

國立交通大學

資訊科學系

碩士論文

一個針對多語系網頁內容過濾的
快速精確之代理伺服器



A Fast Accurate Proxy for
Multi-Language Text Webpage Classification

研究生：黃福祥

指導教授：林盈達 教授

中華民國九十三年六月


一個針對多語系網頁 內容過濾的快速精確之代理伺服器

學生：黃福祥

指導教授：林盈達

國立交通大學資訊科學系

摘要



即時性的內容分析具有低維護成本及低空間需求性的特色，因此對網頁內容過濾來說是一種非常重要的技巧，但其同時也有準確度較低及處理時間過長的問題。由於多語系網頁的影響，相對也影響了準確度，因此我們嘗試以 N-gram 的演算法訓練樣本並找出關鍵字加入到內容過濾器中，評估以加入關鍵字的方式影響準確度的程度。此外，我們提出及早決策的演算法，此演算法包含兩部份，分別稱為及早阻擋和及早通過。前者在分類過程中一旦有足夠條件證明標的網頁屬於禁止類別便予以阻擋。反之，後者在發現標的網頁應屬於正常類別時，就會做出及早通過的決定。實驗結果顯示，在使用 Pentium III 1GHZ CPU 及 NetBSD 1.6 的作業系統環境下，我們提出的方式較原始的方式在傳輸效能上提升六倍，而在傳輸延遲上改善了三倍以上。同時在阻擋率從原來 70% 提升到 99%。

關鍵字：內容過濾，文件分類，N-gram，及早阻擋，及早通過

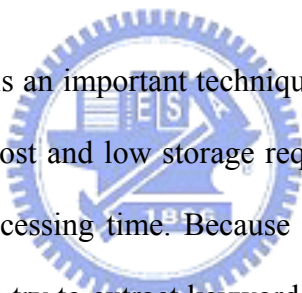
A FAST ACCURATE PROXY FOR MULTI-LANGUAGE TEXT WEBPAGE CLASSIFICATION

Student: Fu-Hsiang Huang

Advisor: Dr. Ying-Dar Lin

**Department of Computer and Information Science
National Chiao-Tung University**

Abstract



Real-time content analysis is an important technique in Web content filtering and has two advantages: low maintenance cost and low storage requirement. However, it may also suffer lower accuracy and longer processing time. Because Web pages in different languages can complicate content analysis, we try to extract keywords from training samples by the N-gram algorithm and evaluate the accuracy. To shorten the processing time, we propose the early decision algorithm that has two parts: early blocking and early bypassing. The former algorithm allows making the blocking decision as early as we have enough confidence that the Web page should belong to a forbidden category, while the latter helps to make the bypassing decision as soon as the Web page is considered a normal one. Experiments performed on NetBSD 1.6 with Pentium III 1GHZ CPU show our algorithm can improve the throughput about six times higher than the original and reduce the latency by two thirds. Furthermore, the blocking ratio is raised from 70% to 99%.

Keywords: content filtering, text classification, N-gram, early blocking, early bypassing

CONTENTS

CHAPTER 1. INTRODUCTION.....	1
CHAPTER 2. RELATED WORKS	4
CHAPTER 3. METHODOLOGY	7
3.1 Improving the accuracy	7
3.2 The language issue.....	7
3.3 Accelerating the filtering	10
CHAPTER 4. IMPLEMENTATION.....	12
4.1 Architecture of the DansGuardian	12
4.2 Possible problems and improvement in DG.....	13
4.3 Implementation Details	16
CHAPTER 5. BENCHMARKING	17
5.1 Benchmarking methodology	17
5.2 External benchmarking results	17
5.3 Internal benchmarking results	20
CHAPTER 6. CONCLUSIONS AND FUTURE WORKS	22
REFERENCES	24

List of Figures

Fig 1. The variations of the N-gram algorithm.....	6
Fig 2. The N-gram pseudo code	8
Fig 3. The pseudo code of early decision	11
Fig 4. Architecture of the DansGuardian	13
Fig 5. Throughput and request rate of each filtering method in DG	14
Fig 6. The ratio of latency in response processing	15
Fig 7. The accuracy when scanning $n\%$ of the Web pages.....	15
Fig 8. The implementation process	16
Fig 9. The number of requests (w/ Early Decision vs. w/o Early Decision).....	19
Fig 10. The throughput (w/ Early Decision vs. w/o Early Decision).....	19
Fig 11. The throughput when scanning $n\%$ of the Web pages	20
Fig 12. The latency of different sizes of Websites (w/ Early Blocking vs. w/o Early Blocking)	20
Fig 13. The latency of different sizes of Websites (w/ Early Bypassing vs. w/o Early Bypassing)	21
Fig 14. The performance w/ and w/o cache (w/ Early Decision vs. w/o Early Decision)....	22

List of Tables

Table 1. Comparison of major products in the market	4
Table 2. The blocked ratio of three functions in the DG	18
Table 3. The false positive ratio.....	18



Chapter 1

INTRODUCTION

The amount of unwanted content containing pornography and violence increases as the Internet grows larger. Without control, the content can be easily accessed by adolescents and children. Workers during office hours can also spend their time in accessing unrelated content in the Internet. Many Web filtering products have come up to block the inappropriate content.

The approaches to Web filtering can be classified into four categories: (1) Platform for Internet Content Selection (PICS) [1], (2) URL-based [2], (3) keyword-based and (4) content analysis. PICS provides a common format for labels, so that any PICS-compliant selection software can process any PICS-compliant label. PICS labels describe content on one or more dimensions. It is the selection software, not the labels themselves, that determine whether access will be permitted or prohibited. Most Web filtering products in the market adopt the URL-based approach as the main approach [3, 4] where a black list [5] of banned URLs is maintained. Some products also include a white list of allowed URLs, usually defined by the user. The keyword-based approach, which counts the number of matching keywords in a Web page for classification, helps to reduce the number of false negatives. The keywords are usually selected manually and may introduce false positives, i.e., the permissible pages being banned. For example, a Web page discussing sex education may be banned because of the keyword 'sex'. Content analysis does more than keyword matching by training some sample pages, extracting features from them, such as keywords, number of images, the type and number of links, and so forth. In this work, we evaluate these design alternatives in terms of the following issues and propose a combination of them to keep both accuracy and speed.

