An Early Decision Algorithm to Accelerate Web Content Filtering

Ying-Dar Lin[1], Po-Ching Lin[1], Ming-Dao Liu[1] and Yuan-Cheng Lai[2]

[1]Department of Computer Science
National Chiao Tung University, HsinChu, Taiwan
Email: {ydlin, pclin, mdliu}@cis.nctu.edu.tw

[2]Department of Information Management
National Taiwan University of Science and Technology, Taipei, Taiwan
Email: laiyc@cs.ntust.edu.tw

Abstract. Real-time content analysis can be a bottleneck in Web filtering. This work presents a simple, but effective *early decision* algorithm to accelerate the filtering process by examining only part of the Web content. The 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 should belong to a banned or an allowable category. The experiments show the algorithms can examine only around one-fourth of the Web content on average, while the accuracy remains fairly good: 89% in the banned content and 93% in the allowable content. This algorithm can complement other Web filtering approaches to filter the Web content with high efficiency.

1 Introduction

Massive volume of Internet content is widely accessible nowadays. One can easily view improper content at will without access control. For example, an employee may watch stock information during office hours. Web filtering products can enforce the access control. The up-to-date products have widely adopted content analysis besides the *URL-based* approach [1]. Content analysis works with the URL-based approach to relieve the efforts of maintaining the URL list and to reduce the number of false negatives. The analysis classifies the Web content to a certain category first, and makes the filtering decision, either to block or to pass the content.

Despite the ongoing research on image and video content classification, *text classification* is typically the most efficient approach to Web content analysis. Many text classification algorithms have been around with high accuracy. They are often assumed to run off-line, so their execution time is rarely discussed. However, the efficiency of these algorithms is critical because slow content analysis in Web filtering incurs long user response time. The issues of accelerating the analysis should deserve attention.

This work presents a simple, but effective *early decision* algorithm to accelerate the filtering from the observation that the filtering decision can be made *before* scanning the *entire* content, as soon as the content can be classified into a certain category. A fast decision is particularly important since most Web content is normally allowable and should pass the filter as soon as possible.

The rest of this paper is organized as follows. Section 2 provides the background of this work. The *early decision* algorithm is described in Section 3. Section 4 exhibits the accuracy and efficiency of this algorithm from the experimental results and discusses the deployment issues in a practical environment. Finally, Section 5 concludes this work.

2 Background

Yang et al. and Sebastiani [2], [3] gave a comprehensive survey of existing text classification algorithms. These algorithms are shown to achieve around 80% of accuracy or higher, measured by the average of recall and precision. Recall is defined to be the ratio of the number of correct positive predictions divided by the number of positive examples, while precision is the ratio of the number of correct positive predictions divided by the number of positive predictions. Among these algorithms, we choose Naïve Bayesian classification as the base of the *early decision* algorithm for its simplicity. Other classification algorithms can follow the principle to accelerate the classification.

The Bayesian classification is divided into two stages: training and classification. The training stage learns the probabilistic parameters of the generative model from a set of training documents, D={d_1,…..,$d_{|D|}$}. Each document consists of a sequence of words from a vocabulary set V={w_1, w_2, … $w_{|V|}$} and has been labeled with some category from a set of categories C = {c_1, c_2, …, $c_{|c|}$} before the training. Two types of parameters are included in the model: (1) P(w_t|c_j): the estimated probability of word w_t given category c_j and (2) P(c_j): the estimated probability of category c_j. These parameters are derived by [4]

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

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

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

In the classification stage, the posterior probability P(c_j|d_i) with which a test document d_i belongs to category c_j is derived. The category c_j that maximizes P(c_j|d_i) is the one that d_i belongs to. P(c_j|d_i) is derived by

$$P(c_j \mid d_i) = \frac{P(c_j)P(d_i \mid c_j)}{P(d_i)} = \frac{P(c_j)\prod_{k=1}^{di} P(w_{di,k} \mid c_j)}{P(d_i)} \quad , \qquad \mathbf{(3)}$$

where $w_{di,k}$ is the k-th word in document d_i. Notice that the document d_i is viewed as an ordered sequence <$w_{di,1}$, $w_{di,2}$, …, $w_{di,|di|}$>, with the assumption that the probability of a word occurrence is independent of its position in the document, given the document category c_j, so that P(d_i|c_j) can be written as the product of individual probabilities P($w_{di,k}$|c_j).

