PAPER

Accelerating Web Content Filtering by the Early Decision Algorithm

Po-Ching LIN[†a)], **Ming-Dao LIU**[†b)], ***Nonmembers***, **Ying-Dar LIN**[††c)], ***Member***, ***and*** **Yuan-Cheng LAI**[†††d)], ***Nonmember***

SUMMARY Real-time content analysis is typically a bottleneck in Web filtering. To accelerate the filtering process, this work presents a simple, but effective early decision algorithm that analyzes only part of the Web content. This algorithm can make the filtering decision, either to block or to pass the Web content, as soon as it is confident with a high probability that the content really belongs to a banned or an allowed category. Experiments show the algorithm needs to examine only around one-fourth of the Web content on average, while the accuracy remains fairly good: 89% for the banned content and 93% for the allowed content. This algorithm can complement other Web filtering approaches, such as URL blocking, to filter the Web content with high accuracy and efficiency. Text classification algorithms in other applications can also follow the principle of early decision to accelerate their applications.

key words: *Web filtering, text classification, World Wide Web, early decision*

1. Introduction

A huge amount of Web content is widely accessible nowadays. As inappropriate content such as pornography proliferates with the growth of World Wide Web, access control of such content is demanded in some situations. For example, an employer does not want the employees to watch stock information during working hours, or parents do not want their children to browse pornographic content. Web filtering products that enforce access control are therefore getting popular on the market. They can be deployed either on a host computer (e.g., in a family), or on the gateway for central management in a company or an Internet service provider.

Four major approaches are generally adopted in Web filtering nowadays: Platform for Internet Content Selection (PICS), URL-based, keyword-based and content analysis [1]. According to a recent review of up-to-date Internet filters, commercial products have widely adopted content analysis besides the URL-based approach [2]. Content analysis automatically classifies the Web content into a category first, and then makes the filtering decision, either to block or to pass the content, according to the management policy. The analysis generally complements the URL-based approach to relieve the effort of frequently updating the URL list and to reduce the number of false negatives due to an outdated URL database.

The efficiency of content analysis algorithms is essential due to their complexity. Slow analysis in Web filtering leads to long user response time and also degrades the throughput of Web filtering systems. We therefore focus on *text classification*, which remains an important and efficient approach to Web content analysis, despite the research on image content analysis for Web filtering [3], [4]. Moreover, image content analysis in Web filtering is mostly designed for pornography recognition, but not as effective for content in other banned categories.

Numerous text classification algorithms with high accuracy have been proposed. They are designed primarily for off-line applications, such as Web categorization for catalogs hosted by Internet portals. The research on these algorithms mostly emphasizes on classification accuracy, but their efficiency on execution is rarely addressed. However, their efficiency should deserve attention for on-line applications such as Web filtering so that text classification will not slow down these applications significantly.

This work presents a simple, but effective *early decision* algorithm to accelerate Web filtering. The algorithm is based on the observation that it is possible to make the filtering decision before scanning the entire content, as soon as the content can be confirmed with a high probability that it really belongs to a certain category. The fast decision is particularly important, since most Web content is normally allowed and should pass the filter as soon as possible.

The rest of this paper is organized as follows. Section 2 reviews related work in Web filtering. The early decision algorithm is presented in Sect. 3. Section 4 presents the accuracy and efficiency of this algorithm from experimental results, and discusses the issues of deployment in a practical environment. Section 5 finally concludes this work.

2. Related Work in Web Filtering

2.1 Approaches of Web Filtering

A Web filtering system can either block HTTP requests according to their URLs, or block the Web content using sev-

Manuscript received February 16, 2006.
Manuscript revised August 25, 2007.

[†]The authors are with High-speed Lab, Department of Computer Science, National Chiao Tung University, Taiwan.
[††]The author is with Department of Computer Science, National Chiao Tung University, Taiwan.
[†††]The author is with Department of Information Management, National Taiwan University of Science and Technology, Taiwan.
a) E-mail: pclin@cis.nctu.edu.tw
b) E-mail: mdliu@cis.nctu.edu.tw
c) E-mail: ydlin@cis.nctu.edu.tw
d) E-mail: laiyc@cs.ntust.edu.tw

DOI: 10.1093/ietisy/e91–d.2.251

eral approaches to be discussed later. The former approach maintains a large database of banned URLs. If the URL in a request is found in the database, the request is blocked. The database is frequently updated by the collaborative effort of human reviewers (may include the users).

