as bi-selector again, and got the new *filter.model* finally.

4.2 Personalized RSS Center

4.2.1 Personalized RSS Server Flow

At regular interval, the Personalized RSS Server would proceed on this flow shown in Figure 11 next page. First at all, center got the system last update time so that could know which feed content should be handled. After obtain feed channel to tabulate in database, center accessed its RSS feed entries in order, then consigned to feed handling procedure to deal with.

Follow-up stage looped that regard feed entry as unit until storing to RSS database server. The roughly procedure is that: got feed elements by feed handling process, passed article link to full text extract process, then assigned part of full article content to segmentation engine in turn. As soon as producing a token term, system delivered it to feature translation process, and then utilized the feature vector which was changed by feature set to predict category number. Among the predict process, it was 2-layer SVM scheme that we introduce in 3.1.6.

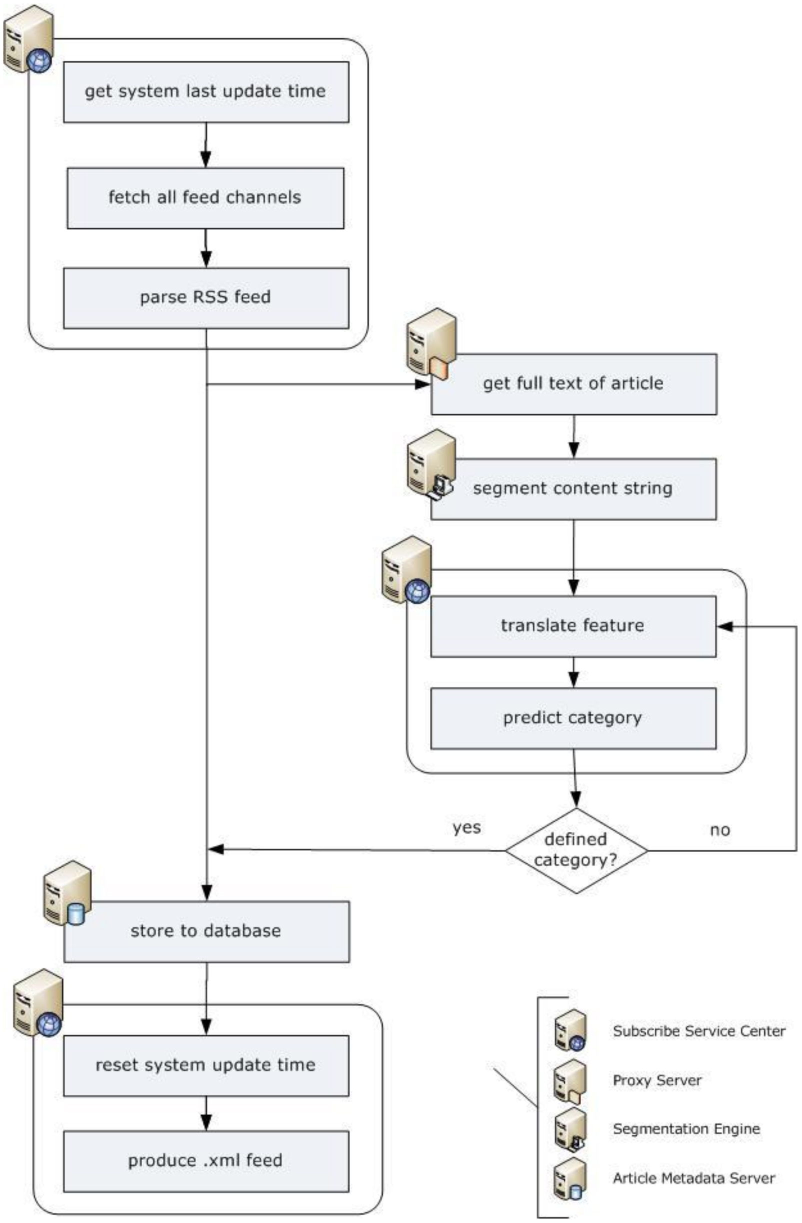
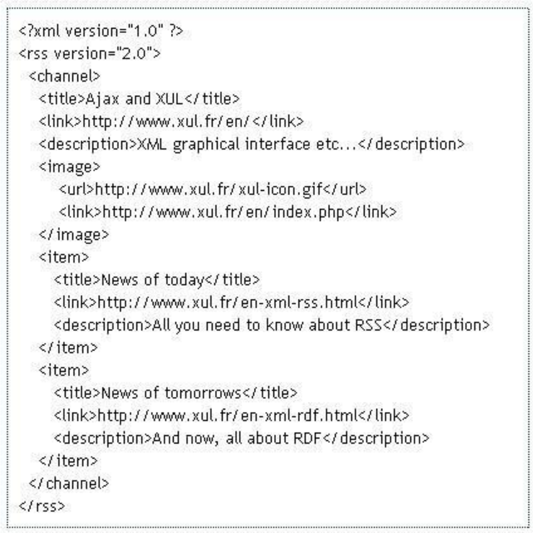


Figure 11. Flow of Personalized RSS Server

Eventually, system store the feed information include: link, title, publish data, describe, author, and category number by article for a unit to RSS database. And at last step, reset the time at the flow beginning as system update time. Whenever received the request from user, the system would access article information from RSS DB Server to produce one personalized feed that was according the user profile.

4.2.2 RSS Feed Handling

In this part, we used the ROME API[8] to parse RSS feeds. ROME is a set of open source Java tools for parsing, generating and publishing RSS and Atom feeds.



```
<?xml version="1.0" ?>
<rss version="2.0">
 <channel>
  <title>Ajax and XUL</title>
  <link>http://www.xul.fr/en/</link>
  <description>XML graphical interface etc...</description>
  <image>
     <url>http://www.xul.fr/xul-icon.gif</url>
     <link>http://www.xul.fr/en/index.php</link>
  </image>
  <item>
    <title>News of today</title>
    <link>http://www.xul.fr/en-xml-rss.html</link>
    <description>All you need to know about RSS</description>
  </item>
  <item>
    <title>News of tomorrows</title>
    <link>http://www.xul.fr/en-xml-rdf.html</link>
    <description>And now, all about RDF</description>
  </item>
 </channel>
</rss>
```

Figure 12. Sample of RSS document

Reference the Figure 12. When accepting a RSS feed, this process would divide each item to fetch its sub tag contents by ROME. Certainly, string format of publish date was converted to time format, in order to judge it for the new contents produced after the system update time. Then these contents were stored in an object that was declared as Array List format to continue the next process.