Improving the accuracy

Without looking deeper into the Web content and banning the request before it reaches the

Web site, the URL-based approach is usually much faster and bandwidth-saving. However, given that there are ten-million new Web pages each day in the Internet, it is a huge effort, if possible, to keep the URL database up-to-date. With the supplement of Web content analysis, the maintenance effort of the URL database could be relieved and the number of false negatives, i.e., the forbidden pages being allowed, can be reduced. Forbidden Web pages, however, may be classified into a banned category even if its URL is not in the black list. Content analysis is usually more time-consuming because of its complexity. The design tradeoffs are to be discussed in this work.

The language issue

Web pages can exist in many languages, which have different characteristics. An ideographic language, such as Chinese, does not have spaces that separate words. The counterpart of an English key phrase can become a keyword in Chinese, and vice versa. It is desirable to design a generic classifier that automatically classifies Web content in a uniform approach. We use the N-gram algorithm [6, 7] to train the classifier with sample Web pages and reduce the keywords by three rules: frequency, breadth and length. We explain these three rules in Section 3. The N-gram algorithm uses statistics to compute the keyword frequency and extracts appropriate keywords and key phrases from Web pages. We will discuss the possibilities and difficulties in designing such a generic classifier.

Accelerating the filtering

Most text classification methods use keywords or key-phrases extracted from training samples as signatures to classify the target Web pages. After the whole page has been scanned, it will be classified into the category with the highest probability. This results in long response time and brings heavy load to the system. For example, some of content analysis filter will check the content of hyperlinks and do complex algorithmic analysis (see Chapter 2) and these methods are not suitable for real-time analysis. Therefore, we propose early decision algorithm, coming from the observation that the classification can be completed before

scanning the entire document. The method includes two parts: early blocking and early bypassing. Early blocking allows us to block the Web contents as early as we have enough confidence that the Web page should belong to some forbidden category. Conversely, early bypassing determines that the contents belong to an allowed category as early as possible and should be considered normal in a content filter and hence bypassed. The ability of fast bypassing such normal traffic is particularly important because the majority of traffic would be normal.

In this work, we address the above three issues with the following solutions: Bayesian classification, N-gram algorithm, and early decision. We implement and integrate these methods by modifying DansGuardian, an open source Web filter and benchmark its external and internal performance. In external benchmarking, we aim at the accuracy and throughput. In internal benchmarking, we aim at the latency.

The rest of this work is organized as follows. Section 2 reviews the related works. Section 3 describes the problem statement and our solutions. Section 4 and Section 5 present the implementation and benchmarking on the open-source content filter, DansGuardian, respectively. Finally, the study is concluded in Section 6.

Chapter 2

RELATED WORKS

Here we summarize the filtering methods of some major Web filtering products in Table 1. All of them use the URL database as the main approach and most of them also provide keyword-based approach for users to customize the keywords. Only few of them adopt the content analysis method. The reasons could be the performance and accuracy issues. First, the URL database is generally faster than content analysis and saves more bandwidth. Second, content analysis could suffer the risk of false positives, a condition that is not generally preferable. Image analysis is even seldom used in major Web filtering products because the analysis is more time-consuming.

Table 1. Comparison of major products in the market

Products	Features	Filtering methods				Filter Categories
		URL database	Keyword based	Content analysis	Image recognition	
Child Safe		●				
Cyber Patrol		●	●	●		13
Content Protect		●	●	●		22
Cyber Sentinel		●	●			
Cyber Sitter		●	●			30
Cyber Snoop		●	●			
Filter Park		●	●			15
McAfee-Patrol Controls		●	●	●		41
Net Nanny		●	●	●		
Norton-Patrol Controls		●				31
WebSense		●	●			80
SurfControl		●	●			55

Unlike the cases in commercial products, content analysis has been studied intensively over years. Most concentrate on text classification because text is still a major part in Web pages. The typical applications are search engines, metadata generation, hierarchical

categorization of Web pages, and Web filtering [8]. There are two major directions in research: feature selection and content classification. The former identifies representative features that can effectively characterize the category that a Web page belongs to. The typical features are document frequency (DF), information gain (IG), mutual information (MI), a χ^2 -test (CHI) and term strength (TS) [9]. In this work, we focus on the keywords as the features because Web pages hold large part of text (keywords) and keywords are more suitable for the above features (DF, IG, etc). The latter identifies the similarity between the selected features of a Web page and those of the training samples to classify the Web page to a certain category. Naïve Bayesian (NB) [10-12], Support Vector Machines (SVM) [13], K-Nearest Neighbor (KNN) [12], Adaptive Resonance Associative Map (ARAM) [14], Decision Tree (DTree) [12], Boosting [15], regression models [16], inductive rule learning [17-19], on-line learning [18] and Neural Network (NNet) [20] are among the classification algorithms.

The Naïve Bayesian classifiers are widely used because of their simplicity and computational efficiency. The NB uses relative frequencies of words in a document to compute the probability that the Web page belongs to a category and assign the page to the category with the highest category.

The SVM uses decision surfaces to divide data points into classes. In its simplest form, training documents are represented as vectors and the algorithm determines hyper-planes that separate different classes of training documents. Test documents are classified according to their positions with respect to the hyper-planes.

The KNN selects clusters of k most similar documents represented by vectors of words from the training set and finds the nearest cluster to which the document to be classified belongs.

Content analysis can become more difficult because of characteristics such as double bytes, multi-character keywords or phrases and no spacing between keywords in some languages, particularly oriental languages. Some extraction algorithms use syntactical

functions and morphological features that help to determine a part of speech and extract the keywords [21, 22], like verb, proper noun and so on. But these methods require a large database of keywords (dictionary) beforehand to extract suitable keywords during training.

The N-gram [6, 23] extracting algorithm is the most popular and suitable for both oriental and western languages. Fig 1 is shown that the N-gram can be used with a dictionary or without. There are two major approaches by N-gram with a dictionary: word segmentation system and tagging, which use syntactical functions and morphological features. It is also possible to use statistics and manual extraction without dictionaries. The N-gram extraction algorithm aims at bi-gram frequency statistics, tri-gram frequency statistics and so on in the training phase. Consequently, it becomes clear what keywords often appear in certain categories. We can classify the Web pages by computing scores that evaluates the probability of a category that a Web page belongs to according to keywords or key phrases.