URL blocking is very efficient in processing, and the content on the banned sites will not occupy the bandwidth of the download link. However, since Web sites on the Internet change very often and new sites grow extremely fast, the database is unlikely to keep pace with the dynamic change of Web sites. Hence the system may fail to ban some sites that should be banned.

Blocking the Web content can remedy the insufficiency of URL blocking. Several types of information can help to determine whether a Web page should be blocked. The Platform for Internet Content Selection (PICS) specification (`http://www.w3.org/PICS/`) allows the content publishers to rate and label Web content so that a Web filtering system can identify the category and judge the offensiveness of the content according to the PICS information. However, labeling the Web content is voluntary. Publishers of banned content may not want to label the content and let their sites be banned. Hence a system cannot rely solely on the PICS information to judge whether the content should be blocked or not.

Another simple approach is looking for offensive keywords in the Web content. The keywords should be carefully selected, or false positives are likely to happen. For example, using 'sex' as a keyword could possibly block Web content about sex education. Therefore, detailed content analysis is often desired rather than simple keyword blocking.

Content analysis is generally based on machine-learning methods. It involves looking for representative features that tell the category of the content. The features could be keywords, hyperlinks, images and so on [5]. The classifier on a Web filtering system first learns the features from a training set of Web pages collected from both banned and allowed categories off-line. After the learning, the classifier is able to judge the Web content according to the features. Most features are text-based in practice, because text classification is more efficient than image analysis and can classify content in categories other than pornography.

2.2 Text Classification Algorithms

Text classification is an important part in Web filtering because the text in Web content provide rich features for filtering. Yang et al. and Sebastiani [6], [7] surveyed and compared comprehensively existing text classification algorithms, such as support vector machine (SVM), *k*-nearest neighbor (kNN), neural network (NNet), decision tree and naïve Bayesian (NB). These algorithms are shown to achieve around 80% or higher in accuracy, measured by the harmonic average of recall and precision, where recall is defined to be the ratio of the number of correct positive predictions divided by the number of positive examples, and precision is the ratio of the number of correct positive predictions divided by the number of positive predictions.

Support vector machine (SVM) uses a process to find a decision surface that can separate positive examples and negative examples in a multi-dimensional feature space, in which training documents are represented as vectors. SVM is efficient and can handle a large number of training examples, but it would be difficult to find a kernel function to map vectors to the multi-dimensional space so that both positive and negative examples are roughly linearly separable [5].

The *k*-nearest neighbor (kNN) method labels the training examples in the feature space to their categories. In the classification stage, the kNN method selects *k* most similar documents (measured by the distance in the feature space) in the training examples, and assigns the test document to the label that is the most frequent one among the *k* nearest training documents. The kNN method is very simple, but it had better allow the possibility that a document is assigned to more than one category, or the accuracy may be degraded [6].

The neural network method (NNet) has been widely studied in artificial intelligence. A neural network is an adaptive system that consists of a group of interconnected artificial neurons. The system can be trained to change its internal states to reflect the association of the documents and their categories. Although neural networks are efficient in handling both linearly and non-linearly separable examples, the classification decision is not easily understandable because of the "black-box" nature of the neural network.

A decision tree represents a test on an attribute as an internal node and the test results as outgoing branches of that node. A document is classified by traversing from the root node to a set of internal nodes successively based on the test results on the attributes, until a leaf node is reached. The decision is easily interpretable, but requires to carefully choose the attributes to avoid the over-fitting problem.

Naïve Bayesian classification uses a probabilistic model, in which the probabilities of words in a document that belongs to each category are estimated. The probabilities can be used to estimate the most likely category of a test document. We leave the details of NB in Sect. 3.2 because our work is based on NB.

Some research work has attempted to exploit the structural information, such as hyperlinks and meta-information to improve the accuracy [8], [9]. These methods require parsing the Web content to extract the semantical information. This direction is currently beyond the scope of this work because the parsing takes more time than our model of viewing the text as a sequence of words, and may not be so efficient for real-time filtering.

3. The Early Decision Algorithm

This key idea of the *early decision* algorithm to accelerate Web filtering is that making the filtering decision is possible before scanning the entire content, as soon as the content is confirmed to really belongs to a certain category with a high probability. The fast decision is particularly important for on-line filtering because most Web content is allowed and should pass the filter as soon as possible.