4.2.3 Article Full Text Extraction

The full text extraction process supports three BSP: Wretch, Pixnet, and Xuite. After observing html form of these webpage, we only extracted the article content from all page information.

[8] https://rome.dev.java.net/

Here we have designed one group proxy pool in order to offer a lot of IP addresses used for accessing the article links. By random switching different proxy server, this extract produce could avoid the problem of excessively frequent access webpage information so that was regarded as DoS (Denied of Service).

By the cause of restriction on the CKIP, we deleted the meaningless symbols, and cut the article into the length of part content string that CKIP limit. Then we could make use of segmentation process as batch.

4.2.4 Segmentation Engine

The segmentation engine was implemented by an open source library: ckip-client[9]. This library collocated dom4j[10] to offer developers a convenient way to use Chinese Knowledge and Information Processing (CKIP) [11] service. We just needed assign the CKIP server IP, CKIP server port, user name, and password in our program code, and then we could get the segment term and its tag by sending the request content string. Due to the academic cooperation between Academia Sinica and National Chiao Tung University, we could get more explicit tag [Appendix 1] than on-line CKIP service. It's helpful not only on feature selection of classification method development, but also on sieving the stop word out and comparing synonymy.

After getting the segment result, we declared a data structure of hash table to put the data of token term which included the token word and it's appear times. As we see in following Figure 13. The procedure picked the normal nouns (*Na*) and place names (*Nc*) by CKIP service, and furthermore precede two audits: ignored stop word and replaced synonymy to filter quantity of tokens.

[9] http://ckipclient.sourceforge.net/

[10] http://www.dom4j.org/

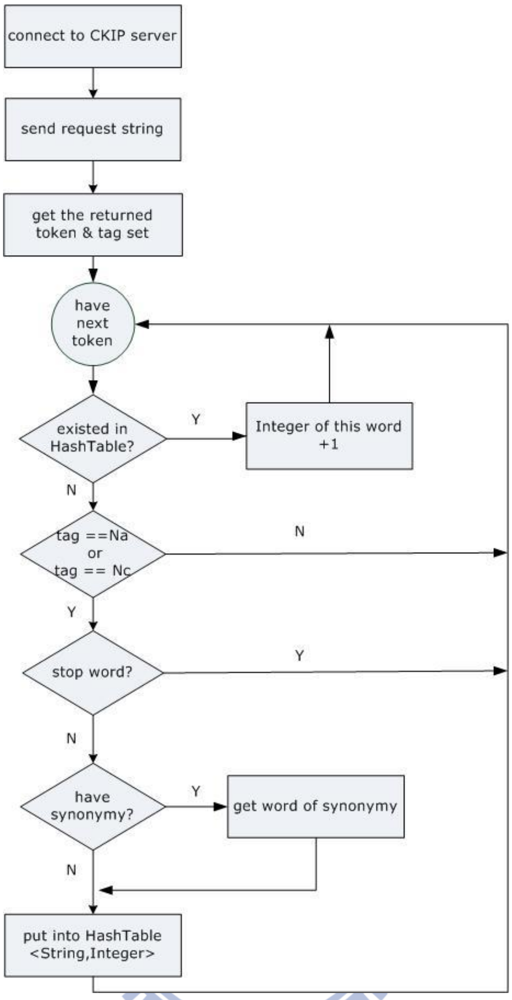


Figure 13. Flow of Segmentation Engine

4.2.5 Feature Translation

This main purpose of this procedure is that translate the token term to the vector representation which SVM could accept according as the pre-defined feature set. After getting the token term from segmentation engine, we picked out the words that also existed in feature set then assigned feature value as normalized appear times.

4.3 Personalized RSS Application

4.3.1 Subscription Mechanism

Personalized RSS Center has a profile data which include the members' option information. When the user connected to the center through this reader, he could add/delete his subscribe channels and modified his selection of categories immediately. By this subscription mechanism, center could provide personalized subscribe service, so that users would receive the update information of articles which were not only published from their subscription channel, but the content also was the type they really interested in.

4.3.2 Personalized RSS Reader

Our personalized RSS reader is a desktop application which implemented by JAVA, it requests that a Java Runtime Environment be in end-users' machine. The area of RSS viewer we used the jdic[11] library to offer a light and simple embedded web browser. User could visit the website of articles they want to read more detail.

The workflow within the reader is shown as Figure 14. User need to set up the relevant options of subscription mechanism at first using time. Evert time to synchronize, reader would send the information: user id and last sync time, which let Subscribe Service Center could produce the personalized RSS feed which include update information during last sync time and current time. The time is based on server time as standard.

[11] https://jdic.dev.java.net/

When After receiving the RSS feed from center, reader would parse this feed into local database which is a relatively small C programming library: SQLite[12]. And, the procedure of parsing is similar to RSS Feed Handling we mentioned in Section 4.2.2 at server site. In the last step, reader would reset the sync time which sends from center.

