

# Discovering Informative Content Blocks from Web Documents

Shian-Hua Lin and Jan-Ming Ho  
Institute of Information Science, Academia Sinica  
128 Academia Road Sec. 2  
Nankang, Taipei 115, Taiwan  
E-Mail: {shlin, hoho}@iis.sinica.edu.tw

## ABSTRACT

In this paper, we propose a new approach to discover informative contents from a set of tabular documents (or web pages) of a web site. Our system, InfoDiscoverer, first partitions a page into several content blocks according to HTML tag <TABLE> in a web page. Based on statistics on the occurrence of the features (terms) in the set of pages, it calculates entropy value of each feature. According to the entropy value of each feature in a content block, the entropy value of the block is defined. By analyzing the information measure, we propose a method to dynamically select the threshold of entropy that partitions blocks into either informative or redundant. Informative content blocks are distinguished parts of the page, whereas redundant content blocks are common parts. Based on the answer set generated from 13 manually tagged news web sites with a total of 26518 web pages, experiments show that both recall and precision rates are greater than 0.95. That is, using the approach, informative blocks (news articles) of these sites can be automatically separated from semantically redundant contents such as advertisements, banners, navigation panels, news categories, etc. Note that the size of informative blocks of a page is much smaller than that of the page. Thus, by adopting the proposed method in information retrieval and extraction applications, it not only increases the retrieval and extracting precision but also reduces the indexing size and extracting complexity.

## Keywords

Informative Content Discovery, Entropy, Information Retrieval, Information Extraction, Web Mining

## 1. INTRODUCTION

The innovation of the Web creates numerous information sources published as HTML pages on the Internet. Based on the report of Cyveillance [14], the size of Internet is 2.1 billion unique pages at 2000. The study also found that the Internet is growing at an explosive rate of more than 7 million pages each day, indicating that it will double in size by early 2001. However, there are many redundant pages on the Web, such as mirror sites or identical pages with different URL. Also, much information is *intra-page redundancy*. For instance, almost all dot-com web sites present their service channels, navigation panels, copyright and privacy announcements, and advertisements in every page for business purpose, easy access and user-friendly. In this paper, we focus on the problem of intra-page redundancy instead of the Internet page redundancy. We propose methods to automatically discover the intra-page redundancy and extract informative contents of a page.

What is intra-page redundancy? We depict it with the example of CNET Tech News<sup>1</sup>. The presentation of each news page begins with CNET's tech sites, the category information, advertisements, a search box, the news content, latest headings, related news, feature services, the copyright, etc. Regarding these content parts as content blocks, all blocks, except for the "news content" block, are identical in each news page. In this paper, we call these identical blocks as *redundant content blocks*. Only "news content" block is distinguishable and semantically meaningful for users, we call it *informative content blocks*. Most sites, especially for dot-com sites, apply the same presentation style for the business purpose. It is convenient for users to easily navigate their related services by one simple click in any pages. However, it's a big challenge for search engines or web miners since these systems are not as clever as human so that they need to process the whole content of a page. For example, searching "game hardware tech jobs" in Google obtains 28 records (including the top 1 result) from CNET in the first 100 results<sup>2</sup>. However, there are no results of "game hardware tech jobs" in CNET and its Job Seeker ([http://dice.cnet.com/seeker.epl?rel\\_code=1&op=1](http://dice.cnet.com/seeker.epl?rel_code=1&op=1)).

Since search engines always index the whole text of each web page, the information such as "CNET Tech Job" appears in every page of CNet. I.e. the information is useless for processing, indexing, and extracting. The problem was found in our past work that we carried out a news search engine (NSE) for news web sites in Taiwan<sup>3</sup>. Since these news sites publish pages of news articles with many redundant blocks, we applied hand-coding approach to our news search engine to deal with the problem of intra-page redundancy and provide a more precise search result. NSE merely reads artificially tagged page contents to avoid indexing redundant contents. Unfortunately, the hand-coding approach is a tedious work so that it is not a scalable method to be applied to index and search all news pages on the Internet.

Obviously, the problem of intra-page redundancy affects two factors widely used to evaluate search engines: the precision of search and the size of index. In the previous example, the current best and most popular search engine, Google, retrieves at least 28 non-relevant results in the top-one-hundred results. As for the size of index, the size of informative content blocks is much smaller than the page size. For the example of NSE, regarding manually tagged contents as informative blocks, the size is about 12.14% of the page size. The presentation of search result is also influenced by the problem. Most search engines automatically capture first

<sup>1</sup> We get it from <http://news.com.com/> at February 20, 2002.

<sup>2</sup> It was obtained from Google at February 20, 2002.

<sup>3</sup> News Search Engine (NSE): <http://nse.yam.com/>.

several sentences as the description of a page. If redundant blocks, such as navigation panels or advertisement banners, are located in the beginning of pages, descriptions of all pages in the same site will be identical.

Fortunately, there are many web sites using `<TABLE>` as a template to layout their pages, especially for dot-com sites. Based on the statistics of our search engine for all pages in Taiwan, there are 49.69% dot-com pages<sup>4</sup>, in which 73.96% are tabular pages with 4.42 `<TABLE>` tags in average. As for pages of non-dot-com sites, 57.32% are tabular pages with 3 `<TABLE>` tags per page. That is 69.59% pages are tabular structures in our search engine. Some pages even contain tens of tables. Intuitively, `<TABLE>` is easy and convenient to modularize an HTML page to several visualized content blocks. For this sake, it is also easy to be applied to identify the content block, which is the unit of content to be justified as redundant or informative in this paper.