Among the aforementioned algorithms in Sect. 2.2, we herein choose naïve Bayesian classification to be the basis of the early decision algorithm because its computation can be easily turned into score accumulation. The probability that a test document belongs to each category can be easily estimated as the classifier scans along the document. However, we believe that NB classification is by no means the only algorithm that can make an early decision from part of the Web content. Other text classification algorithms can follow the similar principle to be introduced in this section to accelerate Web content classification.

3.1 Keyword Distribution

The early decision algorithm can make the filtering decision as soon as possible, by scanning only the front part of the Web content. Because the algorithm does not have to scan the entire content, the filtering is much faster. The trick is just like one may not have to wait until the end of a speech to know its topic, as long as the front part of the speech provides sufficient information about the topic.

The feasibility of the early decision algorithm comes from the observation that the front part of the Web content has adequate keywords to indicate whether the content should be blocked or passed. The keywords herein are defined to be words from the banned categories with high "information gain" [11], which, simply put, measures how indicative a word is to help distinguish the category of content from the others. To justify the feasibility, we investigated the keyword distribution in typical Web content collected from the YAHOO directory service (`http://www.yahoo.com`). Figure 1 presents the average distribution in Web content of both the banned and allowed categories. On the horizontal axis is the keyword position normalized by the content length of each Web page. The position is represented in percentage. For example, if a keyword is the 50-th word in a 100-word Web page, then its position is at 50%. On the vertical axis is the appearance probability of keywords in the positions throughout the Web content. The probability is also represented in percentage.

According to Fig. 1, the keywords in the banned categories start to appear more frequently than those in the allowed categories since the front part of the Web content. In other words, keywords from the front partial content can provide the clue to identify the category that the entire content belong to. Therefore, it is feasible to make the filtering decision before scanning the entire content.

Although the above observation is generally true in normal conditions, a malicious user may deliberately stuff irrelevant content in the front part of the Web page to deceive the filter. The deception is not difficult to avoid. First, the early decision algorithm strips the HTML tags during the content analysis because the tags are generally used to specify the attributes of the content, and will not be displayed on the browser. If the irrelevant content is hidden inside the HTML tags, it will be ignored and unable to deceive the filter. Second, if the irrelevant content is in the Web text outside the tags, it will be displayed on the browser, and will also confuse the viewer who browses the content. This deception approach will lead to a great limitation on the layout design of the Web pages.

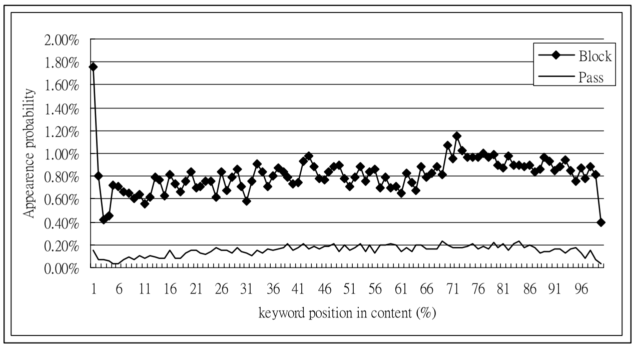

Fig. 1 The keyword distribution in the Web content of both the banned and the allowed categories.

3.2 Naïve Bayesian Classification

The NB classification is divided into two stages: training and classification. In the training stage, the classifier learns the probabilistic parameters of the generative model from a set of training documents, $D = \{d_1, \ldots, d_{|D|}\}$. Each document consists of an ordered sequence of words from a vocabulary set $V = \{w_1, w_2, \ldots w_{|V|}\}$ and is associated with some category from a set of categories $C = \{c_1, c_2, \ldots, c_{|C|}\}$. Two types of parameters are included in the model [10]: (1) $P(w_t|c_j)$: the estimated probability that word w_t appears in the documents of category c_j and (2) $P(c_j)$: the estimated probability of category c_j in the training documents. The former parameter is derived by

$$P(w_t|c_j) = \frac{1 + \sum_{i=1}^{|D|} N_1(w_t, d_i|d_i \in c_j)}{|V| + \sum_{t=1}^{|V|} \sum_{i=1}^{|D|} N_1(w_t, d_i|d_i \in c_j)}, \tag{1}$$

where $N_1(w_t, d_i)$ is the times word w_t appears in document d_i, and

$$P(c_j) = \frac{1 + \sum_{i=1}^{|D|} N_2(d_i, c_j)}{|C| + |D|}, \tag{2}$$

where $N_2(d_i, c_j)$ is 1 if $d_i \in c_j$, or 0 otherwise.