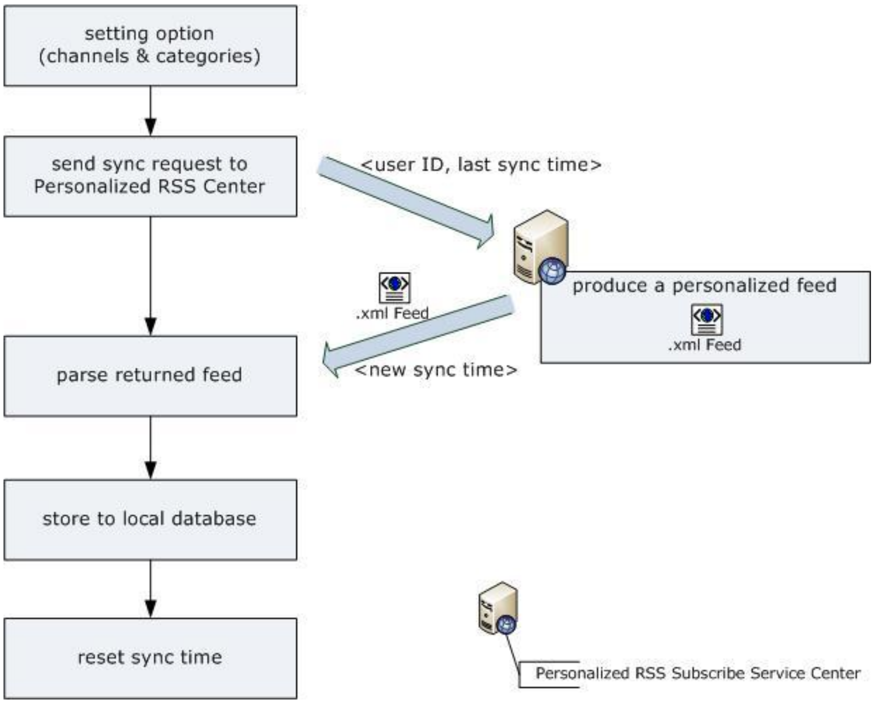


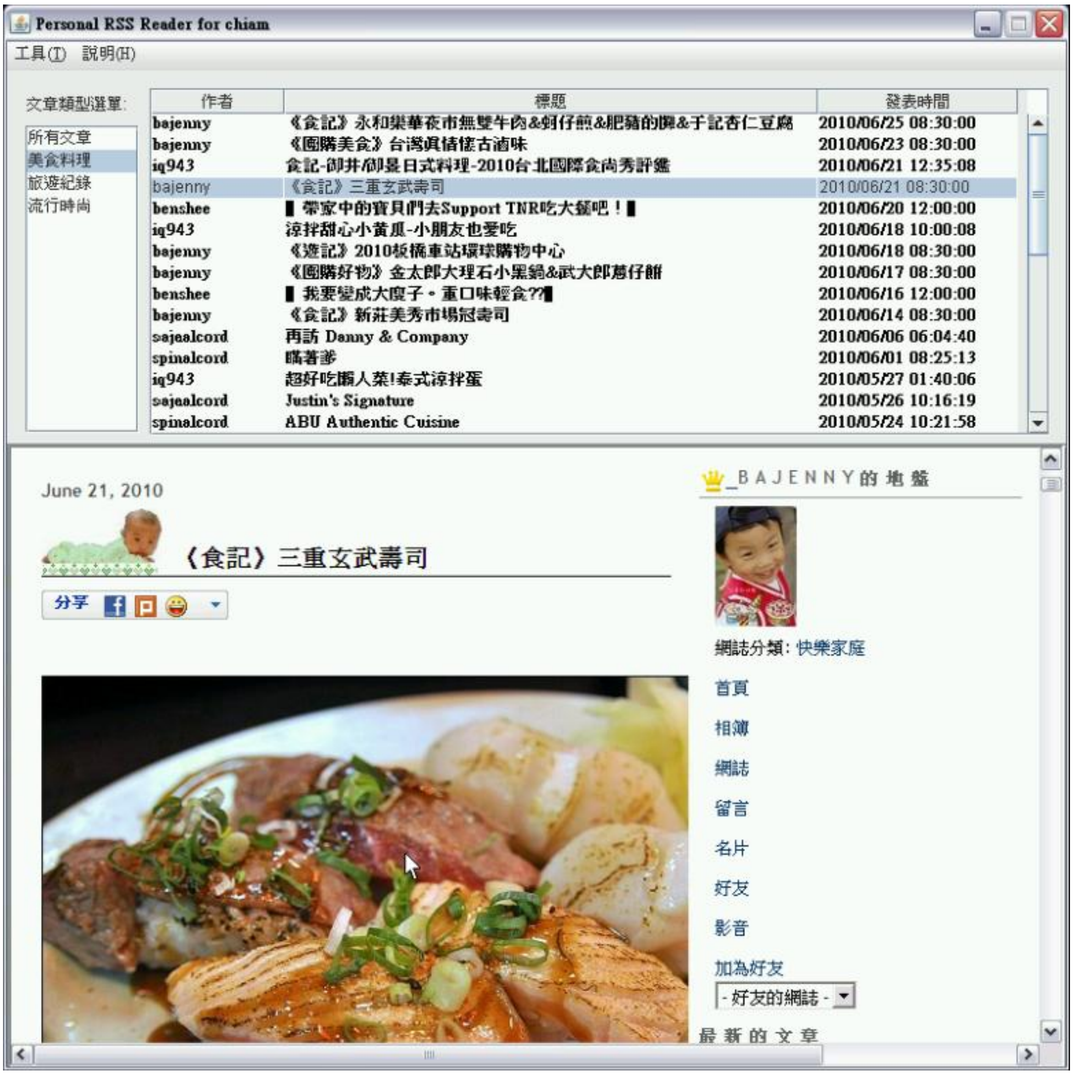
Figure 14. Flow of Personalized RSS Client

4.3.3 Application Demonstration

We put the necessary objects of client into a package. After downloading and setting his subscription mechanism, user can easily use it to read RSS which content are they interesting in. Different with ordinary RSS reader, users search article not that by choosing channel but that by selecting category.

[12] http://www.sqlite.org/

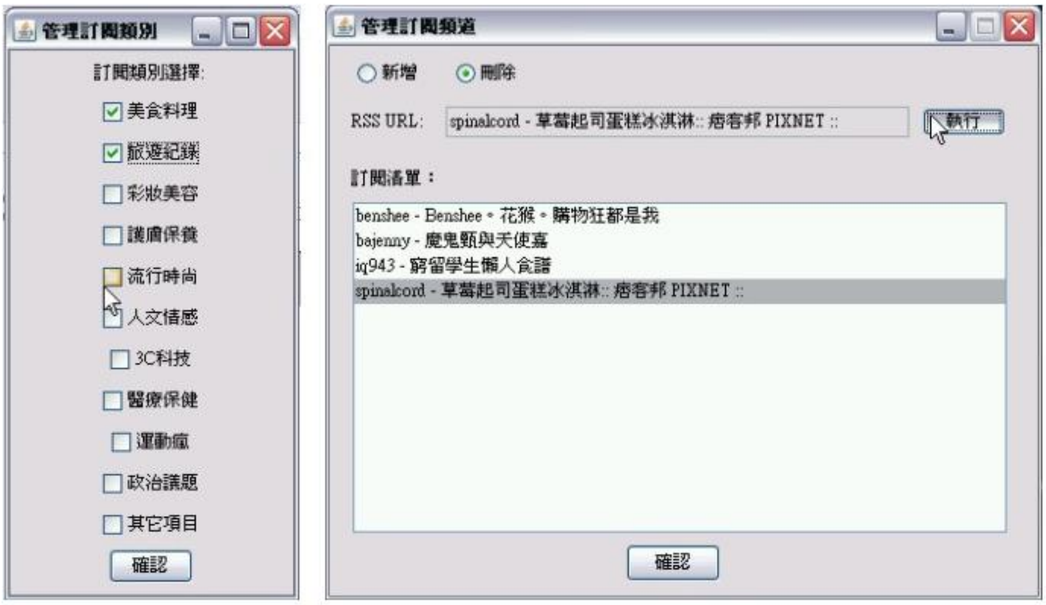
Here is a simple demonstration about our personalized RSS reader shown in Figure 15. There are three function blocks: the upper left area category list, when user choose one of these, the upper right area will show all article entries which belong to this chosen category, and reader can sort by author, topic or publish date from the table headline. The underside is an embedded browser, if reader clicks one entry of article list, the web page of this article link will show in this area, and the information of this entry in article list will be turned from bold type into the normal type, it represented user already read.



