Discovering Informative Content Blocks from Web Documents

Shian-Hua Lin
Institute of Information Science
Academia Sinica
128 Academia Road Sec. 2
Nankang, Taipei 115, Taiwan
shlin@iis.sinica.edu.tw

Jan-Ming Ho
Institute of Information Science
Academia Sinica
128 Academia Road Sec. 2
Nankang, Taipei 115, Taiwan
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 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 entropy-threshold 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 26,518 Web pages, experiments show that both recall and precision rates are greater than 0.956. 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. By adopting InfoDiscoverer as the preprocessor of information retrieval and extraction applications, the retrieval and extracting precision will be increased, and the indexing size and extracting complexity will also be reduced.

Keywords

Informative content discovery, Entropy, Information retrieval, Information extraction.

1. INTRODUCTION

The innovation of the Web creates numerous information sources published as HTML pages on the Internet. 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 purposes with 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[1]. 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, expect for the "news content" block, are identical in all news pages. In this paper, we call these identical blocks *redundant content blocks*. Only "news content" block is distinguishable and semantically meaningful for users and is called *informative content blocks*. Most sites, especially dot-com sites, apply the same presentation style for business purposes. It is convenient for users to easily navigate their related services by one simple click in any page. However, it's a big challenge for search engines or Web miners since these systems are not as clever as humans. Therefore, they need to process the whole content of a page. Since search engines always index the whole text of each Web page, information such as "CNET Tech Job" appears in every page of CNet, i.e., the information is useless for processing, indexing, retrieving and extracting. This problem was experienced in previous work we carried out on a news search engine (NSE) for news Web sites in Taiwan[2]. Since these news sites publish pages of news articles with many redundant blocks, we applied the hand-coding approach to our news search engine to deal with the problem of intra-page redundancy and to provide a more precise search result. NSE merely reads artificially tagged page contents to avoid indexing and retrieving redundant contents. Unfortunately, the hand-coding approach is tedious work and is therefore not a scalable method to process 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. The presentation of search results is also influenced by the problem since most search engines automatically capture first several sentences as the description of a page. Redundant blocks, such as company logos, navigation panels or advertisement banners, are usually located on the top of a page. In this way, descriptions of pages extracted by search engines are 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, 49.69% are dot-com pages[3], of which 73.96%

SIGKDD '02, July 23-26, 2002, Edmonton, Alberta, Canada.

[1] We get it from http://news.com.com/ at February 20, 2002.
[2] News Search Engine (NSE): http://nse.yam.com/.
[3] In http://www.inktomi.com/webmap/, Inktomi reveals that the rate of dot-com pages is 54.68%.

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% of these 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 reason, 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 [11][14]. 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 effectively applied to search engines. 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.

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.

2. RELATED WORK

This study is proposed to deal with 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. Thus, we illustrate studies of both fields. In the rest of the paper, for better understanding, 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 search requirements to be efficiently matched with indexed documents and retrieved in the relevant results. However, it is difficult to rank the order or relevant results due to the fast growth of Web documents. By analyzing the hyperlink structure of the Web, the two best-known algorithms, HITS [12] and PageRank [2], were proposed to cope with the problem. PageRank is successfully used in the Google search engine [3]. As indicated in the appendix described in [2], the ranking result will be inherently biased toward 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 fact, HITS algorithm does not give a concise Web structure, due to the 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 will probably be retrieved. Consequently, IR systems are scalable applications, but they require automatic processes to find meaningful contents for indexing and for improving the precision of retrieval.

IE systems [10] [11] [14] [22] 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 [8]. 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 [13]. The former, called text mining, typically integrate with NLP works to extract the information from unstructured text. With the increasing popularity of the Web, text mining studies were shifted to the structural IE research called Web mining. Wrapper [14] and SoftMealy [11] are well known systems that extract the structural information from Web HTML documents based on manually generated templates or examples.

Cardie [2] 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 [14], 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, the more probable it will work with any new site [6]. SoftMealy [11] 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 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 and therefore cannot be applied to resolve the semantic deficit of search engines.

In this paper, we propose an approach to discover informative contents of pages 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. Our system tries to extract informative contents from Web documents (or other types of documents) to improve the precision and efficiency of IR and IE systems.