Many studies on information extraction (or web mining) also try to discover metadata from a set of web documents [12][17]. However, they perform well only in specific sites based on the guidance of human knowledge. That is these applications are not scalable in par with search engines. In the paper, we try to deal with the semantic and scalability problem with respect to search engines and information extraction systems. Hence, we focus on efficiently and automatically discovering informative content blocks instead of extracting the metadata of a page. In this way, our system can be directly and effectively applied to search engines. Furthermore, it can also be a pre-process of information extraction since focusing on informative blocks rather than the whole page will reduce the complexity and increase the mining precision. Without human knowledge, our system performs as well as hand-coding results of NSE or even better.

In the following section of the paper, we first describe related studies. Then, we illustrate the representation of content blocks in a page, and propose a method to evaluate the information measure of a content block. Based on the information measure, we use the greedy approach to dynamically divide content blocks into either informative or redundant. Regarding the hand-coding data of NSE as the answer set, we perform several experiments to evaluate the effectiveness of our proposed method. Experiments indicate our method is perfect to discover informative content blocks from tabular pages. Finally, we conclude our contributions. The paper proposes a new topic that has many open studies. Thus, we also describe several interesting future works.

## 2. RELATED WORK

This study is motivated from the problem of intra-page redundancy that causes search engines to index redundant contents and retrieve non-relevant results. The problem also affects web miners since they extract patterns from the whole document rather than the informative content. In this section, we illustrate studies on both fields. For better understanding, in the following of the paper, we use information retrieval (IR) systems to denote search engines and information extraction (IE) systems to denote web or text miners.

Many IR systems have been implemented to automatically gather, process, index, and analyze the Web documents for serving users information needs. IR systems (or search engines) can be divided into three automatic processes: preprocessing (crawling), indexing, and searching. In the crawling phase, the web crawler grabs a page and its related pages by following hyperlinks of the page. It also parses contents of the page based on HTML or other markup language like XML. Then, the index engine processes and stores the parsed content as the page's index files or database indexes, which makes the following searching requirements to be efficiently matched with indexed documents and retrieved in the relevant results. However, it is hard to rank the order or relevant results due to the fast growth of the Web documents. By analyzing the hyperlink structure of the Web, two best-known algorithms, HITS [15] and PageRank [3], were proposed to cope with the problem. PageRank is successfully used in Google search engine [4]. As the appendix described in [3], the ranking result will be inherently biased toward to advertising pages and away from the needs of users since all search engines index the whole page content without considering the semantics of content. In our past studies [13], HITS algorithm does not give a concise web structure due to many semantically redundant hyperlinks in pages. Obviously, redundant contents, such as advertisements, company logos, navigation panels, relative channels, and privacy statements, are indexed so that they are probably retrieved. Consequently, IR systems are scalable applications, but they require automatic processes to find meaningful contents for indexing and improving the precision of retrieval.

IE systems [11] [12] [17] [25] have the goal of transforming a collection of documents, usually with the help of IR systems, into information that is more readily digested and analyzed [9]. In par with IR systems that retrieve relevant documents, IE systems aim to extract the structure or representation of a document. There are basically two types of IE: IE from unstructured texts and IE from semi-structured documents [16]. Traditional IE studies, called text mining, typically integrate with NLP works to extract the information from unstructured text. With the increasing popularity of the Web, traditional IE studies was shifted to the structural IE research called web mining. Wrapper [17] and SoftMealy [12] are well known structural IE systems that extract the structural information from HTML documents based on manually generated templates.

As for the processes of IE systems, Cardie [3] defines five pipelined processes for an IE system: tokenization and tagging, sentence analysis, extraction, merging, and template generation. SRI's FASTUS [1] is based on a cascade of six finite-state transducers, which are similar to that of Cardie. Machine learning is usually applied to learn, generalize, and generate rules in the last three processes. However, the domain-specific knowledge such as concept dictionaries and templates for generating rules are necessary to be manually generated. Training instances applied to learning processes are also artificially selected and labeled. For example, text miners usually learn wrapper rules from labeled training tuples. In Wrapper induction [17], the author manually defines six wrapper classes, which consist of knowledge to extract data by recognizing delimiters to match one or more of the classes. The richer a wrapper class is, the more probable it will work with any new site [7]. SoftMealy [12] provides a GUI that allows a user to open a web site, define the attributes and label the tuples in the web page. The common disadvantages of IE systems are the cost of templates, domain-dependent NLP knowledge, or annotations

---

<sup>4</sup> In <http://www.inktomi.com/webmap/>, Inktomi reveals that the rate of dot-com pages is 54.68%.

of corpora generated by hand. This is why these systems are merely applied to specific web applications, which extract the structural information from pages of specific web sites or pages generated by CGI. Consequently, IE systems are not scalable so that they cannot be applied to resolve the semantic deficit of search engines.

In this paper, we propose a general approach to discover informative contents of a page to cope with the problem of intra-page redundancy in IR systems. Also, IE systems will become more efficient by extracting structures from informative contents instead of the whole page. As shown in Figure 1, our system tries to extract informative contents from HTML documents (or other types of documents) to improve the precision and efficiency of IR and IE systems.

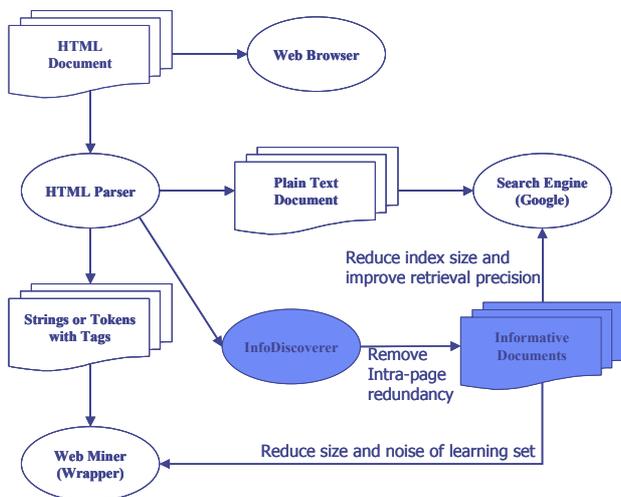


Figure 1: Process HTML documents for diverse requirements.