作者	標題	發表時間
bajenny	《食記》永和樂華夜市無雙牛肉&蚵仔煎&肥豬的脾&于記杏仁豆腐	2010/06/25 08:30:00
bajenny	《團購美食》台灣眞情懷古滷味	2010/06/23 08:30:00
iq943	食記-御丼/御曼日式料理-2010台北國際食尚秀評鑑	2010/06/21 12:35:08
bajenny	《食記》三重玄武壽司	2010/06/21 08:30:00
benshee	▌帶家中的寶貝們去Support TNR吃大餐吧！▌	2010/06/20 12:00:00
iq943	涼拌甜心小黃瓜-小朋友也愛吃	2010/06/18 10:00:08
bajenny	《遊記》2010板橋車站環球購物中心	2010/06/18 08:30:00
bajenny	《團購好物》金太郎大理石小黑鍋&武大郎蔥仔餅	2010/06/17 08:30:00
benshee	▌我要變成大腹子。重口味輕食??▌	2010/06/16 12:00:00
bajenny	《食記》新莊美秀市場冠壽司	2010/06/14 08:30:00
sajealcord	再訪 Danny & Company	2010/06/06 06:04:40
spinalcord	瞞著爹	2010/06/01 08:25:13
iq943	超好吃膩人菜!泰式涼拌蛋	2010/05/27 01:40:06
sajealcord	Justin's Signature	2010/05/26 10:16:19
spinalcord	ABU Authentic Cuisine	2010/05/24 10:21:58

Figure 15. GUI of Personalized RSS Reader

Furthermore, reader can use the menu bar to synchronize feed and set category list and channel list. Every time reader synchronizes to server, the article list will refresh the newly-increased information. The following figure shows the interface of setting, by this, subscribers can get the result of choosing the author as well as choosing the categories.



管理訂閱類別

訂閱類別選擇:

☑ 美食料理
☑ 旅遊紀錄
□ 彩妝美容
□ 護膚保養
□ 流行時尚
□ 人文情感
□ 3C科技
□ 醫療保健
□ 運動瘋
□ 政治議題
□ 其它項目

確認

管理訂閱頻道

○ 新增 ● 刪除

RSS URL: spinalcord - 草莓起司蛋糕冰淇淋:: 痞客邦 PIXNET ::

執行

訂閱清單：

benshee - Benshee。花猴。購物狂都是我
bajenny - 魔鬼甄與天使嘉
iq943 - 窮留學生懶人食譜
spinalcord - 草莓起司蛋糕冰淇淋:: 痞客邦 PIXNET ::

確認

Figure 16. GUI of Category List & Channel List Setting

4.4 Tools and Libraries

We use Java platform to develop this system. There are numbers of open source libraries for development. As shown below.

Table 6. Tools and Libraries

Name	Usage	Version
Java Servlet	Personalized RSS generator Subscribe Service Center	1.2.2
Tomcat	Web server	6.0
MySQL	Database of Subscribe Service Center & RSS DB Server	6.0
SQLite	Database of Personalized RSS Reader	3.6.14.2
LIBSVM[13]	SVM classifier	2.9
CKIP Client	Client of CKIP online service	0.4
dom4j	Parse XML document by CKIP Client	1.6.1
rome	Parse, generate and publish RSS	0.9
jdom[14]	Handle XML document	1.1
jdic	Embedded web browser of RSS reader	0.9.5

[13] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[14] http://www.jdom.org/

Chapter 5 Results and Analysis

The rule to refer to the Section 3.1.1, we finally collected 10 popular categories, respectively are food, travel, makeup, beauty salon, fashion, sentiment, 3C, health, sport, politic, and plus the other-category.

5.1 Experimentation Result

Here we compare two statuses of blog article categorization method; one is single layer of SVM classification, another is our 2-layer SVM scheme in this research. First, we see the result of categorization by SVM without irrelevant articles as below, it has both good recall and precision which average higher than 90%. There among, df-value threshold is 0.1, and there are 768 features.

Table 7. Result of categorization by SVM without irrelevant articles

Measur Category	Recall		Precision	
	Training Set	Test Set	Training Set	Test Set
food	96.0% (72/75)	92.0% (23/25)	92.31% (72/78)	88.47% (23/26)
travel	94.67% (71/75)	92.0% (23/25)	86.59% (71/82)	92.0% (23/25)
makeup	92.0% (69/75)	92.0% (23/25)	92.0% (69/75)	92.0% (23/25)
beauty salon	93.34% (70/75)	88.0% (22/25)	92.11% (70/76)	88.0% (22/25)
fashion	93.34% (70/75)	92.0% (23/25)	93.34% (70/75)	92.0% (23/25)
sentiment	94.67% (71/75)	92.0% (23/25)	89.88% (71/79)	92.0% (23/25)
3C	93.34% (70/75)	100.0% (25/25)	95.9% (70/73)	92.6% (25/27)
health	89.34% (67/75)	88.0% (22/25)	94.37% (67/71)	91.67% (22/24)
sport	92.0% (69/75)	96.0% (24/25)	98.58% (69/70)	96.0% (24/25)
politic	94.67% (71/75)	92.0% (23/25)	100.0% (71/71)	100.0% (23/23)
Avg.	93.34%	92.41%	93.51%	92.48%

However, when considering the irrelevant articles (has 603 of quantity so far), if we regarded those articles as a new category: other-category, and used the same way to classify; no matter how adjust the parameters (df-value threshold, stop word list, and synonymous word list), the result is unsatisfactory (shown in Table 8). We found the recall of each pre-defined categories descend; most erroneous judgments were all judged into the "other-category". That's why the precision of other-category is so low. And so, subscriber might receive too many update information of these uninterested articles.

