Methods For Extracting Content Blocks From Web Pages

Presented By
Rajashri Shinde, Ashwini Bolli, Amruta Kulkarni
Solapur

Abstract

The Web is perhaps the single largest data source in the world .The coverage of Web information is very wide and diverse. It has information which is of type required information by the user i.e. content blocks of the pages & the rest irrelevant information is termed as non content information or blocks like banner ads, navigation bars, and copyright notices. Web mining aims to extract and mine useful knowledge from the Web. But the non content blocks causes harm to web mining .So as to enhance web mining there is necessity of differentiate between contents & non contents blocks and to eliminate the non content blocks from web pages. So as to perform this task this paper deals with some techniques and methods which ultimately provides significant storage and timing saving by providing content blocks from web pages to user.

Index Terms

Entropy value, content information index, common structure, tag ratio, primary contents.

.

1. Introduction

The World Wide Web is a repository of information. All the web information is published over the HTML pages on internet. Search engines crawls the World Wide Web to collect web pages. Basically, the user interacting with the web pages for achieving his required information, which causes difficulty to WWW to identify content blocks from web pages. They often contain non content blocks like advertisements, image-maps, logos, search boxes, navigational links, related links, footers and headers, and copyright information along with the content blocks. The non content blocks are not relevant to the content blocks of the page. Hence there must be facility of distinguishing the contents & non contents from

Web pages. The advantage of identifying non-content blocks from web pages is that if user does not want non-content blocks these can be deleted. These non-content blocks are normally large part of the web pages so eliminating them will be a saving in storage and indexing. This facility causes increases performance of search engine.

The techniques mentioned are based on following observations.In case of web pages using same template style, to extract content blocks from web pages, it will partition page according to HTML tags & calculates the entropy value of block with the feature set of pages. By considering the threshold entropy value it distinguishes the content & non content blocks.[1] In an algorithm k-Maximum Informative Block Mining which uses DOM tree to find out the top-k nodes with maximal aggregated feature values under the given constraint to achieve content blocks.[2]Then in compressed structure tree method which captures the commonalities of the pages in a Web site. Based on this tree, study of some statistical properties of the structures and the contents of the Web page can be collected.[3] In the method of Content Extraction via tag ratio applied on web pages using style sheets & some set of tags, it linearly does calculation of tag ratio on HTML document & defining a threshold value to tag ratio determines content & non content blocks from web pages[4]. The algorithm of content extractor which does partitioning of web pages into blocks using Get Block algorithm & eliminates the non content blocks by finding inverse block document frequency (IBDF) of a block.[5] In block based web search block retrieval is done by ranking document& query expansion by weighting web pages using page segmentation method.[6]

2. APPROCHES:

2.1 Discovering Informative Content Blocks by calculating entropy value of features

Here in this technique first a page is partitioned 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, a method is proposed 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. Here the focus is on the problem of intra-page redundancy instead of the Internet page redundancy.

A Web site usually employs one or several templates to present its Web pages. If all pages of a Web site use the same template, the Web site is regarded as one page cluster. 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>. The second phase is to refine the granularity of the tree while the classification of content blocks is ambiguous. After parsing a page into content blocks, features of each block are simultaneously extracted. In this technique features correspond to meaningful keywords. Stop words are not included. Applying the Porter stemming algorithm and removing stop words in the stop-list, English keywords (features) can be extracted. Then in the next step 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. Considering all pages in a cluster, these lists of pages form the feature-document matrix (F-D Matrix). Based on the F-D Matrix, measuring the entropy value of a feature corresponds to calculating the probability distribution in a row of the matrix. After this the entropy of the content block is estimated. The entropy value of a content block is the summation of its features entropies. That is, the entropy of a content block is the average of all feature entropies in the block. Now the content blocks are classified as follows: Based on entropy of a content block, the content block can be divided into two categories: redundant and informative. If entropy of a content block 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 entropy of a content block is less than a defined threshold, the content block is informative because features of the page are distinguishable from others.

