Extracting Unstructured Data from Template Generated Web Documents

Ling Ma
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
maling@ir.iit.edu

Nazli Goharian
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
goharian@ir.iit.edu

Abdur Chowdhury
America Online Inc.
cabdur@aol.com

ABSTRACT

We propose a novel approach that identifies web page templates and extracts the unstructured data. Extracting only the body of the page and eliminating the template increases the retrieval precision for the queries that generate irrelevant results. We believe that by reducing the number of irrelevant results; the users are encouraged to go back to a given site to search. Our experimental results on several different web sites and on the whole cnnfn collection demonstrate the feasibility of our approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Search Process

General Terms

Design, Experimentation

Keywords

Automatic template removal, text extraction, information retrieval, Retrieval Accuracy

1. INTRODUCTION

A significant portion of the data on the World Wide Web is in the form of HTML pages. HTML pages are designed to describe the formatting of text, navigational functionality and other visual aspects of a document. Since content, navigational information, and formatting have no clear separation in HTML, the conventional information retrieval systems have the additional task of dealing with noisy data when providing full-text search. For example, the query "health technology law" on the *cnn* site retrieves every page that has "Health", "Law", or "Technology" in the navigation bar even if the body text of page is not relevant to the query. In addition to the formatting data, the advertisement section of a page contributes to the same problem. A problem that is not well studied is the negative effect of such *noise data* on the result of the user queries. Removing this data improves the effectiveness of search by reducing the irrelevant results. Furthermore, we argue that the irrelevant results, even covering a small fraction of retrieved results, have the *restaurant-effect*, namely users are ten times less likely to return or use the search service after a bad experience. This is of more importance, considering the fact that an average of 26.8% of each page is formatting data and advertisement. Table 1 shows the ratio of the formatting data to the real data in the eight web sites we examined.

Table 1: Template Ratio in Web Collections

Web page Collection	*Template Ratio*
Cnnfn.com	34.1%
Netflix.com	17.8%
NBA.com	18.1%
Amazon.com	19.3%
Ebay.com	22.5%
Disneylandsource.com	54.1%
Elle.com	19.3%
Launch.yahoo.com	29.3%

We present a novel approach to automatic text extraction and experimentally show the improvement on the search accuracy. The traditional approach to extract relevant text from a page uses a wrapper program according to a page tag structure. Programmers manually find the absolute tag path for the targeted data and customize the extraction code for each type of document. This method, however, has many shortcomings. If a wrapper is not generated from the pages belonging to a web template, an error may occur; or if the site is modified in any tagging level, also an error may occur. Furthermore, many commercial sites are backed by commodity databases that generate HTML dynamically and may correspond to slightly different templates. The difficulty in managing the customized wrappers grows with an increasing number of sites. Therefore, manually coding and maintaining wrappers does not provide a realistic solution.

CIKM '03, November 3--8, 2003,New Orleans, Louisiana, USA

*This work is supported in part by the National Science Foundation under contract # 0119469.

2. RELATED WORK

Crescenzi, et al., in RoadRunner project [1] applied matching techniques in reconstructing template with union free regular expression grammar. Targeting at the limitations of RoadRunner, Arasu et al [2] developed EXALG (extract algorithm) for data extraction**.** EXALG uses equivalent class, which is a maximal set of tokens having the same term frequency vector in every page of the collection. EXALG can also handle disjunction grammars. With disjunction, one data field can have several different grammar representations. In comparison, the main focus of our work is to extract unstructured text while removing the noise such as template navigation bar and advertisement, thus no grammar induction is involved. Ahonen-Myka [3] and Beil, et al., [4] designed efficient algorithms for the frequent term set clustering. We take advantage of the page tag tree structure to find the template in HTML pages, as it is more straightforward than using the frequent term set clustering. Bar-Yossef, et al., observed that templatized pages from the professional designed websites share the same context for browsing [5]. Their result showed that template skews ranking and reduces precision. They introduced the notion of *pagelets*: a self-contained logical region within a page that has a well-defined topic or functionality. Similarly, Li, et al., proposed segmenting the web page into topically more focused micro information units (MIU) [6]. Their algorithm employs text font property and term set similarity, building HTML page tag tree, merging heading with text, and considering adjacent similar paragraphs as MIU. Their results show a high ranking, if every term of a phrase query is located within the same MIU. We also employ tag information to locate and remove the template unit. For information retrieval background, see [7].

3. OUR APPROACH

A critical challenge is to find a unit that optimally separates template from the text. To achieve this, we must first answer the question: What is a template? According to our observation, a template is a consecutive group of text tokens that: (a) appear in every page belonging to that template, (b) share the same geometrical location and size within the web pages, (c) serve primarily as navigation, trademark, or advertising without providing other information.