Table 8. Result of categorization by single-layer of SVM

Measur Category	**Recall**		**Precision**	
	Training Set	Test Set	Training Set	Test Set
food	81.34% (61/75)	76.0% (19/25)	91.05% (61/67)	90.48% (19/21)
travel	72.0% (54/75)	64.0% (16/25)	91.53% (54/59)	88.89% (16/18)
makeup	84.0% (63/75)	72.0% (18/25)	94.03% (63/67)	90.0% (18/20)
beauty salon	78.67% (59/75)	72.0% (18/25)	96.73% (59/61)	90.0% (18/20)
fashion	81.34% (61/75)	80.0% (20/25)	96.83% (61/63)	95.24% (20/21)
sentiment	85.34% (64/75)	80.0% (20/25)	95.53% (64/67)	100.0% (20/20)
3C	81.34% (61/75)	84.0% (21/25)	98.39% (61/62)	100.0% (21/21)
health	77.34% (58/75)	64.0% (16/25)	96.67% (58/60)	88.89% (16/18)
sport	81.34% (61/75)	84.0% (21/25)	96.83% (61/63)	100.0% (21/21)
politic	85.34% (64/75)	80.0% (20/25)	100.0% (64/64)	100.0% (20/20)
Avg.	80.81%	75.6%	95.76%	94.35%
other	98.24% (444/452)	98.68% (149/151)	78.04% (444/569)	74.13% (149/201)

This situation shows this method could not determine efficaciously between those irrelevant articles and that articles we have already defined its category. The possible reason is that these irrelevant articles have features too much and ambiguous, we couldn't classify this "other-category" and the category we pre-defined with the same unit.

Therefore, we propose the 2-layer SVM scheme to filter out those irrelevant articles before real categorization. Now see the result of filtering and whole categorization under this scheme separately. Table 9 shows the effect of first layer. Here, we used the 1000 documents for the "defined-category" and 603 documents for "other-category" to be classified by SVM as bi-selector, or we called it as filter. Then, we put those remained articles to the second layer for real classify. Through adjusting and experimenting, we got acceptable result of filtering. In which, df-value threshold is 0.035, and there are 556 features.

Table 9. Result of filter by 2-layer SVM scheme

Measur Type	**Recall**		**Precision**	
	Training Set	Test Set	Training Set	Test Set
defined-category	88.53% (664/750)	87.2% (218/250)	96.93% (664/685)	95.2% (218/229)
other-category	96.24% (435/452)	95.37% (144/151)	84.14% (435/517)	83.73% (144/172)

Now, we used the filter rule which was created by the dataset based on Table 9 to measure whole dataset. Table 10 shows the macro-effect of categorization by 2-layer SVM. In the second layer, we keep df-value threshold as same as categorization without irrelevant articles at the outset we mentioned. Compare with Table 8, it had obviously outperformance of recall, but the behavior of precision decline slightly.

Table 10. Result of categorization by 2-layer SVM scheme

Measure Category	Recall		Precision	
	Training Set	Test Set	Training Set	Test Set
food	86.67% (65/75)	88.0% (22/25)	90.28% (65/72)	95.66% (22/23)
travel	84.0% (63/75)	88.0% (22/25)	95.46% (63/66)	91.67% (22/24)
makeup	86.67% (65/75)	88.0% (22/25)	97.02% (65/67)	91.67% (22/24)
beauty salon	86.67% (65/75)	84.0% (21/25)	94.21% (65/69)	91.31% (21/23)
fashion	88.0% (66/75)	92.0% (23/25)	95.66% (66/69)	95.84% (23/24)
sentiment	90.67% (68/75)	88.0% (22/25)	94.45% (68/72)	100.0% (22/22)
3C	93.34% (70/75)	88.0% (22/25)	97.23% (70/72)	95.66% (22/23)
health	88.0% (66/75)	80.0% (20/25)	98.51% (66/67)	95.24% (20/21)
sport	85.34% (64/75)	88.0% (22/25)	98.47% (64/65)	100.0% (22/22)
politic	86.67% (65/75)	88.0% (22/25)	98.49% (65/66)	95.66% (22/23)
Avg.	87.6%	87.2%	95.98%	95.28%
other	96.24% (435/452)	95.37% (144/151)	84.14% (435/517)	83.73% (144/172)

5.2 Experiment Discussion

More carefully, here we show the comparison between single-layer SVM & 2-layer SVM (Figure 17 & Figure 18) separately with the chart.