According to the proposed method, we can conclude that these methods are feasible to discover informative contents from Web pages of the same site. The greedy approach of this technique 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 proposed method is applied to tabular Web pages and based on the assumption of knowing page clusters.[1]

2.2 k-maximum informative block mining

Most commercial Web sites, such as search engines, portal sites, e-commerce stores, and news, apply a systematic technique to generate Web pages and to adapt various requests from numerous Web users. These sites are referred to as systematic Web sites . The evolution of automatic Web page generation and the sharp increase of systematic Web sites have contributed to the explosive growth of Web page numbers. There exists much redundant and irrelevant information in these Web pages such as navigation panels, advertisements, catalogs of services, and announcements of copyright and privacy policies which are distributed over almost all pages of a systematic Web site. Such information is still crawled and indexed by search engines and information agents, thus significantly increasing corresponding storage and computing overhead. Here in this technique called Web Intrapage Informative Structure Mining Based on Document Object Model the specific regions of a page that users are interested in as informative blocks (or referred to as IB). The set of these blocks and corresponding connecting structures form the informative structure (or referred to as IS) of the page. This technique works with ISs of individual pages, called intrapage informative structure.

This technique automatically extracts and recognizes ISs of each page in a Web site according to the knowledge in the tree structures of pages. This technique works in three phases: 1) information extraction from DOM trees, 2) k-maximum informative block mining, and 3) block expansion and condensation. In the first phase, the useful features are extracted from the information of the original DOM tree. These features can be classified into two types of information: node information and structure information. According to the extracted information, the technique calculates three extended features to gain more implicit information from the tree, namely, 1) the content information index (CII) which indicates the amount of information contained in the block, 2) the anchor precision index (API) which represents the similarity between the anchor-text and the linked document, and 3) the structure information index (SII) which indicates the distribution of children's feature values of one node in the DOM tree. Each node in DOM tree T contains the tuple values of the feature set F ={ALEN,CLEN, API}. In the second phase, the method aggregates the node information to build the Information

Coverage Tree (ICT). According to the ICT, a greedy algorithm is devised, i.e., k-maximum informative block mining algorithm (k-MIB), to extract subtrees that contain richer information. They form the skeleton set of the IS of a page.

In this phase i.e.The k-Maximum Informative Block mining first the information coverage tree is build for features extracted during the phase one to obtain corresponding aggregated feature values. A tree with bottom-up aggregated features as an information coverage tree (ICT). In an ICT, any feature in the aggregated feature set FA is obtained from the corresponding features in set F. Each node in an ICT contains all feature information of nodes in the subtree rooted by this node. The feature aggregation is a bottom-up process from the leaf nodes to the root node.

The proposed k-MIB algorithm is then applied to extract and filter out the candidate IBs. The proposed maximum informative block mining algorithm MIB(k, fA, ST) is a greedy and top-down tree traversal process. For input value k, the algorithm outputs at most k IBs, i.e., TOC blocks or article blocks. The aim of the algorithm is to find the top-k nodes with maximal aggregated feature fA values under the given SII constraint, i.e., SII Threshold (ST). When the value of ST is larger, the structure constraint is tighter and the children of each extracted node in the resulting candidate set will have more similar values of aggregated features in accordance with the definition of SII.[2]

2.3 Compressed Structure Tree (CST)

This cleaning technique is based on analysis of both layouts (structures) and contents of the Web pages in a given Web site. Thus, in this technique first task is to find a suitable data structure to capture and to represent common layouts or presentation styles in a set of pages of the Web site. This technique proposed the compressed structure tree (CST) for this purpose.

Each HTML page corresponds to a DOM tree where tags are internal nodes and the actual text, images or hyperlinks are the leaf nodes. Although a DOM tree is sufficient for representing the structure of one HTML page and it has been used in many applications, it cannot represent the common structure of a set of Web pages. Here aim is to compress individual DOM trees of Web pages into a single tree (compressed structure tree) which captures the commonalities of the pages in a Web site. Based on this tree, study of some statistical properties of the structures and the contents of the Web pages can easily be done.

Before defining the compressed structure tree, we first define the presentation style of a tag node in a DOM tree. The *presentation style* of a tag node *T* in a DOM tree, denoted by *ST*, is a sequence <*r1*, *r2*, ., *rn*>, where r_i is a pair (*Tag*, *Attr*) specifying the i^{th} child tag node of *T*.