### 3. PAGE REPRESENTATION AND CONTENT BLOCKS

With the increasingly growth of the Web, documents written in HTML are the majority in the Web, even XML was proposed for several years. W3C's Document Object Model (DOM) [22] defines a tree structure for HTML [23] and XML [24] documents, in which tags are internal nodes of the tree, and texts or hyperlinks to another trees are leaf nodes. As the statistics described in Introduction, there are about 70% web pages using HTML tag `<TABLE>`. To reduce the complexity, we concentrate on HTML documents with `<TABLE>` tags. That is the tabular document (page) defined in the paper.

Based on HTML tag `<TABLE>`, a page is partitioned into several content blocks. Since the table-structure can be nested, the partitioned content blocks form a tree to denote a page. As shown in Figure 2, the rectangle with dashed line denotes a page, and other rectangles are content blocks. Applying the following rules, a page is represented by a tree with content blocks.

- Top-and-down and left-and-right ordering corresponds to the left-and-right ordering in the tree.

- Nested content blocks correspond to nodes in the lower level of the tree.

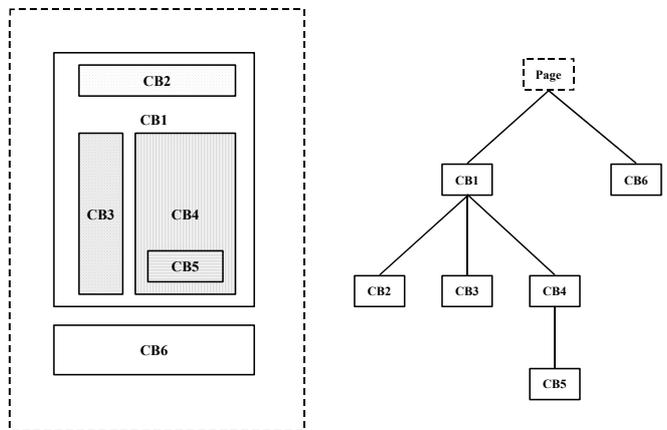


Figure 2. Using the tree structure of content blocks to denote the content of a page.

Obviously, `<TABLE>` is not the only way to partition a page into blocks. If a content block includes too many texts, based on the specification of W3C DOM, it can be partitioned into several smaller blocks according to tags, such as `<TR>` and `<TD>` embedded in `<TABLE>`.

Besides tabular tags, we also consider the content enclosed by HTML tags `<TITLE>` and `</TITLE>` as a special content block since there are many sites assign it pages with the same title, such as the company's name or the default name generated by document editors. If the title-content block is redundant, the title can be replaced by the first sentence of its informative content block. In the example of news page, the first sentence of informative content block is always the news title.

In the following section, we propose methods to estimate the entropy value of a content block, which is used to determine the block's property: informative or redundant.

### 4. APPLYING ENTROPY TO DISCOVERY INFORMATIVE CONTENT BLOCKS

In the paper, we assume that a web site usually employs one or several templates to present its web pages. This is especially true if the web site publishes its pages generated by CGI programs. A *page cluster* is a set of pages that are presented by the same template. In [13], we propose another methods to mine the informative structure of a web site. The structure consists of several TOC pages with links to a set of ARTICLE pages. These ARTICLE pages are categorized into several page clusters. Each cluster is derived from one TOC page. For the example of a news web site, a TOC page probably corresponds to one category with a cluster of ARTICLE pages in the category. InfoDiscoverer is applied to discover informative content blocks from a page cluster. If all pages of a web site use the same template, the web site is regarded as one page cluster. In the following of the paper, we assume that a web site is a page cluster without losing the generality. Hence, the process of discovering informative content block is started after the crawling phase of a IR system completely

grabbed all pages of the site. The crawling sequence can be either breadth first search (BFS) or depth first search (DFS), but it is restricted in the same site. I.e. the training set of InfoDiscoverer only includes pages of the same site. Figure 3 shows modules of InfoDiscoverer. Each module will be described in the following sub-sections.

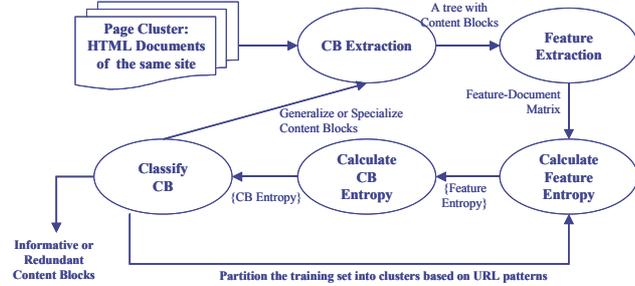


Figure 3. The processes of InfoDiscoverer.

### 4.1 Extracting Content Blocks from a Page

As described in the previous, based on DOM, a web page can be parsed and represented with a tree structure, in which leaf nodes contains content or anchor texts. The process of extracting content blocks is categorized into two phases. In the initial phase, a coarse tree structure is obtained by restrict the parsing of a page based on HTML tag <TABLE>. Each internal node indicates a content block that consists of one or more content strings as its leaf nodes. The nested table structures form the hierarchical tree, i.e., some child blocks can be embedded in a parent block. Obviously, content strings of child blocks are excluded from the parent block. For the example shown in Figure 4, each rectangle denotes a table with embedded tables and content strings. Embedded tables are regarded as child content blocks. That is CB2, CB3, BB4, and CB5 contain content strings S1, S3, S4, and S6 respectively. The parent block CB1 contains strings S1 and S5.