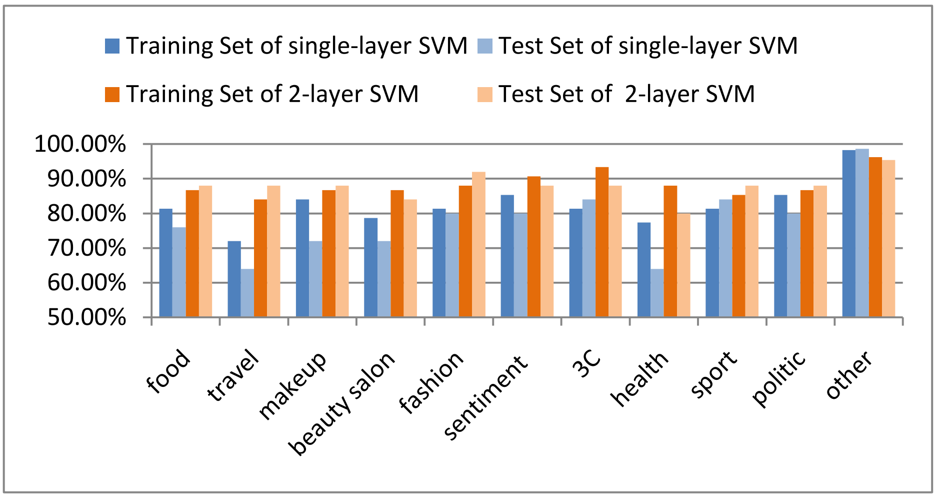


Figure 17. Recall Comparison between single-layer & 2-layer SVM

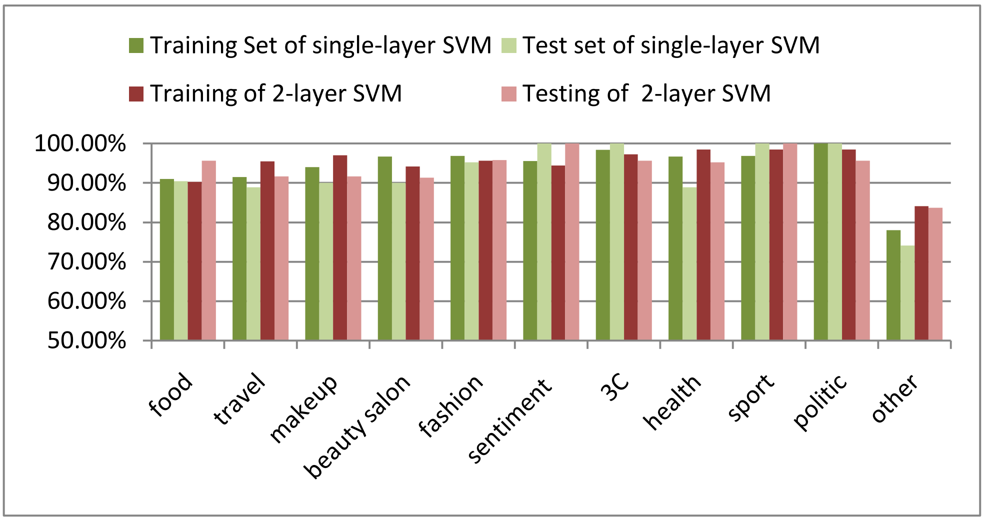


Figure 18. Precision Comparison between single-layer & 2-layer SVM

The recall value represents that in all expectable articles, how many can be obtained; the precision value represents that how many articles which we obtained are correct. We found the “other-category” influence accuracy of else 10 defined categories. For filter this other-category, besides its all recall declined. If we used single-layer SVM, the precision of filter had only less than 80%.

In contrast, if used 2-layer SVM, it can upgraded both more than 6% of precision and recall. Even the precision values of else 10 defined categories are lower slightly than single-layer SVM, but the behavior of recall values are much better. This is what we expect more.

Table 11. Comparison between single-layer & 2-layer SVM

Measure SVM Type | Categ		**Recall**		**Precision**	
		Training Set	Test Set	Training Set	Test Set
single-layer SVM	10 defined-category	80.81%	75.6%	95.76%	94.35%
	other-category	98.24%	98.68%	78.04%	74.13%
2-layer SVM	10 defined-category	87.6%	87.2%	95.98%	95.28%
	other-category	96.24%	95.37%	84.14%	83.73%

5.3 System Usability Scale (SUS)

After the system is finished, we have done a simple assessment for 15 users. Here we use a system usability scale developed by Digital Equipments Co Ltd of Britain in 1986. The basic sample is shown in [Appendix 2], and we have done some adjustment to the question 5. Following are ten items of our test:

1. I think that I would like to use this system frequently.

2. I found the system unnecessarily complex.

3. I thought the system was easy to use.

4. I would need the support of a technical person to be able to use this system.

5. I found the categorized result in this system were well sensible.

6. I thought there was too much inconsistency in this system.

7. I would imagine that most people would learn to use this system very quickly.

8. I found the system very cumbersome to use.

9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

Each item's score contribution will range from 0 to 4. For items 1,3,5,7 and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU. Through calculating the SUS score, we got the highest score is 87.5, the lowest is 65, and the average is 76.33 finally.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

In many research of classification topic, scholars used different way with different classification methods which we mentioned in section 2.2. However, these researches focused on the materials of specific type. When we meet the anticipated materials type, we could not certainly use the same way to classify. In this paper, we studied the novel approach of classification: 2-layer SVM schema to filter the non-defined articles and classify the pre-train defined articles. And we also design a training module that we can train repeatedly when the amount of category increase. Via these two mechanisms, we have avoided the problem of erroneous judgments that cause of anticipated materials.

We applied such a classification mechanism to RSS feed. After fetching full text by article URL from feed metadata, system predicted its category. Thus RSS subscribers could utilize to choose interested top by category.

Through experiment, we got the result which is around 87.5% recall and 95.5% precision. That means the update information from RSS which subscribers got are mostly according with real category, but subscribers may lose about 12.5% of the newer information they want.

6.2 Future Work

For further extension, we settle several challenges in the nearest future.

- Except content of the article, there probably exists some factors can be referred for judging category. In some researches, scholars consider social tag [30], sentiment orientation [31] or other objects for article classification. By using more features to achieve higher categorization accuracy.
- Our training module is executed offline artificially, include collect training set and train the predict model. Maybe we can rely on user feedback to modify the existing dataset then train automatically. With more training documents and constant training, improve predicting stability.
- Different people may have different viewpoint of category on the same article. Even if the system can predict 100% accuracy of classification result, user may not be certain to think this is a satisfactory result. So we should conjoin system definition and user cognizance, design a mechanism to offer user a personalized option of categories.
- In fact, an article may include more than one characteristics of category. We can consider using several binary classifiers to label multi-categories for one article [**32**].

Reference

[1] Libby, Dan, "RSS 0.91 Spec, revision 3," Netscape Communications Retrieved, 2007. http://web.archive.org/web/20001204093600/my.netscape.com/publish/formats/rss-spec-0.91.html

[2] Oh-Woog Kwon, Jong-Hyeok Lee, “Web Page Classification Based on K-Nearest Neighbor Approach,” Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages (IRAL), pp.9-15, 2000.

[3] Zhongda Lin, Kun Deng, Yanfen Hong, "Research of Web Pages Categorization," IEEE International Conference on Granular Computing (GRC), pp.691, 2007.

[4] J-Z Liang, "SVM based Chinese Web page automatic classification," International Conference on Machine Learning and Cybernetics (ICMLC), Vol.4: 2265 - 2268 , 2004.

[5] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, George Paliouras, Constantine D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering," Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, Barcelona, Spain, pp.9-17, 2000.