3 The Early Decision Algorithm

The philosophy behind the *early decision* algorithm is to make the filtering decision from the front partial Web content. Fig. 1 presents the average keyword distribution of both banned and allowable Web pages in our investigation. The keyword position is normalized by the page length. The keywords in almost all Web pages tend to be distributed uniformly throughout the content or appear more in the front part according to this investigation. The Web content in a banned category starts to exhibit much more keywords than that in an allowable category since the front part. In other words, keywords from the front partial content can reveal the category of the Web content and serve as the clues to filtering.

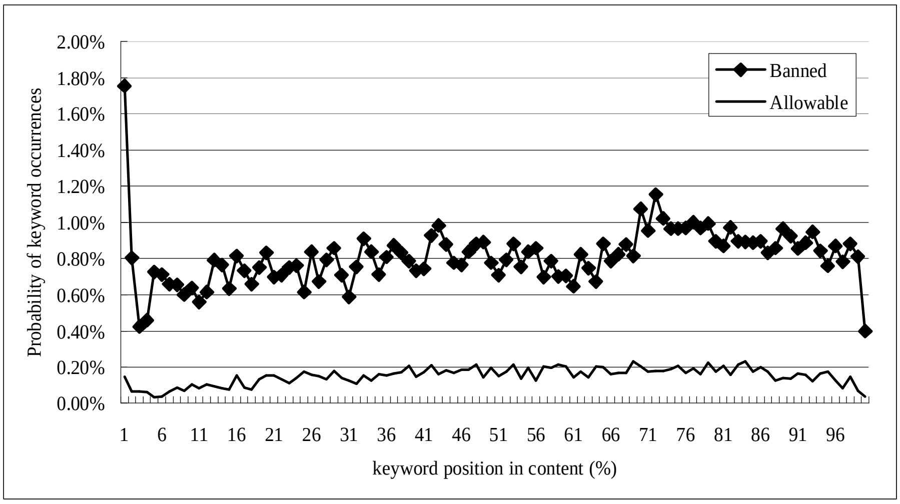

Fig. 1. The distribution of keyword positions in typical Web pages

Like the Bayesian classification, the filtering engine is trained off-line from the Web content in the banned categories. The *Bow* library and its front-end, *Rainbow* [5 perform the training herein, extracting keywords as the features from the target categories. The keywords with the information gains larger than a threshold are selected. Stop words, such as “the”, “of” and so on, should be dropped because they help little in classification. The words inside the HTML tags are also ignored so that a malicious user cannot stuff unrelated content in the tags, particular in the front part of the Web page, to deceive the filter. If the malicious user fills the Web text outside the tags with irrelevant content to confuse the filter, the irrelevant content will be displayed in the browser and will spoil the layout of the Web pages – a great limitation on the design of the Web pages.

The score of keyword w_t that should belong to a category c_j is defined to be logP($w_t|c_j$), which can be derived in the training stage. Taking the logarithm simplifies the computation of the posterior probability P($c_j|d_i$) from multiplication operations to score accumulation with independence assumption between words [4]. The scores are accumulated while the content is scanned from the front to the end.

In the filtering stage, given n% of the content that has been scanned and the score m or less that has been accumulated, the probability that the content should belong to a category c is derived from

$$P(c \mid D(n,m)) = \frac{P(D(n,m) \mid c)P(c)}{P(D(n,m) \mid c)P(c) + P(D(n,m) \mid c')P(c')} . \qquad \textbf{(4)}$$

1. $D(n,m)$: the event that the filter has read n% of the content and has observed the score accumulation m or less.
2. $P(c)$: the estimated probability that category c appears in typical Web content.
3. $P(c')$: the estimated probability that category c does not appear in typical Web content. $P(c') = 1 - P(c)$.
4. $P(D(n,m) \mid c)$: the estimated probability that $D(n,m)$ happens given that the content belongs to category c. The estimate of $P(D(n,m) \mid c)$ is the number of Web pages in c that $D(n,m)$ happens divided by the number of Web pages in c.
5. $P(D(n,m) \mid c')$: defined similarly as $P(D(n,m) \mid c)$, except that c is replaced with c’.