- *Tag* is the tag name, e.g., .TABLE. and .IMG.
- *Attr* is the set of display attributes of *Tag*, e.g., bgcolor = RED, width = 100, etc.

n is the length of the style. For example, in Figure 1, the presentation style of tag node BODY is <(TABLE, {width=800, height=200}), (IMG, {width=800}), (TABLE, {bgcolor=red})>.

We say two presentation styles *Sa*:< ra_1, ra_2, ., ra_m> and *Sb*:< rb_1, rb_2, ., rb_n> are equal, i.e., *Sa* = *Sb*, *iff* $m = n$ and $ra_i.Tag = rb_i.Tag$ and $ra_i.Attr = rb_i.Attr$, $i = 1, 2, ., m$.

An element node (the basic information unit) of a compressed structure tree (CST) is defined as: An *element node E* represents a set of merged tag nodes in the DOM tree. It has 5 components, denoted by (*Tag*, *Attr*, *TAGs*, *STYLEs*, *CHILDs*), where

- *Tag* is the tag name;
- *Attr* is the set of display attributes of *Tag*.
- *TAGs* is the set of actual tag nodes in the original DOM trees that are compressed (or merged) in *E*.
- *STYLEs* is the set of presentation styles merged into *E*.
- *CHILDs* is the set of pointers pointing to the child element nodes of *E* in CST.

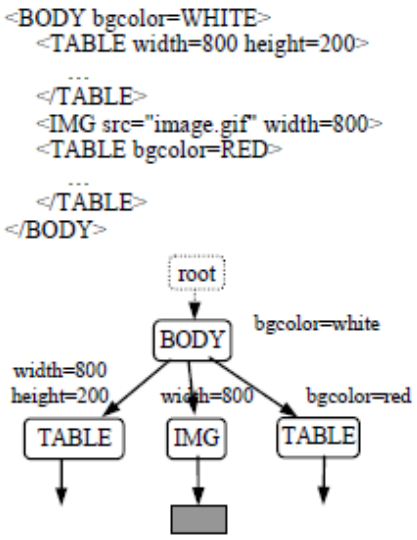

Figure 1: A DOM tree example (lower level tags are omitted)

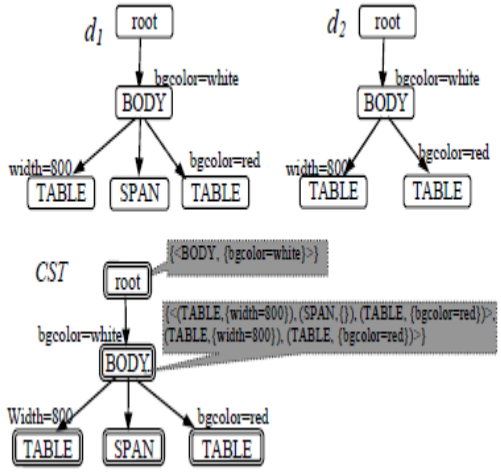

Figure 2: DOM trees and the compressed structure tree (lower levels of trees are omitted)

An example of compressed structure tree is given in Figure 2, which compresses the DOM trees *d1* and *d2*.

CST of a set of Web pages is built from a Web site by merging their DOM trees from top to bottom in following steps:

1. All root tag nodes of the DOM trees are merged to form the first (the top most) element node of the CST. We have *Tag* = root, *Attr* = {}, and *TAGs* being the set of all the root tag nodes in the DOM trees of the Web pages.
2. We compute *STYLEs* of the element node *E* passed from the previous step. *E.STYLEs* is the set of presentation styles of all the tag nodes in the DOM trees covered by *E*. Note that common presentation styles are combined.
3. All the corresponding child tag nodes of those (tag nodes) in *E.TAGs* with the same presentation style are merged, which form the initial child element nodes of *E*.
4. If no child element node is created in step 3, stop; else for each child element node from step 3, go to step 2.