[6] Yiming Yang , Tom Ault , Thomas Pierce , Charles W. Lattimer, "Abstract Improving Text Categorization Methods for Event Tracking," Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2008.

[7] Yiming Yang and Xin Liu, “A Re-Examination of Text Categorization Methods,” Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp.42-49, 1999.

[8] 曾元顯,“文件主題自動分類成效因素探討”,中國圖書館學會會報,第68期,頁62-83，2002年。

[9] Jia-liang Kau, “A Study and Implementation of News Event Clustering and Summarization Search Engine,” National Yunlin University of Science and Technology, Master, 2004.

[10] Wen-Feng Wu, “Design and Implementation of a Classifier for Chinese E-mails,” Feng Chia University, Master, 2002.

[11] Team of Chinese Knowledge Information Processing(CKIP) at Academia Sinica, Taiwan. http://ckipsvr.iis.sinica.edu.tw/

[12] Wei-Yun Ma, Keh-Jiann Chen, "A Bottom-up Merging Algorithm for Chinese Unknown Word Extraction," Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Volume 17, pp.31-38, 2003.

[13] Gerard Salton, Michael J. McGill, “*Introduction to Modern Information Retrieval*,” McGraw Hill Inc , 1983.

[14] Christopher D. Manning, Hinrich Schutze, “*Foundations of Statistical Natural Language Processing*,” MIT Press, 2000.

[15] 數位時代，“*2010台灣網站100強*”，數位時代，第190期，2010年。

[16] James Tin-yau Kwok, ”Automatic Text Categorization Using Support Vector Machine,” Proceedings of the International Conference on Neural Information Processing (ICONIP), pp.347-351, 1998.

[17] Sheng-li Ji, Bo Li, "Chinese Text Categorization Algorithm Based on SVM," Journal of Chongqing Institute of Technology(Natural Science), Volume.22(7), 2008

[18] Jiu-Zhen Liang, "SVM Based Chinese Web Page Automatic Classification," International Conference on Machine Learning and Cybernetics (ICMLC), Volume 4, pp.2265 - 2268 , 2004.

[19] Ling Xia, Zhi Teng, Fuji Ren, "Question Classification in Chinese Restricted-Domain Based on SVM and Domain Dictionary," International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), pp.1 - 6, 2008.

[20] A.Basu, C.Watters, M.Shepherd, “Support Vector Machines for Text Categorization,” Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS), Volume 4, pp.103, 2003.

[21] Corinna Cortes, Vladimir Vapnik, “Support-Vector Networks,” Machine Learning, Volume 20 (3), pp.273-297, 1995.

[22] Thorsten Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", Proceedings of European Conference on Machine Learning (ECML), pp.137 - 142, 1998.

[23] Thorsten Joachims, "*Transductive Inference for Text Categorization Using Support Vector Machines,*" Proceedings of 16th International Conference on Machine Learning (ICML), Morgan Kaufmann Publishers Inc., pp.200-209, 1999.

[24] Jason D. M. Rennie, Ryan Rifkin, "Improving Multi-Class Text Classification with the Support Vector Machines," Master's thesis, Massachusetts Institute of Technology, 2001.

[25] Friedhelm Schwenker, "Hierarchical Support Vector Machines for Multi-Class Pattern Recognition," Proceedings of IEEE 4th International Conference on Knowledge-Based Intelligent Engineering System and Allied Technologies, pp. 561-565, 2000.

[26] Zhijie Liu, Xueqiang Lv, Kun Liu, Shuicai Shi, "Study on SVM Compared with the other Text Classification Methods," Proceedings of Second International Workshop on Education Technology and Computer Science, Volume 1, pp.219-222, 2010.

[27] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, "A Practical Guide to Support Vector Classification," http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf, 2003.

[28] Chih-Wei Hsu; Chih-Jen Lin, "A Comparison of Methods for Multiclass Support Vector Machines," IEEE transactions on Neural Networks, Volume 13, No.2, pp.415-425, 2002.

[29] Chih-Chung Chang, Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines," Technical report, Computer Science and Information Engineering, National Taiwan University, http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001-2004.

[30] Sadegh Aliakbary, Hassan Abolhassani, Hossein Rahmani, Behrooz Nobakht, "Web Page Classification Using Social Tags," Proceedings of the 2009 International Conference on Computational Science and Engineering, Volume 4, pp.588-593, 2009.

[31] Fazel Keshtkar, Diana Inkpen, "Using Sentiment Orientation Features for Mood Classification in Blogs," Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), pp.1-6, 2009.

[32] Nello Cristianini,John Shawe-Taylor, *“An introduction to support vector machines and other kernel-based,“* Cambridge University Press, 2000.