In the training phase, two two-dimensional indexed tables of $P(D(n, m)|c_i)$ and $P(D(n, m)|c_i')$ are built for each n and m from the training examples, where $c_i \in$ C. The values of $P(c_i)$ and $P(c_i$’) can be estimated beforehand or dynamically tuned in a running environment by recording and analyzing actual Web content. Fig. 2 presents the *early decision* algorithm. Two thresholds, $\mathrm{T_{bypass}}$ and $\mathrm{T_{block}}$, are defined to be 0.1 and 0.9 herein. PCD_i is the estimate that the content should belong to a category c_i. If PCD_i is less than $\mathrm{T_{bypass}}$ for all c_i in the list of banned categories, this means the content is unlikely to be banned and the remaining content should be bypassed. In contrast, if there exists some c_i in the list of banned categories such that PCD_i is larger then T_{block}, this means the content is likely to belong to c_i and should be blocked by the filter. A minimum of the content should be scanned in the process to avoid deciding too early from only the little front part of the content, which may render the filtering result incorrect.

```
ıe Early Decision algorithm:
ırlybypass ß False;
ırlyblock ß False;
 ß 0;
) {
  Read next keyword; // Skip stop words and the HTML tags.
  n ß the percentage of content that has been scanned;
  m ß the accumulated score;
  If (n > Min_Scan) {
  // scanning at least Min_Scan% of document,
  // Min_Scan=10 herein
      For (each category $c_i$ in the set of banned categories) {
          $PDC_i$ ß $P(D(n, m)|c_i)$ of current scanning position;
          $PDC_i'$ ß $P(D(n, m)|c_i')$ of current scanning position;
          $PCD_i$ ß $(PDC_i*P(c_i))/(PDC_i*P(c_i)+PDC_i'*P(c_i'))$;
      } // end of For
      If (for all category $c_i$, $PCD_i < T_{bypass}$) {
         Earlybypass:=True;
         Exit;
      }
      If (for some category $c_i$, $PCD_i > T_{block}$) {
         Earlyblock:=True;
         Exit;
      }
  } // End of If (n > Min_Scan)
while (not end of content);
```

g. 2. The pseudo code of the early decision algorithm

4 Experiments

4.1 Performance metrics

The F1 measure, initially introduced by Van Rijsbergen [6], takes the harmonic average of the recall and the precision as the measure of accuracy. To measure the acceleration, the average scan ratio and the average throughput are defined by

$$Average\ scan\ ratio = \frac{Total\ bytes\ scanned}{Total\ bytes\ in\ the\ content}, \tag{5}$$

$$Average\,throughput = \frac{Total\ bits\ in\ the\ content\ being\ filtered}{Total\ execution\ time\ of\ the\ filtering\,(sec)}\,. \qquad (\mathbf{6})$$

4.2 Experimental results and discussion

Totally 300 Web pages are randomly collected from the YAHOO directory services [7] for the experiment in four typically banned categories: Pornography, Game, Online-Shopping and Finance. Another 300 pages are also randomly collected from other categories as the allowable content. The extracted keywords in the training stage are searched through the Web content with a multiple string matching algorithm. Since short patterns are not uncommon in natural languages, a sub-linear time algorithm, such as the Wu-Manber algorithm [8], can hardly take any advantages. The filtering algorithm is implemented with *Lex* [9], which is based on the Aho-Corasick algorithm [10], so the performance is less sensitive to short patterns.

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. Only the shopping category suffers noticeable accuracy degradation whereas the other categories remain fairly good accuracy. A careful examination reveals it is because the keywords in the shopping category include many ambiguous words that also appear in allowable content. If this is the case, more other examples from the category can be trained until better keywords that lead to higher accuracy are derived The filtering accuracy by averaging the accuracy of the four banned categories and that of allowable content are presented in Table 2. The filtering accuracy of both types of content with the early decision keeps fairly close to that when scanning the entire content, but only 17.22% of content in the banned categories and 26.51% in the allowable categories on average are scanned. This means a large portion of the Web content can be bypassed in Web filtering, and the execution time can be significantly shorter.

Table 1. Comparisons of classification accuracy

Algorithm/ Category	Porn	Game	Shopping	Finance
Original Bayesian classifier	Pr Re F1	Pr Re F1	Pr Re F1	Pr Re F1
	1.00 .993 .996	1.00 .971 .985	1.00 .975 .987	.896 1.00 .945
Early decision	.977 .918 .947	.958 .819 .883	.866 .750 .804	.964 0.90 .931

Table 2. Comparisons of filtering accuracy and average scan ratio

Algorithm/ Category types	Banned	Allowable	Average scan ratio in banned content	Average scan ratio in allowable content
Early decision	Pr Re F1 .941 .847 .892	Pr Re F1 .947 .920 .934	17.22%	26.51%

False positives of allowable traffic are usually unacceptable in a practical environment and a higher threshold T_{block} would be better. By lifting the threshold T_{block} to 1.0, false positives in the allowable categories can be almost avoided. Table 3 presents that a higher threshold also results in more false negatives in the banned categories because some banned content cannot reach such a high threshold. Choosing a proper threshold is a tradeoff in a practical environment.