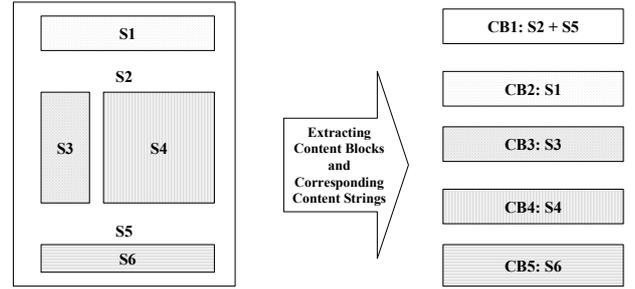


Figure 4. Extracting content blocks and corresponding content strings.

The complexity of the initial parsing process is one-scan pass. That is  $O(n)$ , where  $n$  is the length of the page's HTML source. The process can be done while the crawler parses a grabbed page for furthermore crawling, caching, and indexing. Thus, the extracting process has no extra burden on IR systems.

The second phase of extracting process is to refine the granularity of the tree while the classification of content blocks is ambiguous.

### 4.2 Extracting Features of Content Blocks

After parsing a page into content blocks, features of each block are simultaneously extracted. In this paper, a feature is corresponding to a meaningful keyword except stop words. Applying the Porter stemming algorithm [18] and removing stop words in the stop-list, English keywords (features) can be extracted [19]. Extracting keyword features written in oriental languages seems harder because of no trivial separators specified in these languages. However, many studies have applied statistical approaches to extracting keywords of oriental languages [8]. In our lab, we developed an algorithm to extract keywords from Chinese sentences based on a Chinese term base. The term base is generated via collecting hot queries, excluding stop words, from our search engine<sup>5</sup>. The complexity of extracting Chinese features is  $O(m \log m)$ , where  $m$  is the length of the Chinese sentence. Thus, the complexity is  $O(n \log n)$ , where  $n$  is the average page length. The accumulated complexity is  $O(|D| \times n \log n)$ , where  $|D|$  is the number of documents in the cluster.

### 4.3 Calculating Entropy Values of Features

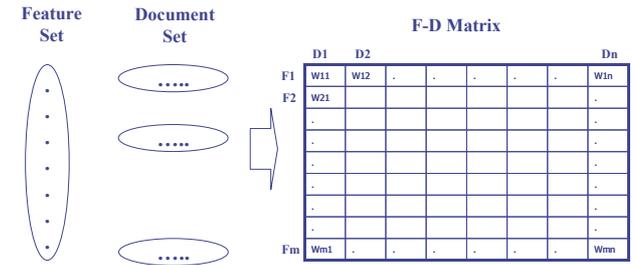


Figure 5 Feature-Document Matrix for calculating entropy values of feature.

The entropy value of a feature is estimated from the weight distribution of features appearing in a page cluster. To easily calculate each feature's entropy, features of content blocks in a page can be grouped and represented as a feature-document list with term frequency (TF) or weight (such as TFxIDF [19] or its variations [21]) stored in the entry of the list. Considering all pages in a cluster or a web site, feature-document lists of pages form the feature-document matrix (F-D Matrix) as shown in Figure 5. The matrix can be generated while extracting features of each document. The time complexity is  $O(|D| \times |F| \log |F|)$  for generating the matrix, where  $|F|$  is the average number of features and  $|F| \log |F|$  is the cost of sorting features in a page. We can regard that the number of features of page is proportional to the page length. Thus, the complexity is  $O(|D| \times n \log n)$ . The complexity for calculating entropy values of all features is linear to the total number of features. I.e. the accumulated complexity is still  $O(|D| \times n \log n)$  up to now. Most importantly, the result of

<sup>5</sup> The searching service is a project sponsored by Yam (<http://yam.com/>). It served the Web users from November, 1998 to December, 2000.

feature-document matrix is reusable since the matrix is corresponding to parsing terms and measuring weight of the indexing process in IR systems.

Based on F-D Matrix, measuring the entropy value of a feature is corresponding to calculating the probability distribution in a row of the matrix. The following is Shannon's famous general formula for uncertainty [20]:

$$0 \leq H = -\sum_{i=1}^n p_i \log_2 p_i \leq \log_2 n, \text{ where } p_i \text{ is the probability of event } i,$$

By normalizing the weight of feature to be [0, 1], the feature entropy is:

$$H(F_i) = -\sum_{j=1}^n w_{ij} \log_2 w_{ij}, \text{ where } w_{ij} \text{ is the weight of } F_i \text{ in document } D_j$$

To normalize the entropy value to the range [0, 1], the base of logarithm is the number of documents, and the above equation is modified as:

$$0 \leq H(F_i) = -\sum_{j=1}^n w_{ij} \log_d w_{ij} \leq 1, \text{ where } d \text{ is the number of documents } (d = |D|)$$

For the example of Figure 6, there are N pages with five content blocks in each one. Features F1 to F10 appear in one or more pages according to the figure. The layout is widely used in the dot-com web site that consists of the logo of company on the top, followed by advertisement banners or texts, navigation panels on the left, informative content on the right, and its copyright policy on the bottom. Without losing the generality, we just consider two pages in this figure and the feature entropy is calculated as the following.

$$H(F_1) = -\sum_{j=1}^2 \frac{1}{2} \log_2 \frac{1}{2} = H(F_2) = H(F_3) = H(F_4) = H(F_5) = H(F_6) = 1$$

$$H(F_7) = -1 \log_2 1 - 0 \log_2 0 = H(F_8) = H(F_9) = H(F_{10}) = 0$$

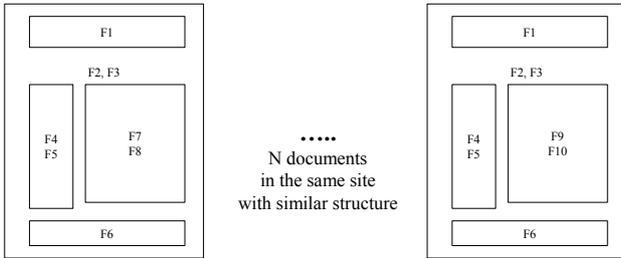


Figure 6. Measuring the entropy value of a feature based on its distribution in documents.