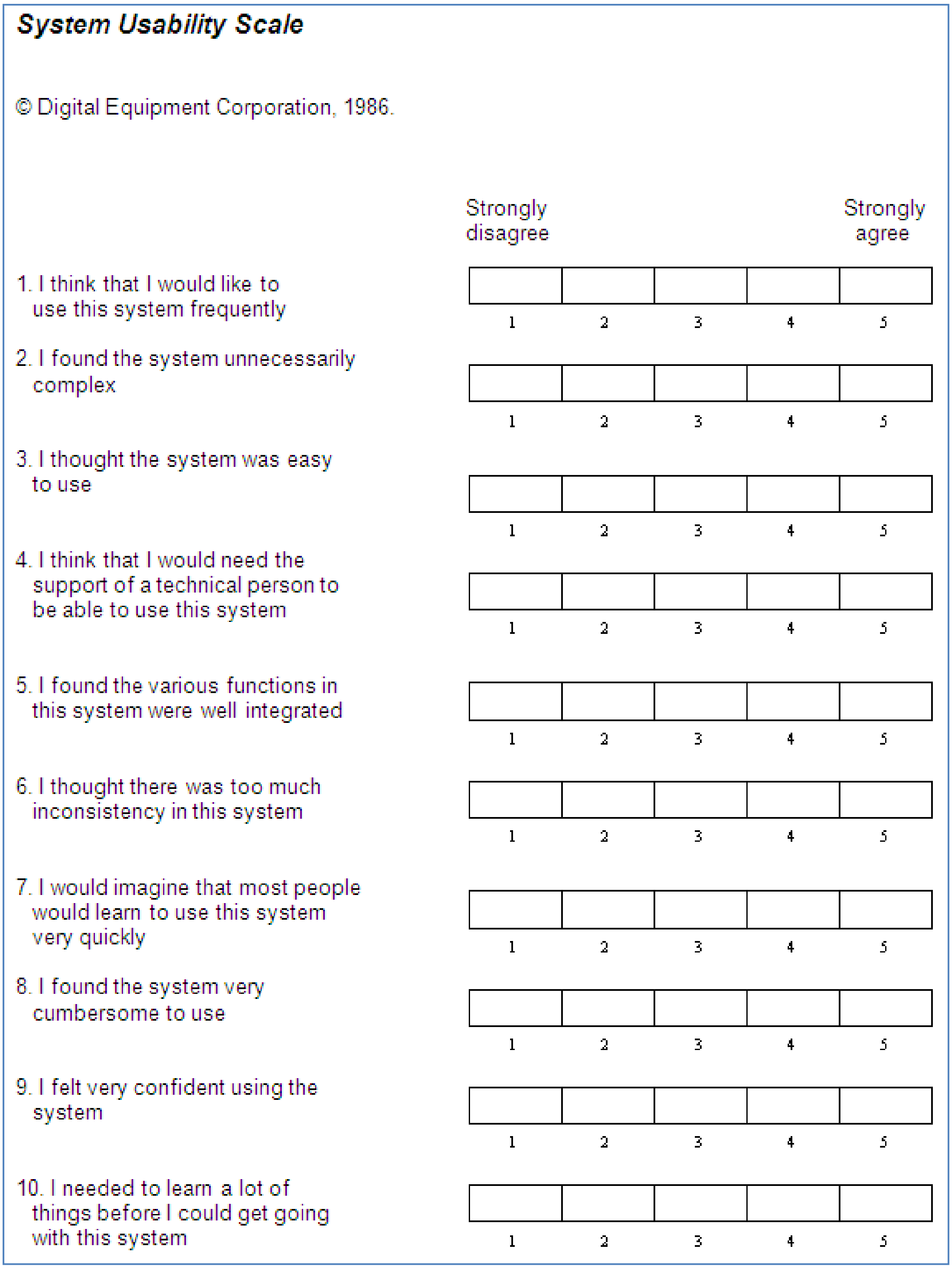
Appendix

Appendix 1. Chinese Speech Tag List by Academia Sinica, Taiwan.

Simple Tag	Anaphoric CKIP Speech Tag[15]	
A	A	/*非謂形容詞*/
Caa	Caa	/*對等連接詞，如：和、跟*/
Cab	Cab	/*連接詞，如：等等*/
Cba	Cbab	/*連接詞，如：的話*/
Cbb	Cbaa, Cbba, Cbbb, Cbca, Cbcb	/*關聯連接詞*/
Da	*Daa*	/*數量副詞*/
Dfa	Dfa	/*動詞前程度副詞*/
Dfb	Dfb	/*動詞後程度副詞*/
Di	Di	/*時態標記*/
Dk	Dk	/*句副詞*/
D	*Dab*, Dbaa, Dbab, Dbb, Dbc, Dc, Dd, Dg, Dh, Dj	/*副詞*/
Na	Naa, Nab, Nac, Nad, Naea, Naeb	/*普通名詞*/
Nb	Nba, Nbc	/*專有名稱*/
Nc	Nca, Ncb, Ncc, Nce	/*地方詞*/
Ncd	Ncda, Ncdb	/*位置詞*/
Nd	Ndaa, Ndab, Ndc, Ndd	/*時間詞*/
Neu	*Neu*	/*數詞定詞*/.
Nes	*Nes*	/*特指定詞*/
Nep	*Nep*	/*指代定詞*/
Neqa	*Neqa*	/*數量定詞*/
Neqb	*Neqb*	/*後置數量定詞*/
Nf	Nfa, Nfb, Nfc, Nfd, Nfe, Nfg, Nfh, Nfi	/*量詞*/
Ng	Ng	/*後置詞*/
Nh	Nhaa, Nhab, Nhac, Nhb, Nhc	/*代名詞*/
I	I	/*感嘆詞*/
P	P*	/*介詞*/
T	Ta, Tb, Tc, Td	/*語助詞*/
VA	VA11,12,13,VA3,VA4	/*動作不及物動詞*/
VAC	VA2	/*動作使動動詞*/
VB	VB11,12,VB2	/*動作類及物動詞*/
VC	VC2, VC31,32,33	/*動作及物動詞*/
VCL	VC1	/*動作接地方賓語動詞*/
VD	VD1, VD2	/*雙賓動詞*/
VE	VE11, VE12, VE2	/*動作句賓動詞*/
VF	VF1, VF2	/*動作謂賓動詞*/
VG	VG1, VG2	/*分類動詞*/
VH	VH11,12,13,14,15,17,VH21	/*狀態不及物動詞*/
VHC	VH16, VH22	/*狀態使動動詞/
VI	VI1,2,3	/*狀態類及物動詞*/
VJ	VJ1,2,3	/*狀態及物動詞*/
VK	VK1,2	/*狀態句賓動詞*/
VL	VL1,2,3,4	/*狀態謂賓動詞*/
V_2	V_2	/*有*/
DE	/*的，之，得，地*/	
SHI	/*是*/	
FW	/*外文標記*/	

[15] Italics Tag, which has not defined in report #93-05, namely afterward increased.

Appendix 2. System Usability Scale Developed by Digital Equipments Co Ltd



System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	□ 1	□ 2	□ 3	□ 4	□ 5
2. I found the system unnecessarily complex	□ 1	□ 2	□ 3	□ 4	□ 5
3. I thought the system was easy to use	□ 1	□ 2	□ 3	□ 4	□ 5
4. I think that I would need the support of a technical person to be able to use this system	□ 1	□ 2	□ 3	□ 4	□ 5
5. I found the various functions in this system were well integrated	□ 1	□ 2	□ 3	□ 4	□ 5
6. I thought there was too much inconsistency in this system	□ 1	□ 2	□ 3	□ 4	□ 5
7. I would imagine that most people would learn to use this system very quickly	□ 1	□ 2	□ 3	□ 4	□ 5
8. I found the system very cumbersome to use	□ 1	□ 2	□ 3	□ 4	□ 5
9. I felt very confident using the system	□ 1	□ 2	□ 3	□ 4	□ 5
10. I needed to learn a lot of things before I could get going with this system	□ 1	□ 2	□ 3	□ 4	□ 5