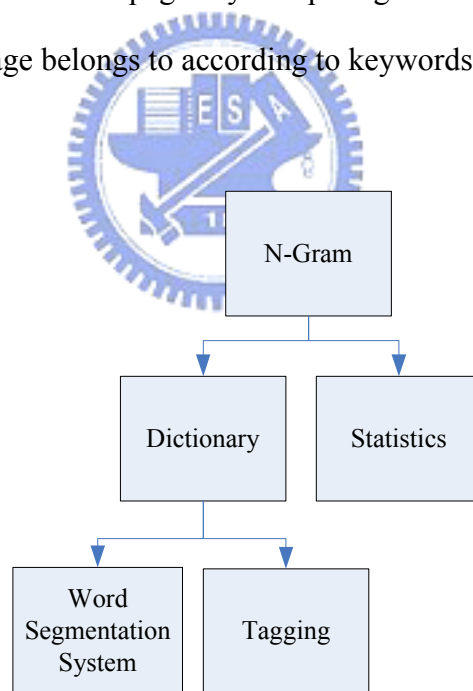


Fig 1. The variations of the N-gram algorithm

Chapter 3

METHODOLOGY

In Chapter 1, we indicate three issues: improving the accuracy, the language issue and accelerating the filtering. And in this chapter, we describe how to evaluate accuracy and improve it by considering the language issue which is resolved by the N-gram algorithm. Finally, we propose the *early decision* algorithm to accelerate the filtering.

3.1 Improving the accuracy

In this work, we evaluate the accuracy of the URL-based method, keyword-based method, and content analysis. We test each method individually and their combinations, totally up to seven testing items with each method either used or not used. We then add Chinese keywords and repeat the experiments. The experiments will be explained in Chapter 4. They are designed to answer the following questions:

- (1) Which method (or the combination) is the most effective in terms of the accuracy?
- (2) How does each method (or the combination) affect the system throughput?
- (3) Does taking the language issue into account in content analysis increase the accuracy significantly?

3.2 The language issue

As we reviewed in Chapter 2, the N-gram algorithm has several variations for text classification of more than one language. Word segmentation system or tagging requires a dictionary, which takes great efforts to build. Some terms, particularly proper nouns, are not easy to exhaust. Here we use N-gram with statistics instead. The algorithm is explained by the

following example. Given a Chinese keyword “證券交易所”，we use N-gram with N from 1 to 5 and get the following terms:

One-gram: “證”，“券”，“交”，“易”，“所”


Bi-gram: “證券”，“券交”，“交易”，“易所”

Tri-gram: “證券交”，“券交易”，“交易所”

4-gram: “證券交易”，“券交易所”

5-gram: “證券交易所”

Fig 2 presents the pseudo code of our N-gram approach. We use a keyword array to store the characters of N-gram and count the keyword frequency if the keyword exists in the keyword table. Otherwise, we add the keyword to the keyword table and then start to count the keyword frequency.



Algorithm:

```

Count:=0;
While(not EOF) {
  If (Count = 0)
    Clear the keyword array;
  If ( (Get a character) <> a blank character) {
    Add to keyword array;
  }
  Else
    Count:=Count + 1;
  If (Count <> N)
    Continue;
  If (keyword array is not in the keyword table) {
    Add the keyword into the keyword table;
  }
  Count the keyword frequency;
  Count:=0;
}

```

// count the N of N-gram

Fig 2. The N-gram pseudo code

There are three issues to be discussed: (1) how to determine the appropriate value of N, (2) how to assign the scores to keywords, and (3) how to delete stop words (extremely common

words, say “the”), which are useless in classification.

The value of N can be determined by the keyword frequency in the training samples. For each N -gram term, we choose the largest N such that the frequency of N -gram term is at least five.

We use three criteria for keyword extraction and score assignment: (1) frequency, (2) breadth, and (3) length. First, the keyword frequency should be greater than a certain threshold. Second, according to breadth to determine which keyword will be selected. It means that the breadth is not enough if the keywords appear in a small number of Web pages and the Web pages belong to the same category. For the length criterion, if s is a substring of t and the appearance of s always means the appearance of t , we prefer use t as the keyword. For example, the keyword “交通大學” is preferable to “交通大”. We use keyword frequency as the basis to determine the scores. The procedure is as follows:

Given:

$F(S)$: frequency (score) of string S

$|S|$: length of string S

$B(S)$: Breadth of string S

1. Gather statistics:

if $F(S) \geq 5$ and $B(S) \geq 0.1$, then pick the string S .

2. Delete the sub-string(S') of the same frequency:

if $F(S)=F(S')$, then delete S' .

Ex. $F(\text{“華民國”}) = 10$, $F(\text{“中華民國”}) = 10$, thus delete “華民國”.

3. Modify the frequency (score) of string:

Assume we get three strings, A , B and C , and $A=B+C$.

Because $F(A) < F(B)$ and $F(A) < F(C)$, $|A| > |B|$ and $|A| > |C|$, $F(A)=F(A)+F(B)+F(C)$.

Ex. $F(\text{“證券”}) = 10$, $F(\text{“交易”}) = 20$, $F(\text{“證券交易”}) = 10$

$\rightarrow F(\text{“證券交易”}) = F(\text{“證券交易”}) + F(\text{“證券”}) + F(\text{“交易”}) = 10 + 10 + 20 = 40$

Because stop words tend to appear in both forbidden and normal Web pages, we use samples of normal Web pages to extract the keywords with the above procedure, and delete stop keywords. As a result, only meaningful keywords remain.

3.3 Accelerating the filtering

The existing approaches to content filtering scan the entire Web pages. Although it is not demanded to scan the entire page with existing methods, how much should be scanned while keeping the accuracy is not addressed at all. Given the percentage $n\%$ of Web pages that has been scanned and the score m that has been accumulated, we can derive the probability that the page belongs to a category with the Bayesian formula as

$$P(C | D(n, m)) = \frac{P(D(n, m) | C)P(C)}{P(D(n, m) | C)P(C) + P(D(n, m) | C')P(C')} \quad (1)$$

where

1. $D(n, m)$: have read $n\%$ of text and have the accumulated score deviation m so far,
2. $P(C)$: the estimated probability that the text should belong to category C ,
3. $P(C')$: the estimated probability that the text should not belong to category C , and
4. $P(D(n, m) | C)$: the estimated probability of $D(n, m)$ given that the Web page belongs to category C . The estimate of $P(D(n, m) | C)$ is

$$P(D(n, m) | C) = \frac{\#(C \text{ in which } D(n, m))}{\#C} \quad (2)$$


The computation of $P(D(n, m) | C')$ is defined similarly.