3. PAGE REPRESENTATION

Pages written in HTML are the majority in the ever-increasing Web, even though XML was proposed for several years. W3C's Document Object Model (DOM) [19] defines a tree structure for HTML [20] and XML [21] documents, in which tags are internal nodes of the tree, and texts or hyperlinks to other trees are leaf nodes. According to counts referred to in the Introduction, about 70% of all Web pages use 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, partitioned blocks form a tree to denote the page based on the following rules:

- 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.

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 the title, headings, <P>, or <TR> and <TD> embedded in <TABLE>. Besides tabular tags, we also consider the content enclosed by the <TITLE> tag as a special block, since many sites assign the same title to pages, such as the company's name or the default name generated by authoring tools.

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. DISCOVER INFORMATIVE CONTENTS

A Web site usually employs one or several templates to present its Web pages. This is especially true for pages generated by CGI programs. A *page cluster* is a set of pages that are presented by the same template. If all pages of a Web site use the same template, the Web site is regarded as one page cluster. In the rest of the paper, we assume that a Web site is a page cluster without losing generality. The process of discovering informative content block is started after an IR system completely grabbed all pages a Web 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 data set includes pages of the same site. Figure 1 shows modules of InfoDiscoverer. Each module will be described in the following sub-sections.

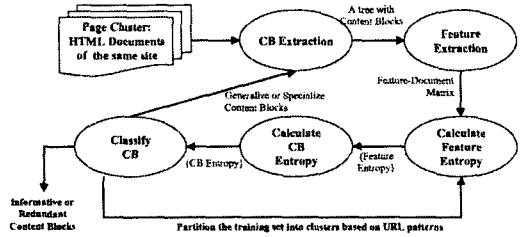

Figure 1: The processes of InfoDiscoverer.

4.1 Extracting Content Blocks from a Page

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 parsing an HTML page based on <TABLE>. Each internal node indicates a content block that consists of one or more content strings (without HTML tags) as its leaf nodes. Due to the nested structure, some child blocks can be embedded in a parent block. Obviously, content strings of child blocks are excluded from the parent block. In the example shown in Figure 2, each rectangle denotes a table with child tables and content strings. Content blocks, CB2, CB3, BB4, and CB5 contain content strings S1, S3, S4, and S6 respectively. The parent block CB1 contains strings S2 and S5. Since the process is scanning the HTML file once, the time complexity is $O(n)$, where n is the file length. The process can be done while the crawler grabs and parses a page for further crawling, caching, and indexing. Thus, the extracting process has no extra burden on IR systems. The second phase is to refine the granularity of the tree while the classification of content blocks is ambiguous. We omit the detail due to the limitation of the paper length.

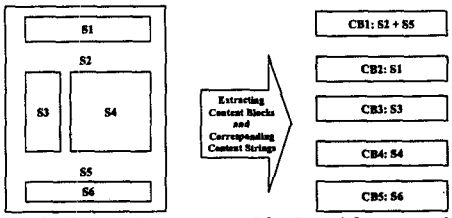

Figure 2: Extracting content blocks with text strings.

4.2 Extracting Features of Content Blocks

After parsing a page into content blocks, features of each block are simultaneously extracted. In this paper, features correspond to meaningful keywords. Stop words are not included. Applying the Porter stemming algorithm [15] and removing stop words in the stop-list, English keywords (features) can be extracted [16]. Extracting keyword features written in oriental languages seems more difficult because of a lack of trivial separators specified in these languages. However, many studies have applied statistical approaches to extracting keywords of oriental languages [7]. In our lab, we developed an algorithm to extract keywords from Chinese sentences based on a Chinese term base. The term base is generated by collecting hot queries (excluding queries with stop words) from our search engine[4]. The complexity of extracting Chinese features is $O(mlogm)$, where m is the length of the Chinese sentence. Thus, the complexity is $O(nlogn)$, where n is the average page length. The accumulated complexity is $O(|D|nlogn)$, where $|D|$ is the number of documents in the cluster.

4.3 Calculating Entropy Values of Features

The entropy value of a feature is estimated according to the weight distribution of features appearing in a page cluster. For easy calculation of 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 [16] or its variations [18]). Considering all pages in a cluster, these lists of pages form the feature-document matrix (F-D Matrix). The F-D matrix can be generated while extracting features of documents in the cluster with the time complexity $O(|D||F|log|F|)$, where $|F|$ is the average number of features and $|F|log|F|$ is the cost of sorting features for efficiently grouping. We can regard the number of features of the page to be proportional to the page length. Thus, the time complexity is $O(|D|nlogn)$. Based on the matrix, the time complexity for calculating entropy values of all features is linear to the total number of features. I.e. the accumulated complexity is still $O(|D|nlogn)$ up to this stage. Most important, the result of the F-D matrix is reusable for the indexing process of IR systems.

