Abstract
The paper is about a novel algorithm, DSE (Data-rich Subtree Extraction) to recognize and extract the data-rich section of an HTML page. The DSE algorithm is used for two typical web information retrieval problems: topic distillation and web information extraction. The DSE algorithm has been developed by Jiying Wang from the University of Science & Technology in Hong Kong.

Introduction
Many Internet sites, especially commercial web sites, try to provide as much information and convenience as possible. As a consequence web page designers usually present their data along with some “decoration” in the pages, such as navigational panels and advertisement bars. Deciding which part of this web page contains the main content might be very easy for humans, but is difficult for computer programs. Fortunately, as the web evolves, web page creation changes from a mainly manual process to a more dynamic procedure using complex templates. Many web pages are no longer created in advance, but are generated on the fly by querying a database server and sending the results to a predefined page structure.

How the algorithm works...
Given a page, we refer to it as the target page and choose one of its outgoing links to explore sample pages in the same site according to the similarity between their URLs. By comparing a sample page to the target page, substructures commonly appearing in the two pages are identified and the non data-rich structures are removed. The remaining structures contain all the data (they are the data-rich sections).

HITS algorithm
The HITS algorithm is one of the most well-known topic distillation algorithms. Given a set of a web pages about one scientific topic, the HITS algorithm
calculates the autohirity score (which is an indication for relevant content) and the hub score (an indication for relevant links) for each page.

\[ a(p) = \sum_{q \rightarrow p} h(q) \quad h(q) = \sum_{q \rightarrow p} a(p) \]

The HITS algorithm does not handle the cases with repeated links in which a person's opinion about the page will be overconsidered.

**The DSE Algorithm**

The DSE Algorithm can be split up in three main phases:

- First, the algorithm tries to discover a set of pages as sample pages that share a common display layout with the target page. The algorithm takes the outgoing links of the target page as the source pool to search for similar pages.

- Second, the HTML pages are being parsed and converted into tag-trees, that can be analysed and manipulated easily.

- Third, the target page tree is compared with the sample page tree to identify their common parts. Nodes appearing in their shared sub-structures are identified and removed and the remainder is the data-rich subtree.

**Discovering sample URLs**

In this phase, our goal is to explore "good" sample pages for the target page, in order to correctly identify its data-rich sections by matching their structures. Goodness here means that the two pages have similar layout. This shows a big problem, because pages from the web can't be randomly chosen. Usually, pages on the same web-site have similar layout. They are often produced by the same content-management-system. Most of the time web-sites have links that show from one page to another. This helps us find similar pages.

The algorithm is using a function called US (URL similarity) to identify the similarity of two web-pages, which works very well. But it does not work in all cases. Some companies register two different domain names with identical content and inter-connect pages across two domains. For example
http://theeventguide.com/ and http://fort.lauderdale.eventguide.com come from the same company. Page layouts are alike, but have very few or even no links pointing back to their own site. Therefore the algorithm is extended in order to handle these problems.

**Tree creation**
After downloading the target page and the sample page, they are parsed and a data structure is employed to represent their layout, because in the next phase an comparison algorithm is run.

As HTML pages are composed of tags and text enclosed by tags, it is possible to represent an HTML page's layout by a tree-like structure referred to as the Document Object Model (DOM).

All the font tags (like FONT, SMALL) and title tags (H1, H5. Etc) are eliminated in the tree. Some HTML tags have attributes (for example The BODY tag has attributes BACKGROUND for background images). Therefore attributes other than the HREF of A and SRC of IMG tags will be ignored when matching the trees.

**Tree Matching**
Given two DOM trees representing the target page and the sample page built in phase 2, the next step is to match the “similar” structures of the trees. The basic idea in phase 3 of the DSE algorithm is to traverse these two trees using a depth-first order and compare them node-by-node from the root to the leaves.
It is worth mentioning that HTML pages vary so much that even two pages that look the same in a browser can have different tag structures. For example, `<TABLE><A>...</A></TABLE>` and `<TABLE><TABLE><A>...</A></TABLE></TABLE>` result in different subtrees, but they look the same in a browser. The algorithm does not consider such cases. However, since the basic assumption of the DSE algorithm is that the same template usually generates HTML pages from the same web site, therefore, their tree structure should be the same.

**Experiments**

The researchers applied the DSE to the HITS algorithm and to the IEPAD algorithm. Both experiments show that the use of the DSE algorithm improved the performance of the original algorithms.

**Conclusion**

Web page authors usually follow some conventions to generate web pages so that the displayed layout is more convenient for users to locate info. Therefore, based on the assumption that documents from the same web-site have similar structure, the DSE algorithm was developed to identify data-rich section of HTML pages. The experiments show that the DSE algorithm accurately detects the data-rich sections of web pages and improves the performance of the HITS algorithm.