In the training phase, we build a table of $P(D(n, m) | C)$ for using (2) from the training examples. The values of $P(C)$ and $P(C')$ can be estimated beforehand or dynamically adjusted in a running environment by recording and analyzing actual Web pages. Early decision,

including *early blocking* and *early bypassing*, comes from the observation that the filtering decision can be made before scanning the whole Web page. The early blocking algorithm allows us to block the Web page as early as we have enough confidence that the Web page should belong to some forbidden category. Conversely, the early bypassing algorithm determines that the Web page belongs to no banned categories as early as possible and should be bypassed. Early bypassing is particularly important because the majority of traffic would be normal. We define two thresholds for early bypassing and early blocking. With the early decision, we can either block or bypass a Web page before scanning the entire page.

Fig. 3 shows the pseudo code of the early decision algorithm. In the following pseudo code of the early decision algorithm, we compute the values of PCD and PDC with (1) and (2). The length of the Web page can be learned from the HTTP response header. We scan at least 30% of the Web page because scanning only a little text can result in more errors in classification.

Early Decision:



```

Earlybypass := False; // initialize
Earlyblock :=False;
I:=0;
Count:=0.3; // scanning at least 30% of document
Do {
  Read next words;
  I is the percentage of text that has been read;
  If (I > Count) {
    PDC:=Get the  $P(D(n, m)|C)$  of Current scanning position;
     $PCD := (PDC * PC) / ((PDC * P_c) + PD\_C * P\_C)$ ;
    //  $P(C|D(n, m)) = [P(D(n, m)|C')P(C)] / [P(D(n, m)|C)P(C) + P(D(n, m)|C')P(C')]$ 
    If (PCD < early bypassing threshold) {
      Earlybypass:=True;
      Break;
    }
    If (PCD > early blocking threshold) {
      Earlyblock:=True;
      Break;
    }
  }
}
} While(I<len);

```

Fig 3. The pseudo code of early decision

Chapter 4

Implementation

In this chapter, we pick an open source Web content filtering package, DansGuardian, for the implementation and experiments because it is still updated frequently and dedicated to Web content filtering. We introduce its architecture in Section 4.1 and indicate problems in DG by the experiments in Section 4.2. And we introduce the implementation details in Section 4.3.

4.1 Architecture of the DansGuardian

Fig. 4 shows that the operation of DansGuardian (DG). The client sends the request to Web server through DG. DG checks the URL and responds to the client with a denying message if the URL falls in the black lists; otherwise it sends the request to the Web server through the Squid proxy, which receives the Web pages from the Web server and returns them to DG. DG analyzes the content and finally returns it to the client if the content is allowable.

DG keeps a list of 801,626 pornographic domains and 150,388 pornographic URLs to date. The domains are a crowd of forbidden domain lists and the URLs are also a crowd of forbidden URL lists. There are 3841 keywords and key phrases of pornography for content analysis.

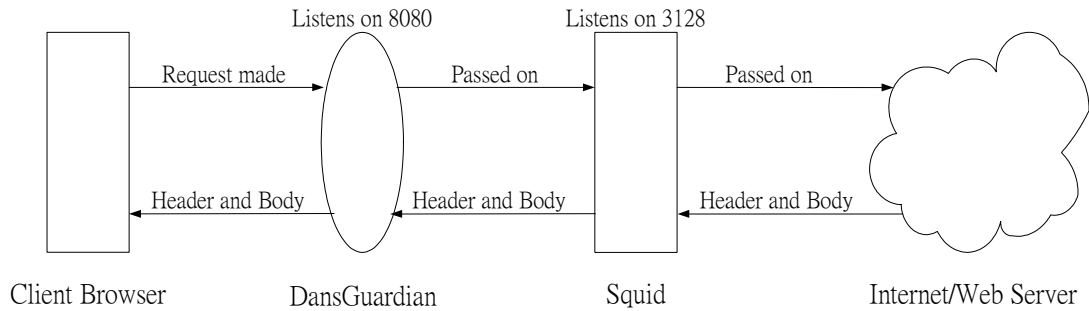


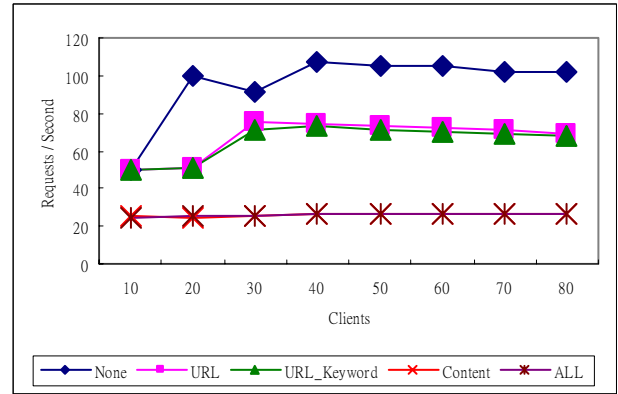
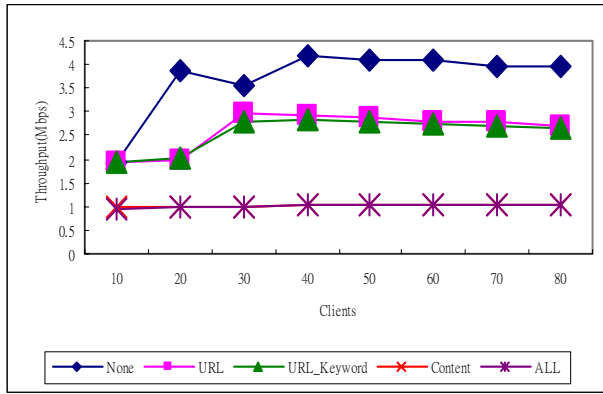
Fig 4. Architecture of the DansGuardian

4.2 Possible problems and improvement in DG

We do the internal and external benchmarks on DG to identify the bottleneck of speed and estimate the accuracy. We can divide the problem into three parts to discuss:

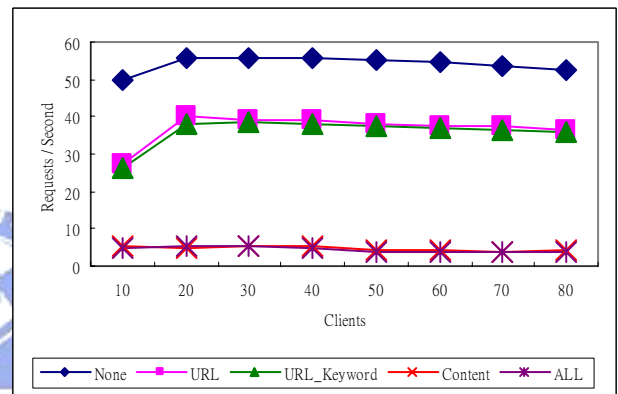
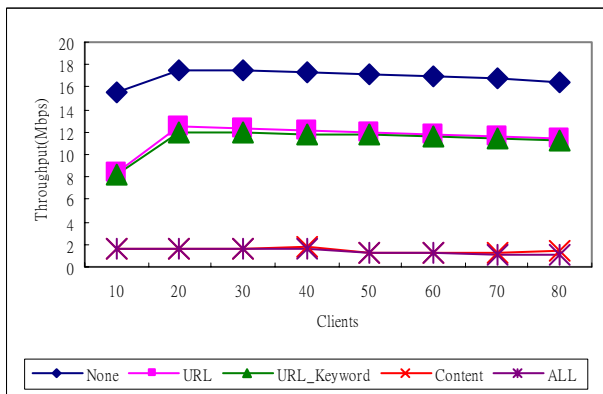
(1) Low throughput

We simulate 10~80 clients to access Web pages of 5KB and 40 KB (a typical Web page size is from 4KB to 8KB [24] and most of normal Web pages do not exceed 40KB in our observation) on the Web server. In Fig. 5, “None” means turning off all checking items. “URL”, “URL Keyword” and “Content” means turning on URL database, URL keyword and content keyword checking items, respectively. “All” means turning on the above three checking items. Fig. 5(a) shows that even when turning off all functions in DG, the throughput is only 4.2 Mbps and Fig. 5(c) shows that the throughput is 17.5 Mbps for 40 KB Web pages. The throughput is extremely awful because DG has a poor control flow in the string matching module and process fork costs a lot of time and causes system overhead. The poor control flow is that DG collects all matched keywords after scanning entire Web page and finally computes the score of Web page. Therefore, we focus on the modification of the control flow in executing the string matching algorithm and implement *early decision* algorithm to reduce the system overhead and improve the throughput.



(a) Throughput with 5 KB of response size

(b) Request rate with 5 KB of response size



(c) Throughput with 40 KB of response size

(d) Request rate with 40 KB of response size

Fig. 5. Throughput and request rate of each filtering method in DG

(2) Bottleneck

The latency of each checking item of responses is illustrated in Fig. 6. DG applies filters for banned MIME-types (ex. video/mpeg, application/zip, etc.) and file extensions (ex. *.zip, *.dll, *.mp3, etc.) and content analysis in response processing. Content analysis occupies 99.72% of the total latency. It is thus clear that content analysis is the bottleneck. It must scan the entire Web page at least once and then determines the category by accumulating scores. We propose the *early decision* algorithm to reduce the processed time after we confirm this bottleneck.

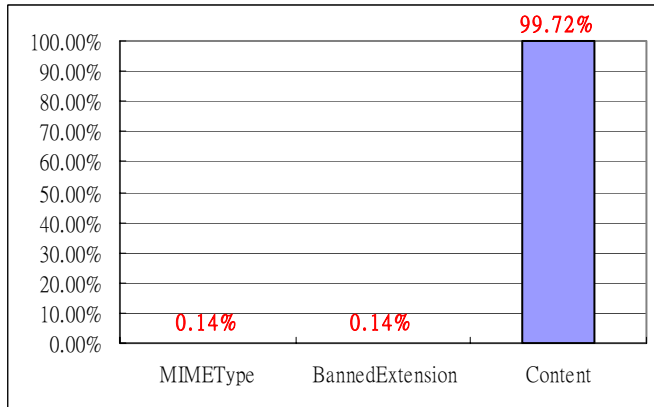


Fig. 6. The ratio of latency in response processing

(3) Early Decision

We choose pornography as the category to be banned. We collect 200 samples for the testing. Fig. 7 shows by scanning only 30% of the Web pages, the accuracy can be close to that by scanning the entire Web pages. Only 2% more errors result from the early decision. The observation shows that we can save 70% of scanning time while keeping the accuracy with our early decision algorithm.

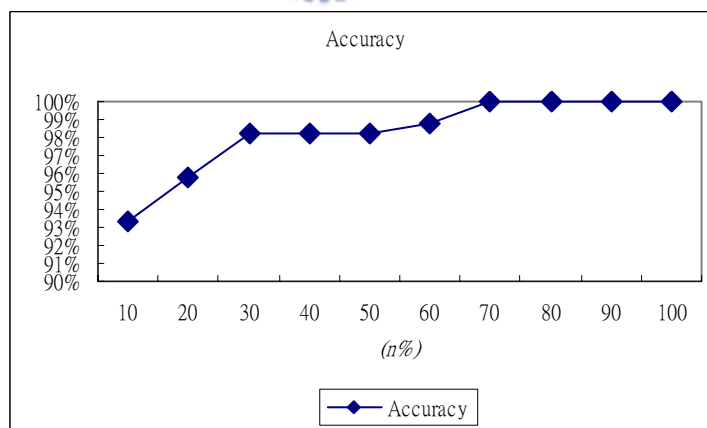


Fig. 7. The accuracy when scanning $n\%$ of the Web pages

4.3 Implementation Details

Fig. 7 presents our implementation in three stages: (1) collecting Web pages, (2) extracting proper keywords with the N-gram algorithm and assigning scores, and (3) implementing the early decision algorithm into DG.