The first task is to define a unit. The data wrapping tags such as <TR> and <TD> separate a page into very small units. It is quite often that such a unit only contains a single token. The use of a single token as a segmentation unit results in false positives and unnecessary computation, as some non-template data also repetitively appear in many pages. Many web sites choose table tag to wrap the templates due to its geometric and layout parameters that help to distinguish template from text. As the result of our studies we define *table text chunk* as the segmentation unit. A *table text chunk* is consecutive terms extracted between a pair of closest HTML table tags.

Figure 1 illustrates the text extraction process. During the preprocessing, the duplicate web pages in the collection are identified and removed. Two pages with different URLs may have the same content because of page redirection and other reasons. We use the *IMatch* duplicate detection algorithm [8] for determining duplicates. The *IMatch* algorithm creates Unicode ascending tree for a document by inserting only the most infrequent terms or removing the most frequent terms. SHA1 hash

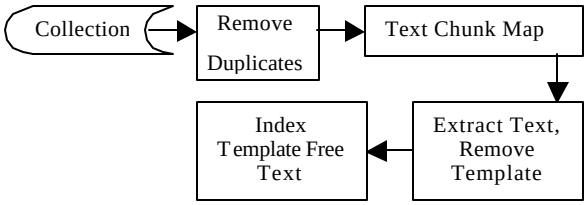


Figure 1: Text Extraction Process

value is computed from the tree. Two documents are duplicates if the SHA1 hash of their tree collide.

After duplicates are removed, our algorithm has two passes over the HTML pages. In the first pass, whenever a deliminator tag or its respective *end tag* such as *table tag* or *image map tag* is encountered, we store the identified text chunk in the *text chunk map* along with the document frequency for that chunk. The document frequency for each *text chunk* is updated while processing the whole collection.

In the second pass, all text chunks that their document frequency is over a determined threshold are identified as *template table text chunk.* The remaining text chunks are the extracted texts that are passed to the *indexer to* be indexed. Thus, the output of the second pass is the extracted text without HTML formatting/ template data.

4. EXPERIMENTATION AND RESULTS

4.1 Data

We used *cnnfn* web site, which is a 2GB financial collection of 55,711 pages from 1998-2003 with 248MB text. The template removal process identified 53MB of template text in this collection. Thus, the size of the template free text is 195MB. This indicates an average template ratio of 21.3%. Note that this ratio is much less than 34.1% of a specific *cnnfn* collection with only 8 pages. We used a query log from AOL with 10,000 *cnnfn* top user queries. Among the 10,000 most frequent queries submitted to *cnnfn* website, most of the queries were single terms, and the rest were phrases or sentences.

4.2 Experiments

A web page collection may have many templates, for example, a template for weather news, a template for financial news, a template before pages produced in year 1997 and a template after year 1997. Initially, we clustered pages belonging to the same template and then removed the template assuming it leads to a more accurate extraction. We clustered 1,276 html page using K. Fukunaga's *min-max* clustering approach [9]. By manual checking, we found that 96.7% pages were clustered correctly. A 1% wrongly clustered pages within a 10,000 page collection indicates that the template of 100 pages are not removed, causing a false higher relevancy due to the term frequency of template term. Thus, we decided not to use clustering in the remaining experiments.

Our next set of experiments validated our claim that "extracting the text and removing the template reduces the number of irrelevant retrieved results". To demonstrate the validity of our claim, we had to identify the queries that their retrieval precision

differs before and after template removal. Section 4.3 describes the framework and the results of search accuracy.

Section 4.4 demonstrates the efficiency and accuracy of our proposed extraction method in terms of the ratio of successful extraction.

4.3 Effects on Search Accuracy

Logically, removing the template has an impact mainly on the result of the queries that contain the template terms. Out of the 10,000 AOL queries, 800 queries contain the template terms. Out of these 800 queries, 520 of them are concrete noun phrases: e.g. "bill gates" and "interest rate cut". We observed that these concrete noun queries produce the same precision before and after extraction. Thus, we used the remaining 280 queries that are composed of abstract nouns.

It is critical to check query result relevancy with adequate and consistent criteria. Document relevancy is not defined solely based on the topic of query, but on its capability to provide information that can be used. According to Froelich [10], the criteria for deeming a document relevant are topicality, perceived validity by the user, and novelty. Su and Chen [11] proposed an evaluation model that consists of measures of effectiveness, efficiency, user satisfaction, reliability of connectivity, and user characteristics.