#### 4.4 Estimating Entropy Values of Content Blocks

By instinct, feature entropies contribute to the semantic measure of a content block that owns these features. I.e. the entropy value of a content block is the summation of its features entropies as the following equation.

$$H(CB_i) = \sum_{j=1}^k H(F_j), \text{ where } F_j \text{ is a feature of } CB_i \text{ with } k \text{ features}$$

Since content blocks contain different number of features, the equation is normalized as:

$$H(CB_i) = \frac{\sum_{j=1}^k H(F_j)}{k}$$

That is the entropy of a content block, H(CB), is the average of all feature entropies in the block.

It is feasible to assume that the average number of content blocks in a page is constant. Undoubtedly, the time complexity is  $O(|D|)$  for calculating the entropy value of each content block. The accumulated complexity is still  $O(|D| \times n \log n)$  up to now.

#### 4.5 Classifying Content Blocks

Based on H(CB), the content block can be divided into two categories: redundant and informative.

- If H(CB) is higher than a defined threshold or close to 1, the content block is absolutely redundant since most of the block's features appear in every page.
- If H(CB) is less than a defined threshold, the content block is informative for the reason that features of the page block are distinguishable from others. I.e. these features appearing in the page's block seldom come out from other pages in the page cluster.

The threshold is not easy to determine since it would be variant for different clusters or sites. If the higher threshold is chosen, the higher recall rate is expected. However, the precision rate may become lower. To get a balanced recall-precision rate, we apply the greedy approach to dynamically determine the threshold for different training sets (page clusters or sites). If the threshold is increased, more informative features (in informative content blocks) will also be included. The basic idea of the greedy approach is:

*By considering the included features from selected content blocks, the increasing of threshold value will include more features from more blocks. If the increase of threshold never increases more features, the boundary between informative and redundant blocks is reached.*

For easily understanding, we will explain the greedy approach based on real experiments in the following section. The time complexity depends how the interval of the increasing of threshold. In the following experiment, we use the interval started from 0.1 to 0.9 with step 0.1. Therefore, we can conclude that the time complexity of InfoDiscoverer is  $O(|D| \times n \log n)$ .

### 5. EXPERIMENTS AND EVALUATION

In our past works, we manually labeled a set of tags for identifying informative content blocks of pages in several news web sites in Taiwan<sup>6</sup>. Regarding these data as the answer set of informative blocks, InfoDiscoverer automatically discovers informative blocks of pages from these sites and compare results with the answer set. We apply two measures widely used in evaluating the performance of IR systems, recall and precision rate, to the following experiments and verify the quality of our proposed methods. Regarding features extracted from

<sup>6</sup> It is a project sponsored by Yam. The search service for Taiwan news web sites is running at <http://nse.yam.com/>.

hand-coding informative content blocks as desired features, measures of recall and precision are shown in Figure 7. Clearly, The ideal case is that both recall and precision rate equals to 1.

To evaluate our methods, we choose 13 new sites that present their pages with <TABLE> tag. Since news articles of different categories may have different presentation style, we choose one category from each site as shown in Table 1. That is each site indicates one page cluster, which indicates a training set applied to InfoDiscoverer. In fact, we do not need to run InfoDiscoverer for all pages in the training set since some sites may contain thousands of pages. Thus, ten training pages are randomly selected from each cluster in the first experiment.

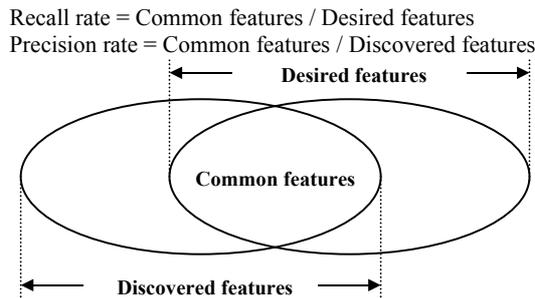


Figure 7: Recall and precision rates.

Table 1. News sites with tabular pages.

Site	Site+Path	Category	Pages
IHome	<a href="http://www.ithome.com.tw/News/Investment/">http://www.ithome.com.tw/News/Investment/</a>	Network Investment	202
ET	<a href="http://www.ettoday.com.tw/life/">http://www.ettoday.com.tw/life/</a>	Life	159
FTV	<a href="http://www.ftv.com.tw/">http://www.ftv.com.tw/</a>	Taiwan News	794
CNet	<a href="http://taiwan.cnet.com.tw/investor/news/">http://taiwan.cnet.com.tw/investor/news/</a>	Investment	499
TSS	<a href="http://www.tssdnews.com.tw/cgi-bin/news_sub/">http://www.tssdnews.com.tw/cgi-bin/news_sub/</a>	Supplement	123
CDN	<a href="http://www.cdn.com.tw/daily/">http://www.cdn.com.tw/daily/</a>	Miscellaneous News	1305
TVBS	<a href="http://www.tvbs.com.tw/code/tvbsnews/daily/">http://www.tvbs.com.tw/code/tvbsnews/daily/</a>	Daily News	9943
CTV	<a href="http://www.chinatv.com.tw/">http://www.chinatv.com.tw/</a>	Taiwan News	3597
CAN	<a href="http://www.cna.com.tw/cgi-bin/readcipt77.cgi?a1&amp;0">http://www.cna.com.tw/cgi-bin/readcipt77.cgi?a1&amp;0</a>	Headlines	5096
UDN	<a href="http://udnnews.com/FLASH/">http://udnnews.com/FLASH/</a>	Stock and Financial	1127
CTimes	<a href="http://news.chinatimes.com.tw/news/papers/online/">http://news.chinatimes.com.tw/news/papers/online/</a>	Society	643
CTS	<a href="http://www.cts.com.tw/news/headlines/">http://www.cts.com.tw/news/headlines/</a>	International	1064
TTimes	<a href="http://www.ttimes.com.tw/">http://www.ttimes.com.tw/</a>	City	1966

TTimes was closed at February 21, 2001.

