Extraction of Content Blocks from Web Pages

Manisha Marathe¹, S.H.Patil², M.S.Bewoor ³
¹ Lecturer department of Computer Engg, PVG's College of Engineering & Technology, Pune,
² Professor and Head, Computer Engg Bharati Vidyapeeth College of Engineering, Pune
³ Assistant Professor of Computer Eng, Bharati Vidyapeeth College of Engineering, Pune
mvnm_comp@pvgcoet.ac.in, shpatil@bvucoop.edu.in, msbewoor@bvucoop.edu.in

Abstract
Web pages often contain several items that cannot be classified as the primary content. Most end-users search for the primary content and largely do not look for the non-informative content. Non-content blocks are navigation sidebars, advertisements, search blocks, copyright notices, etc. A tool that assists an end-user to search and process information from web pages automatically, must separate the primary content sections from the other non-content sections. These sections are called as web page blocks or just blocks. A tool must first segment the web pages into web page blocks. Then the tool must separate the primary content blocks from the non-informative content blocks. A algorithm [2] is defined to partition an HTML page into constituent web page blocks. Algorithms identify primary content blocks by looking for blocks that do not occur in a large number of web pages and by looking for blocks with desired features. A system that applies these algorithms to remove non-informative content blocks and to identify similar blocks across web pages can achieve significant storage savings and timing saving.

Keywords: Data Mining, web page block, information block, inverse block document frequency

1.0 INTRODUCTION
The rapid expansion of the Internet has made the World Wide Web a popular place for disseminating and collecting information. The innovation of the Web creates numerous information sources published as HTML pages on the Internet. Search engines crawl the World Wide Web to collect web pages. These pages are stored and indexed. The user who is performing a search using search engine is interested in primary informative content of the web page. Data mining on the Web thus becomes an important task for discovering useful information from the Web. But a large part of these Web pages is content that can not be classified as the primary informative content of the Web page. Useful information on the Web is often accompanied by advertisements, image-maps, plug-ins, logos, search boxes, category information, navigational links, related links, footers and headers, and copyright information. Although such information items are useful for human viewers and necessary for the Web site owners, they often hamper automated information gathering and Web data mining. These blocks are not relevant to the main content of the page. Such blocks are referred to as non-content blocks. These blocks are very common in web pages.

The content from a Web page can be used, after subdividing the page into smaller semantically homogeneous sections based on their content. These sections are referred as blocks or web page blocks. To design any web page different HTML tags are used. Actual data is enclosed within a pair of open and a close tag. A web page block B is a portion of a Web page enclosed within an open-tag and its matching close-tag. These open and close tags belong to an ordered tag-set T that includes tags like <TABLE>, <TR>, <P>, <HR>, and <UL>. The advantage of identifying blocks from web pages is that if user does not want non-content block 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 cache and indexing.

2.0 RELATED WORK
Yi and Liu[3] have proposed an algorithm for finding out non-content blocks from web pages using style trees. Non-content blocks are referred as noisy blocks by them. Lin and Ho [1] identifies content block by dividing web page into different blocks. For this they have used <table> as a main tag. This method is similar to the method discussed in this paper, but in the method discussed in this paper other tags apart from table are also considered. Kushmerick [4] has proposed a feature based method which identifies internet advertisements in a web page. Ramaswamy[5] propose a Shingling algorithm to identify fragments of web pages and use it to show that the storage requirements of web caching are significantly reduced.

3.0 SYSTEM DESCRIPTION
A table is used for designing maximum web page. An HTML table is defined using the tag <TABLE>. In a table occurring in a Web page, each cell is considered as a block. Where tables are not available, identifying blocks involves partitioning a Web page into sections that are coherent, and that have specific functions. For example, a block with links for navigation is a navigation block. Another example is an advertising block that contains one or more advertisements. Usually, a navigation block is found on the left side of a Web page. Typically, the primary informative content block is laid out to the right of a Web page.

Similar blocks across different Web pages obtained from different Web sites can be identified. For example, a search on Google News on almost any topic returns several syndicated articles. Popular items like syndicated columns or news articles written by global news agencies appear in tens of newspapers. Ideally, the user wants only one of these several copies of articles. Since the different copies of the article are
from different newspapers Web sites, they have similar content blocks while they differ in their non-content blocks only. Separating and indexing only the content blocks it can be easily identified if two Web pages have identical content blocks. This will save on storage and indexing by saving only one copy of the block. This will make search results better by returning more unique articles. Even search times improve because we have less data to search.

Content block can be identified based on the appearance of the same block in multiple Web pages or based on the occurrence of certain features to identify content blocks. First, the algorithms partition the Web page into blocks based on heuristics. These heuristics are based on previous study of HTML editing style. A Web page is divided into blocks on the basis of HTML tables as well as some other tags. Second, the algorithms classify each block as either a content block or a non-content block. While the algorithm decides whether a block, B, is content or not, it also compares B with stored blocks to determine whether B is similar to a stored block. If the block is same as the block stored then it is not necessary to store that again.

4.0 OVERVIEW OF PROCESS

To find primary content block from web pages first web page is divided into blocks and then primary content blocks are identified. Also duplicate blocks from different web sites are checked and only one copy of a block is stored. Two algorithms are used for this, content-extractor and feature-extractor, as explained below. The input to the algorithms is a set of web pages belonging to a class of web pages. A class is defined as a set of web pages from the same web site whose designs or structural contents are very similar. A set of web pages dynamically generated from the same script also belong to the same class. The output of the algorithms is the primary content blocks in the given class of web pages.

4.1: Content-Extractor algorithm

Content-Extractor algorithm is shown in algorithm 1. The first step of algorithms is to use the GetBlockSet routine to partition each page into blocks. The GetBlockSet routine takes an HTML page as input with the ordered tag-set.

GetBlocks function takes a full document or a part of a document, written in HTML, and a tag as its input. It partitions the document into blocks according to the input tag. Fig. 1 shows the structure of two HTML pages and shows the blocks that are identified for these pages. This algorithm eliminates 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.

The function Sim (algorithm 3) returns the cosine between block feature vectors of two blocks. 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 blocks is greater than a threshold value.

4.2 Feature-Extractor algorithm

Content-Extractor algorithm is shown in algorithm 3. This algorithm will identify blocks with a set of desired features. A web page block can have any or all of the features of an HTML page. E.g. text, text tag, list.

This algorithm does not depend on multiple pages, it depends upon the feature set and the chosen feature for output. Suppose the chosen feature is text. The algorithm calculates a value for each feature in each block. If a block contains 1000 words, 2 images, 3 links and a applet, and the maximum values of words, images, links and applets contained in blocks in the data-set are 2000, 4, 50 and 3 respectively. Then the values for the features in the given block are 1000/2000, 2/4, 50/3 and 1/3 respectively. If the sum of the feature values of the desired features is greater than the sum of the feature values of the rest of the features then store that block. Then feature values of this new set of blocks are calculated and one with the highest sum of values of desired feature is chosen.
5.0 CONCLUSION

Content-Extractor, Feature-Extractor is simple but powerful algorithms to detect primary content blocks from web pages. Content-Extractor can detect primary content block by checking presence of a block in multiple pages. Feature-Extractor can identify primary content block if there is a dominant feature. These algorithms reduce storage requirements; provide fast search and unique articles.

REFERENCES