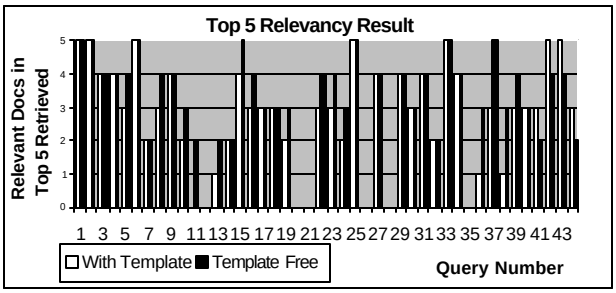
Using those criteria, we recruited human evaluators to evaluate the relevancy of retrieved query result. Before the evaluation, the human evaluators wrote down what information they were looking for at the time of issuing a given query. During evaluation, we count a result as relevant if (a) it is perceived relevant to the topic of search specified by the user; and (b) it provides useful information for the user. We compare the number of relevant documents in the top 1, 5 and 10 results with the original *cnnfn* collection and the collection generated after extraction. The results are presented and discussed in 4.3.1.

Our experiments were conducted over a traditional inverted index based search engine. We applied Okapi BM25 [12] probabilistic ranking function.

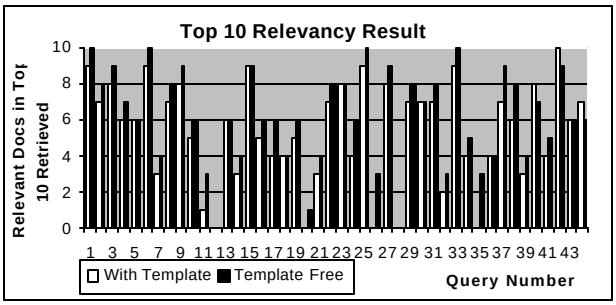
4.3.1 Discussion on Accuracy Results

We observed that the search results improved for 39 queries out of the 280 queries with abstract nouns; and deteriorated for 5 queries, after removing the template. The remaining queries do not show a difference in the retrieved documents before and after template removal. Figure 2 (a,b) illustrates a comparison between the retrieved results for these 44 queries before and after template removal. As shown in Figure 2, the impact of the template removal on the relevant retrieved gradually manifests as more documents are retrieved. Top 5 and top 10 relevancy results demonstrate that template removal boosts precision notably. The precision at the only top document does not differ before and after template removal.

Our data shows that if one would have used a system that switches between two engines, one based on extracted text and one based on non-extracted text, then a 100% improvement can be achieved on irrelevant results retrieved by the engine based on non-extracted text. This result is on our *cnnfn* collection and its queries. This number can be substantial when dealing with real life various sites and large number of queries. On the other hand only 1.7% of



(a)



(b)

Figure 2: Relevant Documents Retrieved With and Without Template

results may get worse, if using only a single engine based on extracted text.

We noticed that a main reason that led to bad results in these 5 queries in our experiments was the inadequate use of phrasing in the user query. For example for the query "computer" the human evaluator intended to look for the general computer business news, however, several documents about "computer science inc." were also retrieved.

Let T be the number of queries that generate different results before and after extraction; N the number of queries that have better precision after extraction; and M the number of queries that have lower precision as the result of extraction. The experiments show that N/T>>M/T, thus, extraction is useful for queries that generate irrelevant results. The number of "bad" results retrieved by the user queries over a period of time is reported as motivating factor for the users not to go back to a given search engine or provider. Assuming we have an infinite population of queries, and we sample *x* queries producing 90% confidence level and 5% error rate. Thus, any event or query type with a 5% or less occurrence may not be captured in the study. While a 5% problem may not seem important, when examined as a reduction in negative user experiences it becomes a valuable problem to be solved for users.

4.4 Extraction Efficiency Results

Given that our approach improves the effectiveness of the search, we next question how efficient our system performs the extraction. We ran our algorithm over a sample collection of various web pages and validated the template detection of our approach as illustrated in Table 2. As the column *Template Table Discover Ratio (ttdr)* shows, our algorithm detected all template tables from all web pages in sample collection, but the NBA.com.

Table 2: Template Table Discovery Ratio

Web Page Small Collection	*ttdr*
Cnnfn.com	1.00
NBA.com	0.86
Netflix.com	1.00
Amazon.com	1.00
Ebay.com	1.00
Disneylandsource.com	1.00
Elle.com	1.00
Launch.yahoo.com	1.00

A template table discovery ratio of 1 indicates every template table is discovered; a value larger than 1 indicates some text are counted as part of the template by mistake; and a value less than 1 means some template tables are not discovered. The values are generated by calculating the ratio of template tables detected to actual number of template tables of a page. To check the accuracy and scalability of our approach, we also check the *ttdr* on the *cnnfn* collection (55,711 pages) by statistical sampling. let D represent the minimum detectable difference and S represent the standard deviation, and Z represent the 1-alpha/2 percentile of a standard normal distribution, then the appropriate sample size would be: sample size = (Z^2 * S^2) / D^2. We randomly sampled 200 pages out of the large collection (90% confidence level with a 5.8% sample error). By manually checking the documents, we found that 92.3% of the templates are discovered.