Based on the F-D Matrix, measuring the entropy value of a feature corresponds to calculating the probability distribution in a row of

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

the matrix. The following is Shannon's famous general formula for uncertainty [17]:

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

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

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

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:

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

For the example of Figure 3, there are N pages with five content blocks in each page. Features F1 to F10 appear in one or more pages according to the figure. The layout is widely used in dot-com Web sites with the logo of a company on the top, followed by advertisement banners or texts, navigation panels on the left, informative content on the right, and its copyright policy at the bottom. Without losing generality, we consider only two pages in this figure and the feature entropy is calculated as follows.

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

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

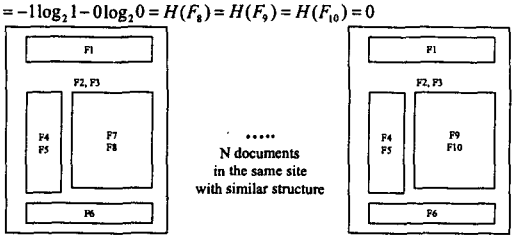

Figure 3: Measuring the entropy value of a feature.

4.4 Estimating Entropy 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 shown in the following equation.

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

Since content blocks contain different numbers 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 $O(|D| \log n)$ up to this stage.

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 because features of the page are distinguishable from others. I.e. these features of the page seldom appear in other pages.

The threshold is not easy to determine since it would vary 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:

Starting the entropy-threshold from 0 to 1.0 with an interval such as 0.1, increasing threshold value will include more features since more blocks are probably included. If the increase of the threshold never includes more features, the boundary between informative and redundant blocks is reached.

For easy reference, we will explain the greedy approach based on real experiments in the following section. The time complexity depends on the interval of increasing of the 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 total time complexity of InfoDiscoverer is $O(|D| \log n)$.

5. EXPERIMENTS AND EVALUATIONS

In our previous works, we manually labeled a set of tags for identifying informative content blocks of pages published by several news Web sites in Taiwan[5]. Regarding this data as the answer set for 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 hand-coding informative content blocks as desired features, measures of recall and precision are shown in Figure 4. Clearly, The ideal case is that both recall and precision rate equals 1.

Recall rate = Common features / Desired features

Precision rate = Common features / Discovered features

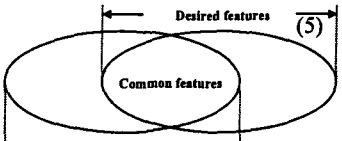

Figure 4: Recall and precision rates.

To evaluate our methods, we choose 13 new sites that present their pages with <TABLE> tags. Since news articles of different categories may be published with different presentation styles, 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.

To find the optimal threshold of H(CB) for each cluster, recall and precision are measured by increasing H(CB) from 0.1 to 0.9 with

[5] It is a project sponsored by Yam. The search service for Taiwan news Web sites is running at http://nse.yam.com/.

the interval 0.1. In par with the hand-coding data, the recall rate of each site and corresponding H(CB) intervals are shown in Figure 5. The X-axis is the increase of the threshold for H(CB) and the Y-axis is the recall rate based on the answer set. The recall rate is equivalent to the number of features included in selected blocks due to the increase 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 in Figure 5, the optimal thresholds of ET and CTV are 0.2 with respect to recall rate 1.0 since it is converged after this point. The result shows that all sites, except for UDN, have very high recall rates (at least 0.956). These optimal thresholds of sites are distributed from 0.1 to 0.7. For example, CNet is converged at 0.5 with recall 0.956. That is, optimal thresholds vary among different sites. The recall of UDN, 0.760, is not perfect. By tracing to the training pages and corresponding hand-coding data, we found that the hand-coding data of UDN is wrong because of the inclusion of the title information of news categories. Consequently, the greedy approach is able to dynamically find the optimal threshold of H(CB) for different Web sites.

Table 1: News sites with tabular pages.

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

TTimes was closed at February 21, 2001.