Notably, the above two equations are filtered by Laplace smoothing to avoid estimating probabilities to be zero. In the classification stage, the posterior probability $P(c_j|d_i)$ that a test document d_i belongs to each category c_j is computed. The category c_j that maximizes $P(c_j|d_i)$ is the one that document d_i belongs to most likely. $P(c_j|d_i)$ is derived by

$$\begin{aligned} P(c_j|d_i) &= \frac{P(c_j)P(d_i|c_j)}{P(d_i)} \\ &= \frac{P(c_j)\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_j)}{P(d_i)}, \end{aligned} \tag{3}$$

where $w_{d_i,k}$ is the k-th word in document d_i. The document d_i is viewed as an ordered sequence $< w_{d_i,1}, w_{d_i,2}, \ldots, w_{d_i,|d_i|} >$, with the assumption that the probability of a word occurrence is independent of its position in the document, given the category c_j. Therefore, $P(d_i|c_j)$ can be written as the product of $P(w_{d_i,k}|c_j)$, for $k = 1 \ldots |d_i|$. Taking the logarithm on both sides of Eq. (3) simplifies the computation of the posterior probability $P(c_j|d_i)$ from a series of multiplications to a series of additions. The computation then becomes

$$logP(c_j|d_i) = log\frac{P(c_j)}{P(d_i)} + \sum_{k=1}^{|d_i|} logP(w_{d_i,k}|c_j). \tag{4}$$

Since the logarithm function is increasing and $log\frac{P(c_j)}{P(d_i)}$ is kept constant in the classification stage, accumulating only the term $logP(w_{d_i,k}|c_j)$ during text scanning is sufficient to derive the maximum of $P(c_j|d_i)$. The score of each word $w_t \in V$ that belongs to category c_j can be defined by $logP(w_t|c_j)$. These scores are pre-computed in the training stage and accumulated for each word $w_{d_i,k}$, while the content is scanned from the beginning to the end. The computation in Eq. (4) is very fast because only the addition operation is performed for each word. This is why we select NB as the basis of the early decision algorithm herein.

3.3 Keyword Extraction in the Training Stage

The classifier of the early decision algorithm is trained off-line from sample Web content in both the banned and allowed categories. We used the *Rainbow* program [11] and its library, *Bow*, from Carnegie Mellon University to train the classifier. They can extract keywords with high information gain as the features from the training categories. Common words, such as "the", "of" and so on (called stop words) should be dropped from the set of keywords because they help little in classification (with low information gain).

Automatic keyword extraction could become complex for Web content in some oriental languages. Unlike in western languages (e.g., English), no space characters delimit the words in these languages. Hence it is unclear how many characters compose a semantic "word", not to mention a keyword.

We suggest the N-gram method [12], [13] for extracting keywords in CJK (Chinese, Japanese and Korean) languages, where N-gram means N characters. The tool to extract keywords, *Rainbow*, is modified to do so. The modified algorithm for the N-gram keyword extraction is detailed in [14]. Simply put, the N-gram algorithm looks for keywords of various lengths (i.e., the N-gram) and determines the best length of each keyword.

A simple rule can eliminate redundant keywords in the N-gram extraction: If (1) a string s is a substring of another string t, and (2) every appearance of s implies the appearance of t, only t is left as the keyword. We prefer leaving only the long keyword (i.e., t) to reduce the possibility of false positives due to short keywords. The word s can be eliminated without harm because it provides no more clues for classification than t does. The score of a concatenated string is defined to be the maximum score of each composite substring. For instance, $Score(st) = max\{Score(s), Score(t)\}$, where $Score(s)$ denotes the score of string s.