To find the optimal threshold of H(CB) for each cluster, recall and precision are measured according to increasing H(CB) from 0.1 to 0.9 with the interval 0.1. In par with the hand-coding data, the recall rate of each site and corresponding H(CB) intervals is shown in Figure 8. X-axis is the increasing of threshold for H(CB) and Y-axis is recall rate based on the answer set. The recall rate is equivalent to the number of features included in selected blocks due to the increasing of H(BC). Thus, the optimal threshold is found while the number of features (recall rate) is not increased with the increasing of H(CB). For the example of Figure 8, the optimal threshold of ET and CTV is 0.2 since the recall rate (1.0) is never increased with the interval. The result shows that all sites, except for UDN, have very high recall rates (at least 0.95). The recall of UDN, 0.759, is not perfect. We trace to our training pages and hand-coding data. We found that the hand-coding data

of UDN is wrong because of including the title information of news categories. Those optimal thresholds of sites are distributed from 0.2 to 0.7. That is optimal thresholds are variant among different sites. Consequently, based on the increasing of the greedy approach is able to find the optimal threshold of H(CB) dynamically.

Of course, the precision has to be verified at the optimal threshold. The precision rate is shown in Figure 9. In par with recall rate shown in Figure 8, the highest precision rate of each page cluster is almost reached at each corresponding optimal threshold, except for TTimes. The precision of TTimes is 0.53 at the optimal threshold of H(CB) 0.7, but the highest precision value is 0.65 at 0.5 according to the answer set. We check news articles of TTimes and find that each page includes an extra content block consisting of “anchors of related news”, which are news pages related to the current article. Since the block consists of too many anchors, the total text length of the block is even longer than the article length in many pages. If contents of “related news” are different among training pages, their corresponding H(CB) will be lower enough and become an informative content block. Thus, this kind of block is also included. These included noisy features affect the decision of the threshold. However, the judgment of informative or redundant is ambiguous since it depends on the perspective of users.

Based on the optimal H(CB) threshold found by our greedy approach, recall and precision of each cluster are shown in Figure 10. In most cases, recall and precision are larger than 0.95, except for the precision of TTimes and the recall of UDN. As we described, the low precision of TTimes is due to the content block “related news” and the low recall of UDN is for the sake of wrong hand-coding data. Thus, our approach almost achieves a perfect recall and precision in discovering informative contents from tabular HTML pages.

To investigate the effect of the number of randomly selected training examples, we redo the same experiments on all page clusters except for UDN and TTimes for their corresponding wrong hand-coding data and semantically ambiguous content blocks of related news. The number of training examples is started from 5 to 40 with interval 5. The result is shown in Figure 11, in which the dashed line denotes the recall rate (started with “R”) and the solid line represents the precision (started with “P”). Most of clusters have perfect recall and precision that approach to 1 (many lines are overlapped in recall/precision 1.0), but few clusters are not while the number of randomly selected examples is increased. It reveals that the number of examples may influence on the precision rate since the precision of three sites (CTS, ET, and CTimes) is degraded below 0.9 while the number is increased. And the random number almost has no effect on the recall rate since no clusters, except for CNet, have recall rates less than 0.95. Intuitively, if contents of a cluster are similar, the more examples involved, the higher entropy threshold would be selected for filtering informative content blocks. Consequently, more training examples do not imply the higher precision. However, the recall rate is not affected because higher threshold means more features included.

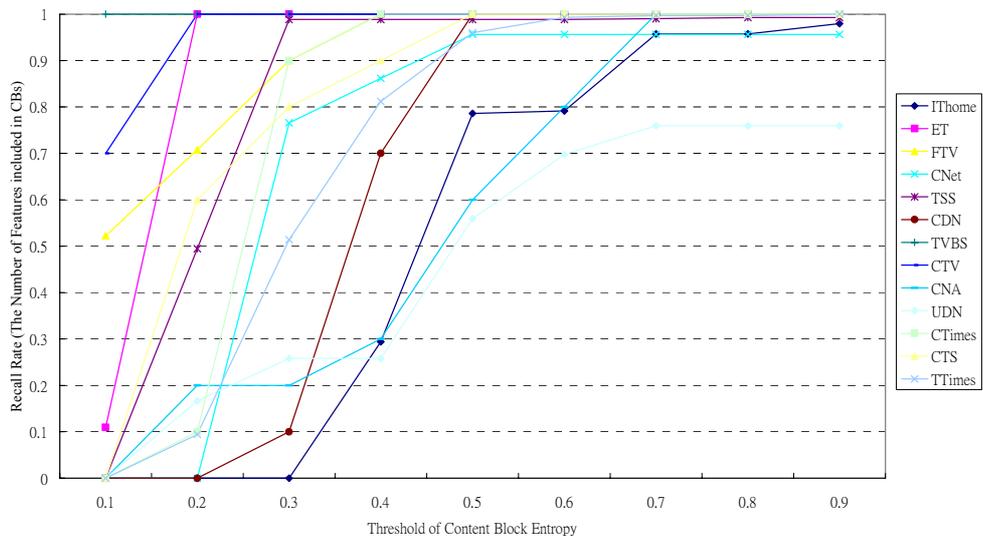


Figure 8. Recall rate of each page cluster.