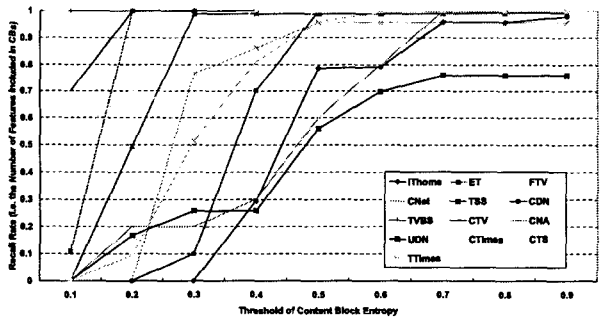

Figure 5: Recall rate of each page cluster.

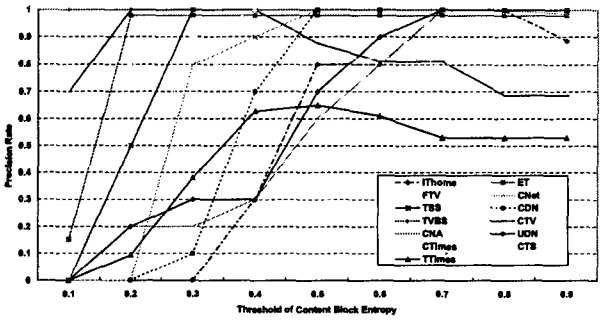

Figure 6: Precision rate of each page cluster.

The precision rate is shown in Figure 6. In par with the recall rate shown in Figure 5, the threshold associated the highest precision rate of each page cluster almost corresponds with its optimal threshold of H(CB). The result is summarized in Table 2. It shows that all sites have perfect precision rates, except for TTimes. The precision of TTimes is 0.530 at the optimal threshold of H(CB) 0.7. However, the highest precision is 0.651 at H(CB) 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 text length of the block is even longer than that of the article in many pages. If contents of "related news" are different among training pages, their corresponding H(CB) will be low enough to become an informative content block. Thus, this kind of block is also included. These included noisy features affect the decision of the threshold. However, in the case of "related news" content block, the judgment of informative or redundant is ambiguous since it depends on the perspective of users.

Table 2: Recall and precision at optimal threshold of H(CB)

Site	Optimal H(CB)	Recall	Precision
IThome	0.7	0.957	1.000
ET	0.2	1.000	0.979
FTV	0.4	1.000	1.000
CNet	0.5	0.956	1.000
TSS	0.3	0.989	1.000
CDN	0.5	1.000	1.000
TVBS	0.1	1.000	1.000
CTV	0.2	1.000	1.000
CAN	0.7	1.000	1.000
UDN	0.7	**0.760**	1.000
CTimes	0.4	1.000	1.000
CTS	0.5	1.000	0.959
TTimes	0.7	0.997	**0.530**

These experiments prove that the greedy approach achieves very high recall and precision of at least 0.956, except for the precision of TTimes and the recall of UDN. As we described, the low recall of UDN is for the reason of wrong hand-coding data and the low precision of TTimes is due to the content block "related news". Consequently, our approach almost achieves a perfect recall and precision in discovering informative contents from tabular Web pages.

To investigate the effect of the number of randomly selected training examples, we redo the same experiments on all page clusters. Since UDN has wrong hand-coding data and pages of TTiems contain semantically ambiguous content blocks of related news, both sites are not included in the experiments. The number of training examples is started from 5 to 40 with interval 5. The result is shown in Figure 7, in which the dotted line denotes the recall rate (R) and the solid line represents the precision (P). Most clusters have perfect recall and precision rates approaching to 1 (many R or P lines are overlapped at the highest value 1.0), but precision rates of few clusters (solid lines) are not when the number of randomly selected examples is increased. It reveals that the number of examples may have an influence on the precision rate since the precision rates of CTS, ET, and CTimes are degraded below 0.9 when the number is increased. And the random number almost has no effect on the recall rate since most dotted lines have recall rates larger than 0.956, except for CNet's 0.942. 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 higher precision. However, the

recall rate is not affected because higher threshold means more features included.