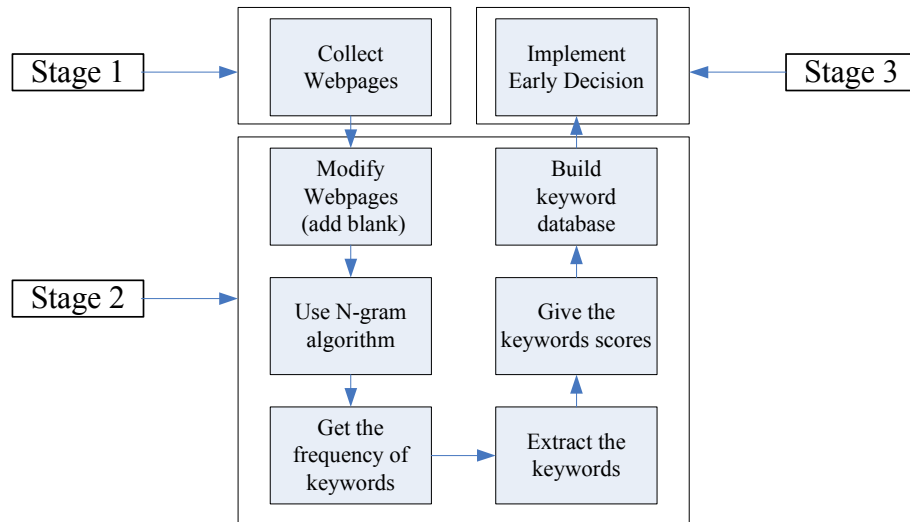


Fig. 8. The implementation process

In Stage 1, we collect 250 western and 250 Chinese pornographic Web pages randomly using the Google searching engine, delete the tags of these Web pages and keep the text part of the content. In Stage 2, we extract keywords of bi-gram, tri-gram and so on from 100 Chinese pornographic Web pages and assign scores to them with the algorithms in Section 3.2. In the final stage, we implement the early decision algorithm into DG. According to the observation in Section 4.2, we choose to scan at least 30% of the content.

Chapter 5

BENCHMARKING

5.1 Benchmarking methodology

In the accuracy benchmarking, we collect 250 western and 250 Chinese pornographic Web pages for testing. We either turn on or turn off URL, URL keyword and content analysis, totally seven possible cases, to observe the blocking ratio of each combination.

In the performance benchmarking, we use WebBench 5.0 and seven Pentium III 1GHz computers as the clients. Each simulates ten clients sending requests through the content filter to a Web server with a Pentium4 2.8GHz. The size of the Web size is set to 40KB.

In the internal benchmarking, we measure the average latency of filtering 100 Web pages with and without early decision. The testing samples for early blocking are real pornographic Web pages of 1KB, 6KB, 18KB, 29KB and those for early bypassing are Web pages of Google (4.12KB), NCTU (20.1KB) and PCHOME (35.6KB).

5.2 External benchmarking results

In Table 2, the blocking ratios of each filtering combination after adding Chinese keywords are written in bold characters. From the table, we observe that content analysis alone with Chinese keywords can reach about 98% of accuracy, which means the URL-based approach can be replaced in theory if the accuracy is the only concern. Both URL-based approaches and content keywords are effective methods for Web filtering. URL keywords are of little use because many pornographic Web sites do not have the keywords in their URLs.

Table 2. The blocked ratio of three functions in the DG

URLs	URL Keywords	Content Keywords	Chinese Pornographic Websites (250)		Western Pornographic Websites (250)	
			Blocked Pages	Blocked Ratio	Blocked Pages	Blocked Ratio
⊙			159	63.6%	241	96.4%
	⊙		4	1.6%	41	16.4%
		⊙	174→ 243	69.6%→ 97.2%	226	90.4%
⊙	⊙		159	63.6%	241	96.4%
⊙		⊙	218→ 247	87.2%→ 98.8%	245	98.0%
	⊙	⊙	175→ 243	70.0%→ 97.2%	227	90.8%
⊙	⊙	⊙	218→ 247	87.2%→ 98.8%	245	98.0%

Furthermore, we collect 60 Chinese Web pages and 60 western Web pages, where each language has 15 pages of finance, 15 pages of shopping, 15 pages of games and 15 pages of news. We want to know if there is any false positive. Table 3 shows that we turn on pornographic check in the Web content filter and get no false positives in the four categories. It looks that such text classification can work very well given the high blocking ratio and the very low false positives. But in practice, the Web pages may contain only objects such as Flash or Java applet. Web filtering cannot completely rely on content analysis with only text classification.

Table 3. The false positive ratio

	Chinese Web sites (60)	Western Web sites (60)	False positive Ratio
Finance (15)	0/15	0/15	0%
Shopping (15)	0/15	0/15	0%
Games (15)	0/15	0/15	0%
News (15)	0/15	0/15	0%

Fig. 9 and Fig. 10 indicate that the request rate and throughput can be improved about six times with early decision. And we also implement a new program (Webfd) without using process fork and gets better throughput.

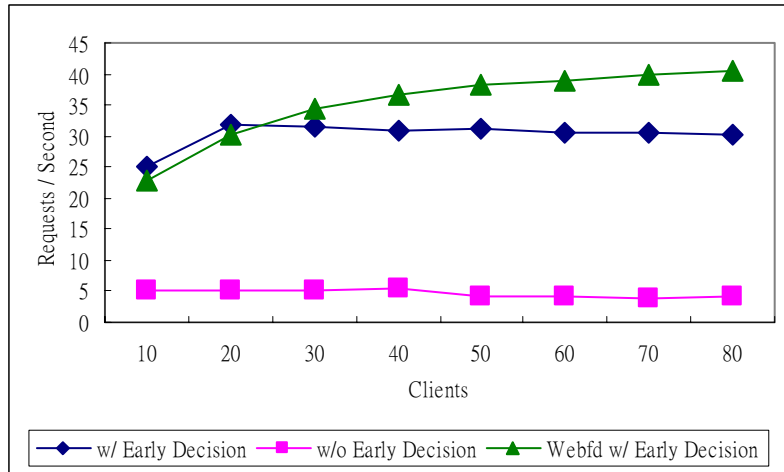


Fig. 9. The number of requests (w/ Early Decision vs. w/o Early Decision)

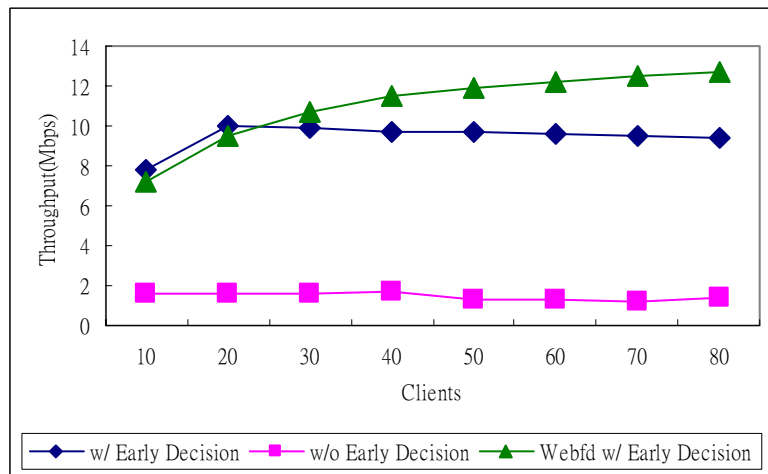


Fig. 10. The throughput (w/ Early Decision vs. w/o Early Decision)

Fig. 11 shows that we improve the throughput by only modifying the classification algorithm of DG and the improvement of classification algorithm is over three times.