3.4 The Filtering Stage

In the filtering stage, the incoming content of a Web page is scanned from the beginning for the extracted keywords. Suppose $n\%$ of the page content has been scanned. The event $E_{n,m}$ denotes that the accumulated score has reached m when the filter has scanned $n\%$ of the content. The probability that this content belongs to a category $c_j \in C$ is derived from

$$P(c_j|E_{n,m}) = \frac{P(E_{n,m}|c_j)P(c_j)}{P(E_{n,m}|c_j)P(c_j) + P(E_{n,m}|c'_j)P(c'_j)}. \tag{5}$$

- $E_{n,m}$: the event that when the filter has scanned $n\%$ of the content, the accumulated score has reached m.
- $P(c_j)$: the estimated probability that category c_j appears in typical Web content. $P(c_j)$ can be estimated from sample traffic beforehand or dynamically measured in a running environment by recording and analyzing actual Web content.
- $P(c'_j)$: the estimated probability that category c_j does not appear in typical Web content. $P(c'_j)$ = 1 - $P(c_j)$.
- $P(E_{n,m}|c_j)$: the estimated probability that $E_{n,m}$ happens given that the content belongs to category c_j. The estimate of $P(E_{n,m}|c_j)$ is the number of Web pages in c_j that $E_{n,m}$ happens divided by the number of Web pages in c_j.
- $P(E_{n,m}|c'_j)$: defined similarly as $P(E_{n,m}|c_j)$, except that c_j is replaced with c'_j.

To accelerate the computation of $P(c_j|E_{n,m})$, two two-dimensional tables of $P(E_{n,m}|c_j)$ and $P(E_{n,m}|c'_j)$ are built for each n and m from the training sets in the training stage, for each $c_j \in C$. Note that the table look-up is a little bit tricky herein. The accumulated score of m in the filtering may not exactly match any subscripts of m on the tables because the subscripts of m on the tables are discrete and finite. Therefore, the probabilities $P(E_{n,m}|c_j)$ and $P(E_{n,m}|c'_j)$ are derived in practice by looking up the tables with n and the maximum subscript of m no larger than the accumulated score of m.

Figure 2 presents the pseudo-code of the early decision algorithm. Two thresholds, T_{bypass} and T_{block}, are set arbitrarily to be 0.1 and 0.9 herein. Let PCE_j be the estimate of $P(c_j|E_{n,m})$. If $PCE_j < T_{bypass}$, for all c_j in the list of banned categories, the content is unlikely to be in a banned category, and the remaining content can be bypassed. On the contrary, if there exists some c_j in the list of banned categories such that $PCE_j > T_{block}$, the content is likely to belong to c_j, and should be blocked by the filter. The computation overhead in Eq. (5) is nearly negligible, occupying less than 0.1% of the total classification time in our profiling.

The classifier should scan a minimum amount of the

```
$Earlybypass \leftarrow False$;
$Earlyblock \leftarrow False$;
$n \leftarrow 0$;
while not end of content do
  Skip stop words and the HTML tags.
  Read the next keyword;
  $n \leftarrow$ percentage of the content that has been scanned; {scanning at least $min\_scan\%$ of content}
  if $n > min\_scan$ then
    $m \leftarrow$ the accumulated score;
    for each banned category $c_j$ do
      $PEC_j \leftarrow P(E_{n,m}|c_j)$ of current scanning position;
      $PEC'_j \leftarrow P(E_{n,m}|c'_j)$ of current scanning position;
      $PCE_j \leftarrow (PEC_jP(c_j))/(PEC_jP(c_j) + PEC'_jP(c'_j))$;
    end for
    if $(\forall c_j, PCE_j \leq T_{bypass})$ then
      $Earlybypass \leftarrow True$;
      Exit;
    end if
    if $(\exists c_j, PCE_j \geq T_{block})$ then
      $Earlyblock \leftarrow True$;
      Exit;
    end if
  end if
end while
```

Fig. 2 The pseudo-code of the early decision algorithm.

content in a Web page to avoid deciding too early from only the very front part of the content. If the content in a banned category happens to not have keywords in this part, false negatives may occurs. The parameter *min_scan* in Fig. 2 denotes the minimum amount in percentage. We set *min_scan=15* arbitrarily since it is sufficient to effectively avoid false negatives in our experiment.

4. Experiments

4.1 Performance Metrics

To measure the accuracy of the early decision algorithm, we use the F1 measure that combines the recall and the precision by taking the harmonic average of them with equal weight [15]. We also use two metrics: the *average scan rate* (ASR) and the throughput, defined by

$$ASR = \frac{Total\ bytes\ that\ are\ scanned}{Total\ bytes\ in\ the\ content} \times 100\% \qquad (6)$$

and

$$Throughput = \frac{Total\ bits\ in\ the\ content}{Total\ execution\ time\ (sec)}, \qquad (7)$$

to measure the effectiveness of acceleration. The former reflects the percentage of the Web content that is scanned in the early decision algorithm, and the latter shows the actual throughput in Web content filtering.