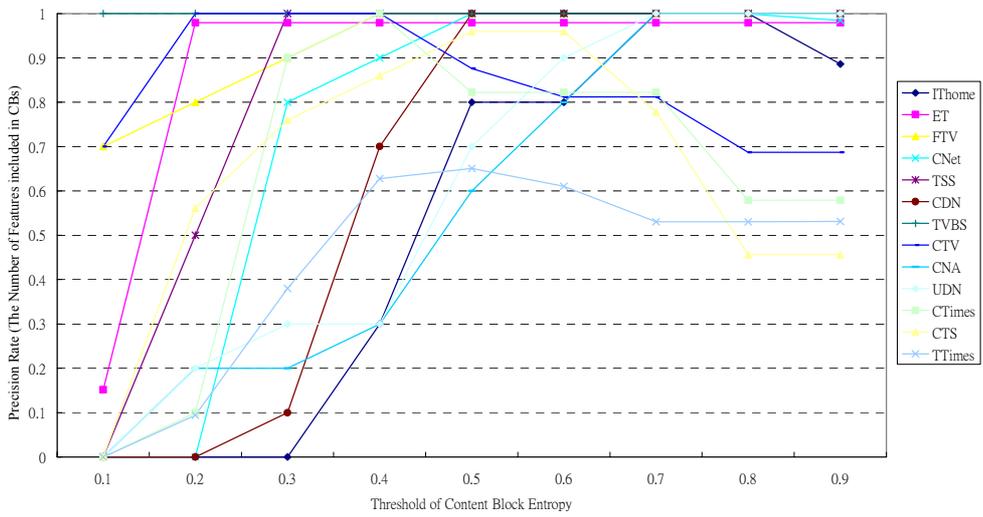


Figure 9. Precision rate of each page cluster.

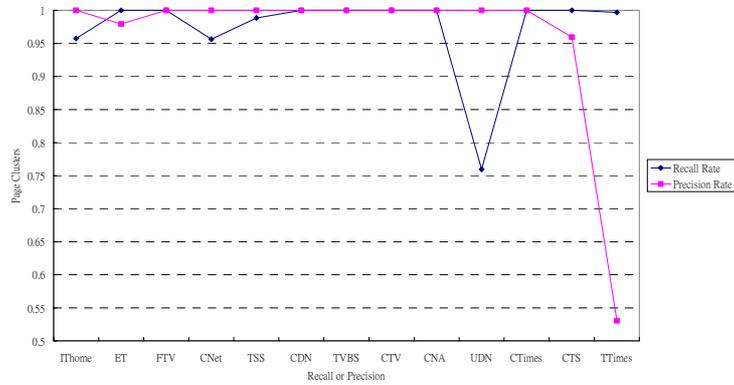


Figure 10. Recall and precision rates of each site based on the dynamic threshold.

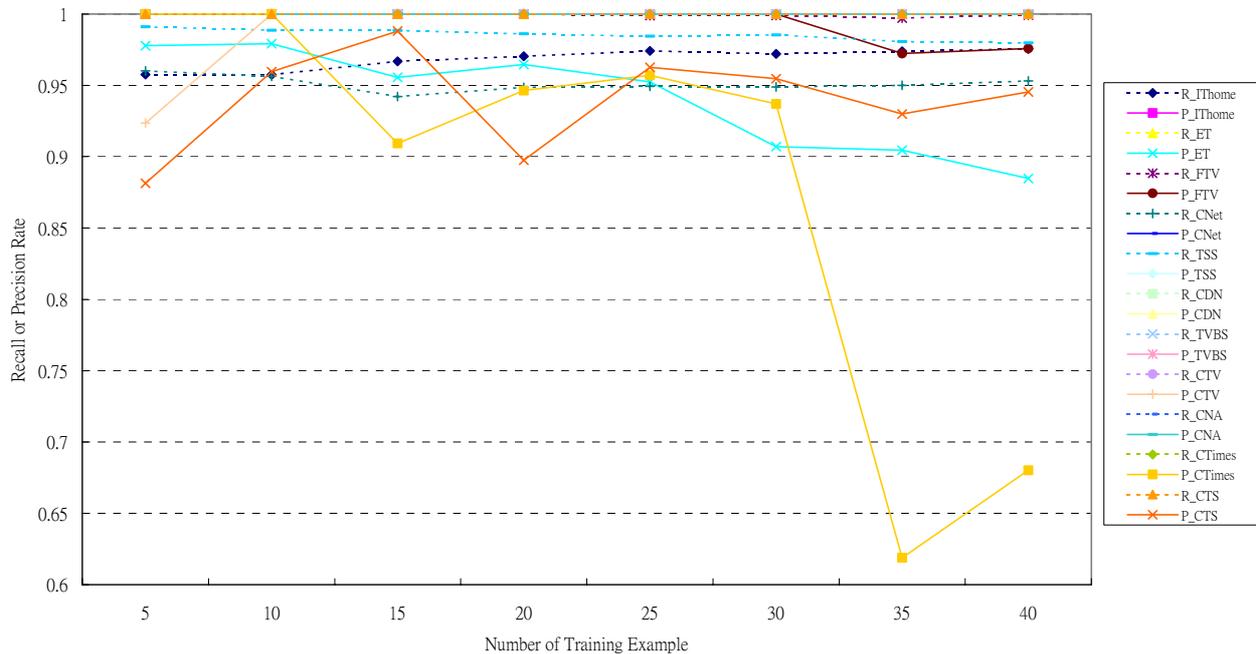


Figure 11. Recall and precision based on the number of training examples.