After the CST is built, it is used to compute a weight for each word feature in each block of a Web page. Intuitively, if an element node contains many different presentation styles, it is more likely to be important and hence should be assigned a high weight. Otherwise, it will
be assigned a low weight, i.e., it is more likely to be noisy. The technique uses the entropy of presentation styles to encode the importance of an element node *E* in the CST.[3]

2.4 CETR-Content Extraction via Tag Ratios

CETR Algorithm:
To design this algorithm they used style sheets and or tags for structural information. In the CETR algorithm it construct a tag ratio (TR) array with the contention that for each line in the array, the higher the tag ratio is for the line the more is the content-text within the HTML document which is represented as histogram. Tag Ratios (TRs) are the basis by which CETR analyzes a webpage in preparation for clustering. TRs, essentially, are the ratios of the count of non-HTML-tag characters to the count of HTML-tags per line. Before computing TR ratio script, remark and style tags are removed from the HTML document because this information would be treated as non-tag text by the algorithm and likely skew the results. Empty lines are also removed because their inclusion would potentially hinder the performance of the clustering procedure.
Example 1. If Html page has following data:
1.
3. James Smith
4. OKLAHOMA CITY - Police were told that. . .
5. . . . The Oklahoman reported Sunday.

Jones had. . .

The Tag Ratios for these lines are computed as follows:

1. Text=0, Tags=1, TR=0
3. Text=11, Tags=2, TR=5.5
4. Text=37, Tags=0, TR=37
5. Text=41, Tags=2, TR=20.5

After the TR-histogram *T* is calculated a smoothing pass is made on the histogram. So that important content lines should not get lost.

A. Selecting Content from Threshold:

Now the threshold method is used where it defines τ which discriminates TRs into content and non-content sections. That is, any TR value greater than or equal to τ should be labeled content otherwise is non content. Where it initially removes all non-tag chars. This method is CETR-TM (Threshold Method).

B. Selecting Content via Clustering:

Alternatively, we apply the k-means clustering method for that it groups content & non contents into clusters. The resulting k clusters are used for selecting the cluster which has its centroid closest to the origin is non content. The remaining clusters are assigned to content. This method is CETR-KM (k-Means Method).
This algorithm also operate on 2D MODEL, They view set of values of histogram as ordered sequence so as to cause jumps in TR Histogram properly. Hence, it Constructs 2D Tag Ratios & performs Clustering of 2-dimentional histogram of Tag Ratios. This approach has two beneficial ways (1) It forces the remaining clusters to migrate away from the origin where the non-content points are

located, and (2) it provides an easy means for labeling the resulting clusters; specifically, the cluster with the origin-centroid will always be labeled no content because points near the origin most likely represent non-content points, All remaining clusters are therefore labeled content.[4]

2.5 Content-Extractor Algorithm

To design content-extractor algorithm it uses the predefined ordered set of tags started with <TABLE>. Similar blocks across different web pages obtained from different web sites can also be identified using this algorithm. In a table occurring in a web page, each cell is considered as a block. Where tables are not available, blocks can be identified by partitioning a web page into sections that are coherent. Many times news articles written by global news agencies appears in many news papers. User wants only one of these several copies of articles. These copies of articles differ only in their non-content blocks, so by separating non-content blocks from content blocks these same copies can be identified. As only unique articles are returned this will improve search results.

Content block can be identified based on the appearance of the same block in multiple web pages. The algorithm first partitions the web page into blocks based on different HTML tags. The algorithm then classifies each block as either a content block or a non-content block. The algorithm compares a block, B, with the stored block to check whether it is similar to a stored one, if so then it is not necessary to store that block again.

A block or web-page block *B* is a portion of a webpage enclosed within an open-tag and its matching close tag. The open and close tags belong to an ordered tag-set *T* that includes tags like<TABLE>, <TR>, <P>, <HR>, and <UL>. <TABLE> comes as the first tag in that list. The order of the tags is based on the observations of webpage design. For example, <TABLE> comes as the first partitioning tag since more instances of <UL> in <TR> <TD> which are sub-element of <TABLE>, than <TABLE>s coming inside <LI>. Algorithm partitions a web-page into blocks, based on the first tag in the list. It continues sub-partitioning the already-identified blocks based on the next tags in the list. Blocks may include other smaller blocks, and have features like text, images, applets, JavaScript, etc. Most, but not all, features are associated with their respective standard tags. For example, an image is always associated with the tag <IMG>.

This algorithm eliminates redundant blocks depending upon the inverse block document frequency (IBDF) of a block. The IBDF is inversely proportional to the number of documents in which the block occurs. The blocks that occur in multiple pages are redundant blocks and block which appears in one page is a content block.