4.2 Experimental Results and Discussion

From the experiment, totally 300 sample Web pages in four typically banned categories, Pornography, Game, Online-Shopping and Finance, are randomly collected from the YAHOO directory service, and another 300 pages are from other categories to serve as the allowed content. The early decision algorithm searches the Web content with a multiple-string matching algorithm for the keywords extracted in the training stage. A sub-linear time algorithm (e.g., the Wu-Manber algorithm, which can skip characters in the text by nearly the length of the shortest keywords [16]) hardly helps the performance here because short keywords are not uncommon in natural languages. The filtering routine is implemented on Lex [17], which uses the linear-time Aho-Corasick algorithm [18], and thus its performance is independent of the keyword lengths.

Table 1 Comparison of classification accuracy in four banned categories. Here Pr denotes the precision, Re denotes the recall, and F1 denotes the F1 measure, which is the harmonic average of Pr and Re.

Category	Original Bayesian classifier			Early decision		
	Pr	Re	F1	Pr	Re	F1
Porn	1.00	.993	.996	.977	.918	.947
Game	1.00	.971	.985	.958	.819	.883
Shopping	1.00	.975	.987	.866	.750	.804
Finance	.896	1.00	.945	.964	.900	.931

Table 2 Average accuracy and average scan rate in the early decision algorithm.

Banned			Allowed			ASR	ASR
Pr	Re	F1	Pr	Re	F1	(Banned)	(Allowed)
.941	.847	.892	.947	.920	.934	17.22%	26.51%

The accuracy of the original Bayesian classifier, which scans the entire content, is compared with that of the early decision algorithm for the four banned categories in Table 1. Among the categories in comparison, only the shopping category presents noticeable accuracy degradation, while the others remain fairly good accuracy. After a careful examination, we observed that the Web pages in the shopping category have many common words that also appear in allowed categories. Therefore, the score accumulation from keywords is slow. Lacking representative keywords reduces the accuracy if the scanned part is not long enough. We consider the categorization should be more specific in this case so that precise keywords can be extracted.

Table 2 presents the average filtering accuracy of the content in the four banned categories (summarized from Table 1) and the allowed categories. The accuracy of both types of content with the early decision algorithm is close to that when the entire content is scanned. The speed-up is obvious because the early decision algorithm scans only 17.22% of content in the banned categories and 26.51% in the allowed categories on average. A large portion of the Web content is bypassed in the Web filtering, and the classification time is significantly shortened.

False positives of allowed content may be considered unacceptable in a practical environment, and a high threshold T_{block} is set. Lifting the threshold T_{block} to 1.0 can effectively avoid false positives in the allowed categories, as shown from the high precision in Table 3. Note that lifting T_{block} also leads to more false negatives in the banned categories because some banned content is unable to reach such a high threshold. Therefore, deciding a proper threshold is a

Table 3 Accuracy in the setting of no false positives in allowed content.

Category	Original Bayesian classifier			Early decision		
	Pr	Re	F1	Pr	Re	F1
Porn	.977	.918	.947	1.00	.773	.871
Game	.958	.819	.883	1.00	.623	.767
Shopping	.866	.750	.804	1.00	.55	.709
Finance	.964	.900	.931	1.00	.730	.843

Table 4 Comparison of the throughput of the early decision algorithm and the original Bayesian classifier.

Algorithm	Execution time (ms)	Throughput (Mb/s)
Original Bayesian classifier	1333.77	41.05
Early decision for banned content	241.89	226.36
Early decision for allowed content	239.90	156.68

tradeoff in practice.

Both the execution time and throughput of the early decision algorithm are compared with those of the original Bayesian classifier to manifest the improvement. Both classifiers are implemented on a PC with Intel Pentium III 700 MHz and 64 MB of RAM. Table 4 presents the comparison results of filtering both the banned and allowed content. The results show a significant improvement in throughput, about five times higher than that of the original Bayesian classifier for banned content and nearly four times higher for allowed content.