The lower ratio for the template detection in a collection contributes to the document frequency threshold that is used to identify *table text chunk*. Figure 4 illustrates the *template table discovery ratio* when picking different number of pages from Amazon collection, with document threshold set to 4.

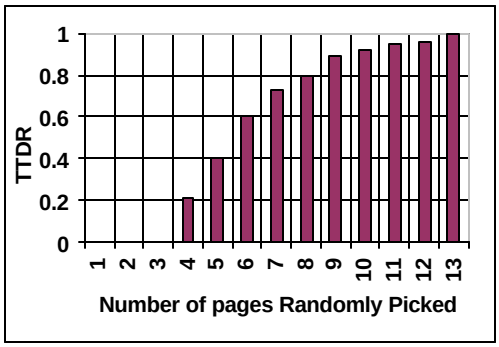


Figure 3: TTDR for Random Amazon Pages

We observed that most of the template tables appear in every page of the collection. Thus, our approach can be even faster by running on less number of pages. In one experiment, we randomly picked 9 out of the 13 Amazon web pages. The program could still recognize 86% of the template tables. This trade-off saves the execution time for the large web page clusters. There are 41 web pages in the ebay.com cluster, picking 9 pages randomly achieved *template table discovery ratio* of 1 (ttdr=1), which indicates that every template table was still discovered. Meanwhile, the processing time of our program is reduced to 27% of the running time when using all 41 pages.

5. CONCLUSIONS

We designed a novel approach for automatic text extraction to identify web page template data and extract unstructured data. Our experimental results demonstrated that using extraction reduces the irrelevant results for the queries that generate "bad" results. Retrieving irrelevant results as the result of a search is a motivating factor for users to avoid a given search engine or provider. Our Experimental Results shows that if one would use a system with extracted text instead of non-extracted text, then a 100% improvement can be achieved on irrelevant results retrieved by the engine based on non-extracted text. On our experiment with cnnfn collection and AOL user queries, we improved all bad results.

REFERENCE

[1] V. Crescenzi, G. Mecca, P. Merialdo, RoadRunner: "Automatic Data Extraction from Data-Intensive Web Sites", *Special Interest Group on Management of Data (SIGMOD02)*, 2002.

[2] A. Arasu, H. Garcia-Molina, "Extracting Structured data from Web pages", *Special Interest Group on Management of Data (SIGMOD03*), 2003.

[3] H. Ahonen-Myka, "Discovery of Frequent Word Sequences in Text", Lecture Notes In Computer Science, Springer-Verlag Heidelberg, 2447/2002, 180-189, ESF Exploratory Workshop, 2002.

[4] F. Beil, M. Ester, X. W. Xu, "Frequent Term-based text clustering", *ACM Special Interest Group on Knowledge Discovery in Data (SIGKDD02)*, 2002.

[5] Z. Bar-Yossef, S. Rajagopalan, "Template Detection via Data Mining and its Applications", *WWW02*, 2002.

[6] X. Li, B. Liu, T. Phang, M. Hu, "Using Micro Information Units for Internet Search", *ACM Special Interest Group on Knowledge Discovery in Data (SIGKDD02*), 2002.

[7] David A. Grossman, Ophir Frieder, "Information retrieval algorithms and heuristics", *2nd ed. Kluwer Academic Publishers*, 2003.

[8] A. Chowdhury, O. Frieder, D. Grossman, and M. McCabe, "Collection Statistics for Fast Duplicate Document Detection," *ACM Trans. on Information Systems (TOIS*), 20(2), April 2002.

[9] K. Fukunaga, "Introduction to Statistical Pattern Recognition", *Academic Press, Inc*., Boston, second edition ,1990.

[10] T. Froehlich, M. Eisenberg,, "Special topic issue on relevance research", *Journal of the American Society for Information Science*, 45 (3), 124-134., 1994.

[11] L. Su,, H. Chen,, and X. Dong, "Evaluation of Web-based search engines from the end-user's perspective: a pilot study", *Proc. of the Conf. for the American Soc. for Inf. Science*, 1998.

[12] S.E. Robertson, S. Walker and M. Beaulieu. "Okapi at TREC-7: automatic ad hoc, filtering, and interactive", Proceedings of the Seventh Text REtrieval Conference (TREC-7), 1999.

Additional Author: Misun Chung
Information Retrieval Lab ,Computer Science Department,
Illinois Institute of Technology,
chunmi@ir.iit.edu