## 6. CONCLUSION AND FUTURE WORK

According to previous experiments, we can conclude that our proposed methods are feasible to discover informative contents from web pages. The greedy approach of InfoDiscoverer is adaptive to different web sites with different templates. Based on the approach, the optimal threshold of informative content blocks is dynamically selected for different sites. The result shows that recall and precision rates are larger than 0.95, which is very close to the hand-coding result. In the following, we conclude the contributions of InfoDiscoverer to web IR and IE systems.

### 6.1 Contributions to IR Systems

Obviously, the experiment results prove contributions to our news search engine since InfoDiscoverer knows how to automatically extract informative contents, i.e. news articles, from news web pages. Evidently, it can be applied to general search engines by reducing the size of index and increasing the precision of retrieval. The complexity of the discovering process is polynomial. Most importantly, most results generated by processes of InfoDiscoverer are shared with the crawler and indexer of web IR systems. For example, extracted features and corresponding weights appearing in informative content blocks can be directly indexed by web IR systems. Thus, its low time complexity and fully automatic property are feasibly used to integrate with web IR systems.

### 6.2 Contributions to IE Systems

Applying InfoDiscoverer to IE systems is also efficient as IE systems simply consider smaller informative content blocks instead of the whole page content. Fields of a page may be deduced from the entropy-curve based on the feature sequence in informative blocks. For example the web site, IMDB (<http://www.imdb.com/>), contains movie pages presented with a fixed format that contains metadata “Directed by”, “Writing credits”, “Genre”, “Plot Outline”, “User Comments”, “Cast overview”, etc. Entropy values of these features are highest owing to uniformly supporting to all movie pages. That is features with the highest entropies,  $H(CB) = 1.0$ , are fields for structuring these pages.

In addition to these potential applications for IR and IE systems, we will intensely investigate the influence of generalization and specialization to InfoDiscoverer. W3C DOM provides the guideline to generalize and specialize the DOM tree, which correspond to merge and split content blocks generated by InfoDiscoverer. For example, the entropy of a large content block B is probably moderate so that the system cannot determine it is classification (redundant or informative). By reducing the granularity of the block based on DOM, the block B is possibly specialized as several distinguishable blocks that are easily classified as redundant or informative. Several content blocks with the same classification can also be generalized as a larger block to make the tree concise.

## 7. REFERENCES

- [1] Bear, J., Israel D., Petit, J., and Martin, D., "Using Information Extraction to Improve Document Retrieval," the Sixth Text Retrieval Conference (TREC 6), 1997, pp. 367-378.
- [2] Blahut, R. E., "Principles and Practice of Information Theory," Addison Wesley, 1987.
- [3] Brin, S. and Page, L., "The Anatomy of a Large-Scale Hypertextual Web Search Engine," the Seventh International World Wide Web Conference, 1998.
- [4] Brin, S. and Page, L., Google Search Engine, <http://www.google.com/>.
- [5] Cardie, C., "Empirical Methods in Information Extraction," AI Magazine, 18(4):5-79, 1997.
- [6] Chakrabarti, S., "Integrating the Document Object Model with Hyperlinks for Enhanced Topic Distillation and Information Extraction," the Tenth International World Wide Web Conference (WWW10), 2001, <http://www10.org/cdrom/papers/489/>.
- [7] Chidlovskii, B., "Wrapper Generation by k-Reversible Grammar Induction," Workshop on Machine Learning for Information Extraction, August, 2000.
- [8] Chien, L. F., "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval", Proceedings of the ACM SIGIR International Conference on Information Retrieval, 1997.
- [9] Cowie, J. and Lehnert, W., "Information Extraction," Communications of the ACM, 39(1):80-91, 1996.
- [10] Frakes, W. B. and Baeza-Yates, R., "Information Retrieval – Data Structures & Algorithms," Prentice Hall, 1992.
- [11] Freitag, D., "Machine Learning for Information Extraction," Ph.D. Dissertation of Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1998.
- [12] Hsu, C. N. and Dung, M. T., "Generating Finite-state Transducers for Semi-structured Data Extraction from the Web," Information Systems, 23(8):521-538, 1998.
- [13] Kao, H. Y., Lin, S. H., Ho, J. M., and Chen, M. S., "Exploiting Hyperlink Analysis to Mine Informative Structures of Web Sites," submitted to ACM SIGKDD Conference, 2002.
- [14] Kennon, J. and Johnson, A., "Internet Exceeds 2 Billion Pages," <http://www.cyveillance.com/us/newsroom/pressr/000710.asp>, July 10, 2000.
- [15] Kleinberg, J. M., "Authoritative Sources in a Hyperlinked Environment," Journal of the ACM, 46(5):604-632, 1999.
- [16] Kosala R. and Blockeel, H., "Web Mining Research: A Survey," SIGKDD Explorations, 2(1):1-15, 2000.
- [17] Kushmerick, N., "Wrapper Induction for Information Extraction," Ph.D. Dissertation, Department of Computer Science and Engineering, University of Washington, 1997.
- [18] Porter, M., "The Porter Stemming Algorithm," <http://www.tartarus.org/~martin/PorterStemmer/>.
- [19] Salton, G., "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer", Addison Wesley, 1989.
- [20] Shannon, C., "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27, pp. 379-423 and 623-656, July and October, 1948.
- [21] Shasha, D. and Wang, T., "New Techniques for Best-Match Retrieval," ACM Transactions on Office Information System, 8(2):140-158, 1990.
- [22] W3C DOM, "Document Object Model (DOM)," <http://www.w3.org/DOM/>.
- [23] W3C HTML, "HyperText Markup Language," <http://www.w3.org/MarkUp/>.
- [24] W3C XML, "Extensible Markup Language," <http://www.w3.org/XML/>.
- [25] Wang, K. and Liu, H. Q., "Discovering Structural Association of Semistructured Data," IEEE Transactions on Knowledge and Data Engineering, 12(3):353-371, 2000.