To extract content block similarity between two blocks must be find out. For this block feature vectors of two blocks are used. These features are number of images, number of terms etc. If a feature is present in a block then its corresponding entry in the feature vector is one otherwise it is zero. Two blocks are identical if the similarity feature between two bocks is greater than a threshold value.[5]

2.6 Web Information Retrieval using page segmentation

A) Fixed-length approach (FixedPS)
B) DOM-based approach (DomPS)
C) Vision-based approach (VIPS)
D) Combined approach (CombPS)
All of these page segmentation methods are evaluated on the following
Two important techniques of information retrieval.
Block Retrieval – Block retrieval performs the selection of content blocks on the basis of ranking the documents with content blocks.
Query Expansion – For query expansion, expanded terms are extracted from relevant blocks, not the whole web pages.
These two techniques are based on the evaluation of web page segmentation on both data sets using both query sets. Each query set contains 50 queries and only the <title> field is used for retrieval. They have chosen Okapi as the retrieval system.
The block retrieval & query expansion techniques are conducted according to the following steps:

a. Initially list of ranked web pages is made using Okapi system.
b. Page partition is done to get blocks.
c. Now documents get replaced into blocks & pages are re-ranked then BR & DR are used to determine performance .They found that ComboPS is very close to performance.
d. Expansion term selection is made by selecting top-ranked blocks.
e. Final retrieval is collected by assigning weight to set of expanded query & to get final result it does comparison on the basis of query expansion..

They used web page segmentation method and chose top-ranked blocks to do query expansion a general conclusion can be made that partitioning pages into blocks can improve the performance of query expansion, regardless of which page segmentation method is used. Again, CombPS shows best performance.

3. CONCLUSION

There are various techniques used to find the content blocks from web pages & remove non content blocks efficiently.
So as to collect the content blocks from web pages it is beneficial to use entropy value based method since it also support web sites with different template styles. In case of k-maximum informative block mining it is using DOM tree, feature aggregation to select top-k nodes with maximal aggregated feature values as content blocks.
The compressed structure tree assigns weight to the features of node & justifies the commonality. It is also supporting the different nodes with different presentation styles by assigning high & low weight to the node by using entropy value.
CETR-Content Extraction via Tag Ratio deals with calculation of tag ratio of web page document linearly. It then selects contents by threshold value & trims non-tag chars. It also uses k-means clustering method to select content blocks. The another method Content Extractor is helpful to collect content blocks from web pages by determining feature vectors & to remove non content block using Get block routine & IBDF method. So both the method found beneficial to distinguish content blocks & non content blocks from web pages.
Using web page segmentation to web pages causes retrieval of content blocks by block retrieval method which does ranking of documents & query expansion which assigns weight to the blocks. But this technique is found feasible with web pages partitioned using CombPS (combined approach) because it shows better performance.

REFERENCES

[1] Lin, S. and Ho, J. 2002." *Discovering informative content blocks from Web documents*". In Proceedings of the Eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining
[2] Kao, H., Ho, J., and Chen, M. 2005. "*WISDOM: Web Intrapage Informative Structure Mining Based on Document Object Model*". IEEE Trans. on Knowl. and Data Eng. 17, 5 (May. 2005), 614-627.
[3] Yi, L. and Liu, B. 2003 *" Web page cleaning for web mining through feature weighting*". In Proceedings of the 18th international Joint Conference on Artificial intelligence .Publishers, San Francisco, CA, 43-48.
[4] Ziegler, C. and Skubacz, M. 2007. "*Content Extraction from News Pages Using Particle Swarm Optimization on Linguistic and Structural Features*". In Proceedings of the IEEE/WIC/ACM international Conference on Web intelligence(November 02 - 05, 2007). Web Intelligence. IEEE Computer Society, Washington, DC, 242-249.
[5] Debnath, S., Mitra, P., Pal, N., and Giles, C. L. 2005. *"Automatic Identification of Informative Sections of Web Pages".* IEEE Trans. on Knowl. and Data Eng. 17, 9 (Sep. 2005), 1233-1246.
[6] Cai, D., Yu, S., Wen, J., and Ma, W. 2004. "*Block-based web search*". In Proceedings of the 27th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Sheffield, United Kingdom, July 25 - 29, 2004). SIGIR '04. ACM, New York, NY, 456-463.