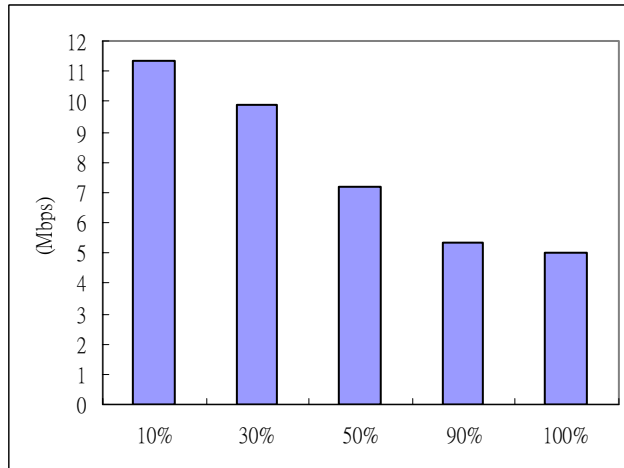


Fig. 11. The throughput when scanning $n\%$ of the Web pages

5.3 Internal benchmarking results

Fig. 11 shows that the latency with early blocking is about four times shorter than that without early blocking.

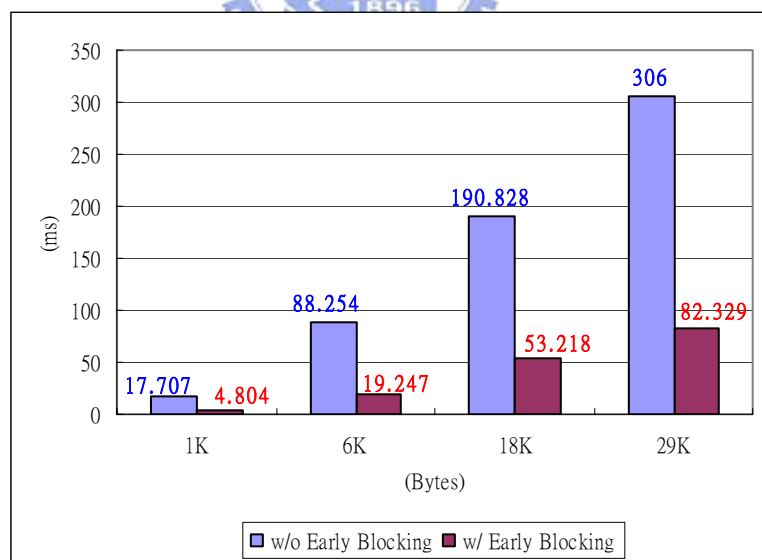


Fig. 12. The latency of different sizes of Websites (w/ Early Blocking vs. w/o Early Blocking)

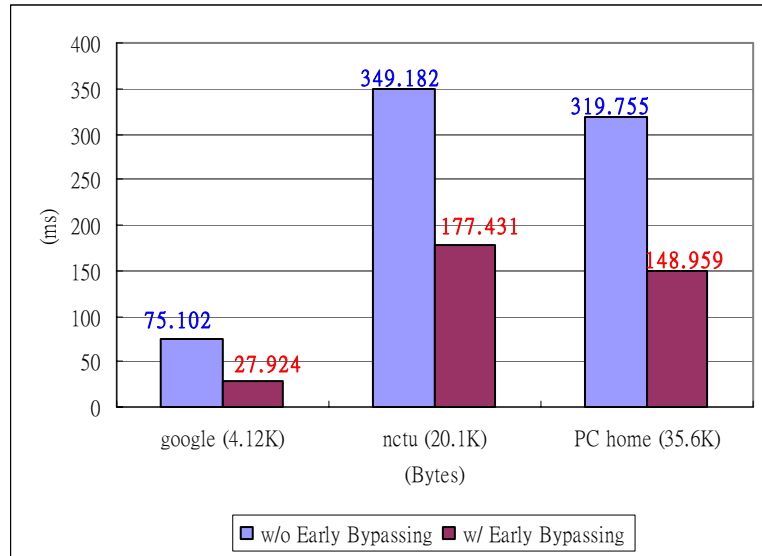
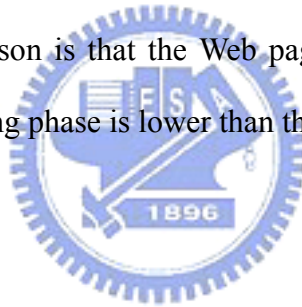


Fig. 13. The latency of different sizes of Websites (w/ Early Bypassing vs. w/o Early Bypassing)

The observation from Fig. 12 is that the latency difference is uncertain with or without early bypassing. The main reason is that the Web page should be passed by early decision when the score of some scanning phase is lower than the early bypassing threshold.



Chapter 6

CONCLUSIONS AND FUTURE WORKS

The observation from the results of benchmarking is that we could replace the database of URLs of the URL-based approach by content analysis while improving the accuracy.

The throughput can be further improved with caching. By caching we mean the classification results are stored in a cache. If the same request goes through the content filter again, the filter neither checks the URL nor analyzes the response content, but make the blocking decision from the cache. Fig. 13 shows that the throughput with and without caching.