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

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

Setting	Porn	Game	Shopping	Finance
Original Bayesian classifier	Pr Re F1 1.00 .993 .996	Pr Re F1 1.00 .971 .985	Pr Re F1 1.00 .975 .987	Pr Re F1 .896 1.00 .945
No false positives	1.00 .773 .871	1.00 .623 .767	1.00 .55 .709	1.00 .730 .843

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

Algorithm	Execution time (μs)	Throughput (Mb/s)
Original Bayesian classifier	1333772	41.05
Early decision for banned content	241887	226.36
Early decision for allowable content	239895	156.68

Many commercial products and open source packages in our investigation, such as DansGuardian [11], can block a page as the score accumulation achieves the given threshold configured arbitrarily by the users. The early decision algorithm compares the threshold with the probability estimation of the classification, rather than the score itself. The advantages of the early decision algorithm over the method in DansGuardian are two points. (1) The two parameters, T_{bypass} and T_{block}, have stronger association with the accuracy than the threshold of score in DansGuardian. The filtering can then be better tuned directly according to the desired accuracy. The choice of a proper threshold of score in DansGuardian to get the desired accuracy needs to take more efforts by trial and error. (2) The early decision algorithm accelerates not only filtering blocked Web pages, but also filtering allowable pages. The acceleration is particularly significant when the Web accesses are mostly allowable content.

The early decision algorithm is also implemented by modifying the filtering code in DansGuardian. The throughput is enhanced by about three times on average than that in DansGuardian in our testing samples because of the acceleration from the allowable content and the better criterion to decide the blocking. This algorithm can also be implemented into other Web filtering products to accelerate the filtering process.

4.3 Practical considerations 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. The problem with increasing number of categories is not that serious.

The early decision algorithm is supposed to complement other Web filtering approaches, such as URL filtering, not to replace them. Some situations, such as SSL connections and content of images, video, Flash objects or Java applets, are non-trivial to analyze on line. This algorithm can work with other approaches to filter the Web content with high efficiency.

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.

5 Conclusions

This work addresses the problem of possibly long delay from text classification algorithms to perform run-time content analysis of Web content. An early decision algorithm to decide to either block or pass the content as soon as the decision can be made is presented. A significant performance improvement is observed. The throughput is increased by about five times higher for banned content and nearly four times higher for allowable content while the accuracy remains fairly good. In the F1 measure, the accuracy can achieve about 89% for filtering banned content, and about 93% for allowable 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 also 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 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 list is also facilitated.

Acknowledgement

This work was supported in part by the Taiwan Information Security Center (TWISC), National Science Council under the Grants NSC 94-3114-P-001-001-Y and NSC 94-3114-P-011-001.

References

1. Internet Filter Review 2005. Available at http://internet-filter-review.toptenreviews.com/
2. Y. Yang and X. Liu, “A re-examination of text categorization methods”, Proc. of SIGIR’99, 22nd ACM International Conference on Research and Development in Information Retrieval (1999) 42-49
3. F. Sebastiani, “Machine learning in automated text categorization,” ACM Computing Survey, vol. 34, No. 1 March (2002) 1-47
4. Tom Mitchell. Machine Learning, McGraw Hill (1996)
5. The Rainbow library. Available at http://www-2.cs.cmu.edu/~mccallum/bow/rainbow/
6. C.J.Rijsbergen. Information Retrieval, Butterworths, London (1979)
7. YAHOO directory services. Available at http://www.yahoo.com
8. S. Wu and U. Manber, “A fast algorithm for multi-pattern searching”, Technical Report TR-94-17, University of Arizona (1994)
9. M. E. Lesk, “Lex – A lexical analyzer generator”, Comp. Sci. Tech. Rep. No. 39. Bell Laboratories (1975)
10. Aho, A. V., and M. J. Corasick, “Efficient string matching: an aid to bibliographic search” Comm. of the ACM, 18 (1975) 333-340
11. The DansGuardian Web filtering tools. Available at http://dansguardian.org/