Many commercial products and open source packages in our investigation, such as DansGuardian [19], can block a page as soon as the score accumulation achieves the given threshold configured arbitrarily by the user. In contrast, the early decision algorithm compares the threshold with the probability estimation of the classification, rather than the score itself. This approach has two advantages over that in DansGuardian. First, the two parameters, T_{bypass} and T_{block}, have stronger association with the accuracy than the threshold on the score in DansGuardian. Therefore, it is easier to customize the thresholds in the early decision algorithm to achieve the desired accuracy. In comparison, deciding a proper threshold in DansGuardian to get the desired accuracy will take more efforts in trial and error, since the threshold provides few clues to the accuracy. Second, the early decision algorithm accelerates not only filtering blocked Web pages, but also filtering allowed pages. The advantage is particularly significant when the Web accesses are mostly allowed content.

The early decision algorithm is also implemented on the content analysis of DansGuardian by modifying its filtering code. In our testing samples, the throughput is about three times higher than that in the original version of DansGuardian. The increasing primarily comes from the better criterion in the content filtering and the acceleration of filtering the allowed content. The principle of early decision can also be implemented into the content filtering process in other Web filtering products.

4.3 Practical Consideration in Deployment

With the increasing number of categories to be classified, ambiguity between these categories may increase. In our opinion, the proper place to perform Web content filtering is restricted to the edge devices for performance reason. Such edge devices usually require fewer banned categories, and thus the problem with increasing number of categories is not that serious.

The two thresholds, T_{bypass} and T_{block}, can be tuned according to the tradeoffs between accuracy and efficiency. The accuracy can be increased at the cost of less efficiency by decreasing T_{bypass} or increasing T_{block}, and the efficiency can be increased at the cost of less accuracy by increasing T_{bypass} or decreasing T_{block}. The tuning depends on which is more important for an organization: accuracy or efficiency.

Even though the early decision algorithm significantly speed up the filtering decision, we believe that it should complement other Web filtering approaches, especially URL blocking, but not to replace them. First, URL blocking is faster than content analysis since a URL has much fewer characters to be processed than the Web content. Besides, if a banned URL is successfully blocked, no network bandwidth will be wasted to download the banned content. As discussed in Sect. 2.1, content analysis is still needed to successfully catch the banned content. The early decision can accelerate this part significantly. Second, Web content may contain images, video, Flash objects, Java applets and so on, which are non-trivial to analyze. Analyzing these objects is beyond the scope of this paper, but it is still helpful to increase the accuracy in filtering the Web content.

In summary, a Web filtering system can support various approaches in practice, just like many commercial products and open source packages. The system first blocks URLs according to the database of banned URLs that is constantly maintained. To reduce false negatives due to the outdated database, content analysis can catch the banned content whose source is not in the URL database. The early decision algorithm can speed up content analysis to reduce the latency perceived by the user and to increase the system throughput. Although analyzing other types of objects in the Web content, such as images, could increase the accuracy, it is still a trade-off between performance and processing effort so far. It depends on the user to evaluate whether turning on such an analysis is worthwhile.

5. Conclusion

This work addresses the problem with possibly long delay in text classification algorithms that perform run-time content analysis in Web filtering. We present an early decision algorithm to decide to either block or pass the content as early as possible. A significant performance improvement is observed. The throughput is increased by about five times higher for banned content and nearly four times higher for allowed content, while the accuracy remains fairly good.

In the F1 measure, the accuracy is about 89% for filtering banned content, and about 93% for allowed content.

The early decision algorithm is simple but effective. The same rationale behind this algorithm can be applied to other content filtering applications as well, such as anti-spam. The algorithm can be combined with more features other than keywords from the text to further increase the overall accuracy of the content filter. Besides, the filtering can be further accelerated by combining the URL-based method with the cached results. That is, by caching the decisions on URLs of the filtered Web pages, duplicate filtering on the same Web page can be avoided. Content analysis can be skipped if the cached URL is matched. The maintenance of the URL database is also facilitated.

Acknowledgment

This work was supported in part by the Taiwan Natioanl Science Council's Program of Excellence in Research, and in part by grants from Cisco and Intel.