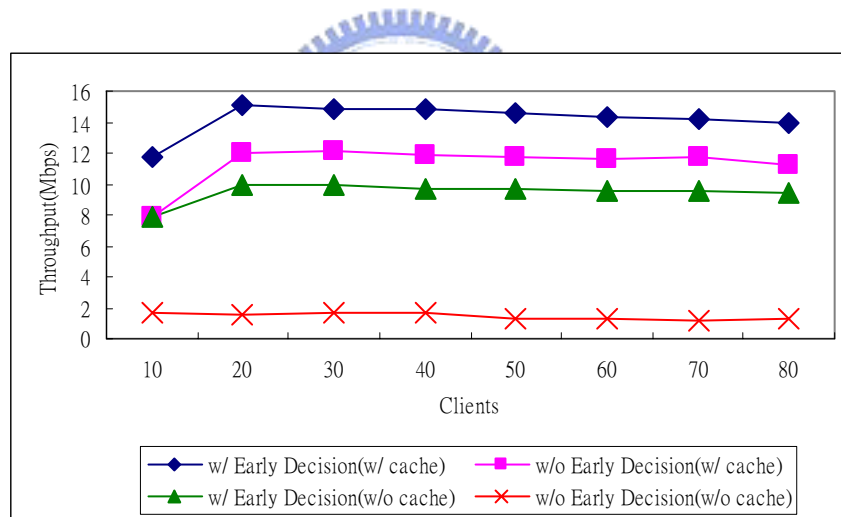


Fig. 14. The performance w/ and w/o cache (w/ Early Decision vs. w/o Early Decision)

By adjusting the early blocking and early bypassing thresholds, we can increase the accuracy at the cost of throughput degradation or promote the throughput at the cost of lower accuracy. It depends which is more important, accuracy or throughput, in a practical environment.

In this work, we discuss the accuracy and throughput in content analysis. By analyzing the

Web content with extra keywords from the N-gram algorithm, the accuracy is increased from 70% to 99%. Content analysis has the potential to replace the URL-based method in terms of accuracy. Furthermore, we also find that keyword filtering in the URLs, which is implemented in many commercial products, is of little use in general.

We propose the early decision algorithm to improve the filtering speed. The performance evaluation in Chapter 5 shows that the latency of forbidden Web pages and normal Web pages with early decision, compared with the a non-early decision one, gains the shorter latency about four times for forbidden Web pages, and about three times for normal Web pages and the throughput improvement of about six times.

However, some issues need further study: increasing the number of categories to be classified, picture recognition, white lists or black lists caching and ASIC accelerating. Furthermore, it will be better in Web content filtering if we can use a hybrid method with URL-based. It is quite obvious in performance improvement with cache of white lists or black lists. If the accessed Web page was not in database of URLs, it will be stored into the cache after content analysis determining that the Web page should be a forbidden or passed Web page. The request will not need to do URL database matching or content analysis next time and only need to determine to block or pass the request by looking up the white list or black list.

Web content filtering is a promising technology and has already been widely used in the education systems and commercial organizations. With a fast automated Web text classification algorithm, it is possible to build a real-time content analysis filter that has high throughput and low maintenance overhead.

REFERENCES

- [1] Paul Resnick and Jim Miller, PICS: Internet access controls without censorship. *Communications of the ACM*, 39(10):87-93, 1996.
- [2] Harold Kester, Websense Web Catcher White Paper, <http://www.websense.com/products/resources/wp/>, 2001
- [3] Pui Y. Lee, Siu C. Hui, Alvis Cheuk M. Fong, "Neural Networks for Web Content Filtering," in *IEEE Intelligent Systems*, Sept.-Oct., 2002, pp. 48-57.
- [4] Internet Filter Reviews 2004, <http://www.internetfilterreview.com/?engine=adwords!883&keyword=%28internet+filter%29>.
- [5] DansGuardian. <http://dansguardian.org>.
- [6] Cavnar, William B. and John M. Trenkle, "N-Gram Based Text Categorization," in *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 11-13 April 1994, pp. 161-169.
- [7] Stochastic Language Models (N-Gram) Specification, W3C Working Draft 3. <http://www.w3.org/TR/n-gram-spec/>
- [8] F. Sebastiani, "Machine Learning in Automated Text Categorization," in *ACM Computing Surveys*, 34(1):1-47, 2002.
- [9] Yang, Y., Pedersen, J.O., "A Comparative Study on Feature Selection in Text Categorization," in *Proceedings of the 14th International Conference on Machine Learning ICML97*, 1997, pp. 412-420.
- [10] K. Tzeras and S. Hartman, "Automatic indexing based on Bayesian inference networks," in *Proceedings of the 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pp. 22-34, 1993.
- [11] A. McCallum and K. Nigam, "A comparison of event models for naïve bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

- [12] Tom Mitchell, *Machine Learning*, McGraw Hill, 1996
- [13] Thorsten Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *European Conference on Machine Learning (ECML)*, pages 137-142, Berlin, 1998. Springer.
- [14] A.-H. Tan. Adaptive Resonance Associative Map. *Neural Networks*, 8(3):437-446, 1995.
- [15] Schapire, R.E., Singer, Y., "Boostexter: a boosting-based system for text categorization," *Mach. Learn.* 39, 2/3, 135-168, 2000.
- [16] N. Fuhr, S. Hartmann, G. Lustig, M. Schwantner, and K. Tzeras, "Air/x – a rule-based multistage indexing system for large subject fields," in 606-623, editor, *Proceedings of RIAO'91*, 1991.
- [17] C. Apte, F. Damerau, and S. Weiss, "Towards language independent automated learning of text categorization models," in *Proceedings of the 17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, 1994.
- [18] William W. Cohen. And Yoram Singer, "Context-sensitive learning methods for text categorization," in *Proceedings of the 19th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, pp. 307-315, 1994.
- [19] I. Moulinier, G. Raskinis, and J. Ganascia, "Text categorization: a symbolic approach," in *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, 1996.
- [20] Erik D. Wiener, Jan O. Pedersen, and Andreas S. Weigend., "A neural network approach to topic spotting," in *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, 317-332, 1995.
- [21] Hsin-Hsi Chen and Jen-Chang Lee, "Identification and Classification of Proper Nouns in Chinese Texts," in *Proceedings of 16th International Conference on Computational Linguistics*, Aug. 1996.
- [22] 曾慧馨, 劉昭麟, 高照明, 陳克健, "A Hybrid Approach for Automatic Classification

of Chinese Unknown Verbs,” in *International Journal of Computational Linguistics & Chinese Language Processing*. Vol.7, no. 1, Feb. 2002.

[23] Fuchun Peng, Dale Schuurmans, “Combining Naïve Bayes and n-Gram Language Models for Text Classification,” in *The 25th European Conference on Information Retrieval Research (ECIR)*, Dec. 2003.

[24] Web Protocols and Practice, Balachander Krishnamurthy & Jennifer Rexford, pp. 380, 2001.