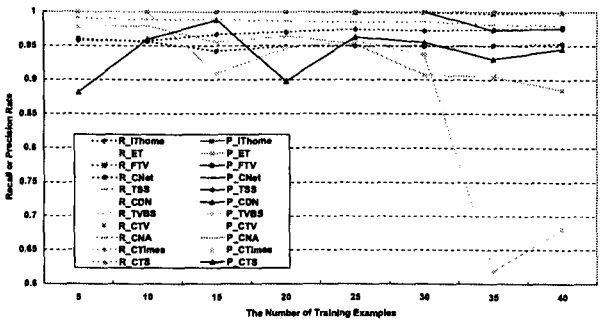

Figure 7: R/P based on the number of training examples.

6. CONCLUSION

According to previous experiments, we can conclude that our proposed methods are feasible to discover informative contents from Web pages of the same site. The greedy approach of InfoDiscoverer is adaptive to find the optimal threshold of block entropy for different Web sites with different templates. Based on this approach, the optimal threshold of informative content blocks is dynamically selected for different sites. The result shows that both recall and precision rates are larger than 0.956, which is very close to the hand-coding result.

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 Web IR systems (search engines) by reducing the size of index and increasing the precision of retrieval. The complexity of the discovering process is polynomial. Most important, intermediate results, such as the page representation and keywords with weights, generated by the InfoDiscoverer are shared with the crawler and indexer of Web IR systems. Applying InfoDiscoverer to Web IE systems is also efficient as these systems simply consider smaller informative content blocks instead of the whole page content. Thus, InfoDiscoverer can be the preprocessor Web IR and IE systems.

The proposed method is applied to tabular Web pages and based on the assumption of knowing page clusters. Experiments are merely evaluated for Chinese pages published by news Web sites. In the future, we will develop other methods to automatically discover page clusters from Web sites. To make our method be applicable to general Web pages instead of restricting to tabular pages, we will apply generalization and specialization processes to merge or split content blocks based on HTML document object model. Also, evaluations on English Web pages are important to polish our method.

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.

[1] Blahut, R. E., "Principles and Practice of Information Theory," Addison Wesley, 1987.

[2] Brin, S. and Page, L., "The Anatomy of a Large-Scale Hypertextual Web Search Engine," the Seventh International World Wide Web Conference, 1998.

[3] Brin, S. and Page, L., Google Search Engine, http://www.google.com/.

[4] Cardie, C., "Empirical Methods in Information Extraction," AI Magazine, 18(4):5-79, 1997.

[5] 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/.

[6] Chidlovskii, B., "Wrapper Generation by k-Reversible Grammar Induction," Workshop on Machine Learning for Information Extraction, August, 2000.

[7] Chien, L. F., "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval", Proceedings of the ACM SIGIR International Conference on Information Retrieval, 1997.

[8] Cowie, J. and Lehnert, W., "Information Extraction," Communications of the ACM, 39(1):80-91, 1996.

[9] Frakes, W. B. and Baeza-Yates, R., "Information Retrieval – Data Structures & Algorithms," Prentice Hall, 1992.

[10] Freitag, D., "Machine Learning for Information Extraction," Ph.D. Dissertation of Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1998.

[11] 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.

[12] Kleinberg, J. M., "Authoritative Sources in a Hyperlinked Environment," Journal of the ACM, 46(5):604-632, 1999.

[13] Kosala R. and Blockeel, H., "Web Mining Research: A Survey," SIGKDD Explorations, 2(1):1-15, 2000.

[14] Kushmerick, N., "Wrapper Induction for Information Extraction," Ph.D. Dissertation, Department of Computer Science and Engineering, University of Washington, 1997.

[15] Porter, M., "The Porter Stemming Algorithm," http://www.tartarus.org/~martin/PorterStemmer/.

[16] Salton, G., "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer", Addison Wesley, 1989.

[17] Shannon, C., "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27, pp. 379-423 and 623-656, July and October, 1948.

[18] Shasha, D. and Wang, T., "New Techniques for Best-Match Retrieval," ACM Transactions on Office Information System, 8(2):140-158, 1990.

[19] W3C DOM, "Document Object Model (DOM)," http://www.w3.org/DOM/.

[20] W3C HTML, "HyperText Markup Language," http://www.w3.org/MarkUp/.

[21] W3C XML, "Extensible Markup Language," http://www.w3.org/XML/.

[22] Wang, K. and Liu, H. Q., "Discovering Structural Association of Semistructured Data," IEEE Transactions on Knowledge and Data Engineering, 12(3):353-371, 2000.