References

[1] P.Y. Lee, S.C. Hui, and A.C. Fong,"Neural networks for Web content filtering," IEEE Intelligent Systems, vol.17, no.5, pp.48–57, Sept.-Oct., 2002.
[2] Internet Filter Review 2006, available at http://internet-filter-review.toptenreviews.com/
[3] J.Z. Wang, Integrated region-based image retrieval, pp.107–122, Kluwer Academic Publishers, Dordrecht, Holland, 2001.
[4] J.Z. Wang, WIPE: Wavelet image pornography elimination, available at http://wang.ist.psu.edu/docs/projects/wipe.html
[5] M. Hamammi, Y. Chahir, and L. Chen, "WebGuard: A Web filtering engine combining textual, structural and visual content-based analysis," IEEE Trans. Knowl. Data Eng., vol.18, no.2, pp.272–284, Feb. 2006.
[6] Y. Yang and X. Liu, "A re-examination of text categorization methods," Proc. SIGIR'99, 22nd ACM International Conference on Research and Development in Information Retrieval, pp.42–49, 1999.
[7] F. Sebastiani, "Machine learning in automated text categorization," ACM Comput. Surv., vol.34, no.1, pp.1–47, March 2002.
[8] E.J. Glover, K. Tsioutsiouliklis, S. Lawrence, D.M. Pennock, and G.W. Flake, "Using Web structure for classifying and describing Web pages," Proc. World Wide Web (WWW), pp.562–569, 2002.
[9] G. Attardi, A. Gulli, and F. Sebastiani, "Automatic Web page categorization by link and context analysis," Proc. THAI-99, First European Symp. Telematics, Hypermedia, and Artificial Intelligence, pp.105–119, 1999.
[10] T. Mitchell, Machine learning, McGraw Hill, 1996.
[11] The Rainbow library, Available at http://www-2.cs.cmu.edu/~mccallum/bow/rainbow/
[12] F. Peng and D. Schuurmans, "Combining naïve Bayes and n-gram language models for text classification," 25th European Conference on Information Retrieval Research (ECIR), pp.335–350, Dec. 2003.
[13] H.H. Chen and J.C. Lee, "Identification and classification of proper nouns in Chinese texts," Proc. 16th International Conference on Computational Linguistics, pp.222–229, Aug. 1996.
[14] F.H. Huang and Y.D. Lin, Evaluating the accuracy and efficiency of a multi-language content filter, MS.c. Thesis, Department of Computer and Information Science, National Chiao Tung University, 2003.
[15] C.J. Rijsbergen, Information retrieval, Butterworths, London, 1979.
[16] S. Wu and U. Manber, "A fast algorithm for multi-pattern searching," Technical Report TR-94-17, University of Arizona, 1994.
[17] M.E. Lesk, "Lex - A lexical analyzer generator," Comp. Sci. Tech. Rep., no.39. Bell Laboratories, 1975.
[18] A.V. Aho and M.J. Corasick, "Efficient string matching: An aid to bibliographic search," Commun. ACM, vol.18, pp.333–340, 1975.
[19] The DansGuardian Web filtering tools, Available at http://dansguardian.org

Po-Ching Lin received the bachelor's degree in Computer and Information Education from National Taiwan Normal University, Taipei, Taiwan in 1995, and the M.S. degree in Computer Science from National Chiao Tung University, Hsinchu, Taiwan in 2001. He is a Ph.D. candidate of Computer Science in National Chiao Tung University. His research interests include content networking, algorithm designing and embedded hardware software co-design.

Ming-Dao Liu received the bachelor's degree and the M.S. degree in Computer Science from National Chiao Tung University, Hsinchu, Taiwan in 2001 and 2003. His research interests include network security and content networking.

Ying-Dar Lin received the bachelor's degree in Computer Science and Information Engineering from National Taiwan University in 1988, and the M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles in 1990 and 1993. He joined the faculty of the Department of Computer and Information Science since 1993. From 2005, he is the director of the graduate Institute of Network Engineering, and then the director of Computer and Network Center since 2006. He is also the founder and director of Network Benchmarking Lab since 2002. His research interests include design, analysis, implementation and benchmarking of network protocols and algorithms, wire-speed switching and routing, quality of services, network security, content networking, network processors and SoCs, and embedded hardware software co-design.

Yuan-Cheng Lai received the bachelor's degree and the M.S. degree in Computer Science and Information Engineering from National Taiwan University in 1988 and 1990, and the Ph.D. degree from Computer and Information Science, National Chiao Tung University in 1997. He joined the faculty of National Cheng-Kung University, Tainan, Taiwan in 1998. He is an associate professor in Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include high-speed networking, wireless network and network performance evaluation, Internet applications.