International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 5(2), September - October 2015 ISSN 2278-6856

A review on Web page change detection using document tree based method

Annu Saini[1], Taruna Kumari[2]

[1]M.Tech (C.S), AIM & ACT Department, Banasthali University, Rajasthan

[2] M.Tech (C.S), YMCA University of Science and technology, Faridabad

Abstract

There are many documents available online and among them, most of the documents are dynamic. Due to the dynamic nature, they are changing there, contents continuously so there is a need to efficiently identify the changes in them. So there is a need to compare the old page with the new one. Four kinds of changes can occur in a web page. They are Structural changes, Content changes, Presentation changes and Behavioral changes. A number of algorithms are available to identify these changes.

In this paper, these various kinds of changes are explained and among them structural changes in a web document using document tree based structural change detection method is presented in detail.

Keywords: web document, change detection, HTML tags, parser, and document tree.

1. INTRODUCTION

In today's world internet has become a largest source of information. Internet is the effective way of exchanging information among two parties placed in any corner of the world. Tremendous amount of data sources are available online. And these data sources provide information mainly in the form of web pages. Most of the documents available online are dynamic in nature means there contents changes continuously. It is quite possible that after downloading web pages, their local copies residing in the database of a search engine become obsolete compared to the copy on the web server end. Therefore, need arises to refresh the web pages of database at regular interval. If a user is observing a web page that is refreshed very frequently then the user might not be interested in downloading or viewing the entire web page each and every time. He will be interested in observing the changes that have taken place in the web page since last visit. To identify the changes we have to compare the old page from the database with the new page. The user may be interested only to know the changes in the web pages every time he visits the same page. Once it is decided to update the document, it should be ensured that minimal resources are used in the updating process. Updating only those documents of the database, which have undergone actual changes, require a small amount of data processing as compared to the huge size of the web [1].

In a web collection a user can find information on every topic and due to dynamic nature of web pages the pages in the collection are added, removed, modified. All these changes can be reflected in a short span of time. For example: a stock trader may only be interested in knowing the current status of market or new prices of the stocks. Similarly a news website user is only interested in the current news. All these types of web pages are updated in very small interval of time .The web page change detection system helps the user to detect such changes efficiently and in minimum browsing time. In order to make optimum use of this changing collection, different systems have been conceptualized with the goal of bringing to the web users a simplified view of the Web and efficiently dealing with the changes on the Web.

2. CLASSIFICATION OF CHANGES IN A WEB PAGE

Change in a web page can be classified in following four categories [2]:

- Structural changes
- Content or semantic changes
- Presentation changes
- Behavioral changes

2.1. Structural Changes

In structural changes a web page is changed with respect to its structure by addition or deletion of tags. Sometimes the addition/deletion/modification of a link also causes a structural change. These types of changes are important to detect, as sometimes they often might not be visually perceptible [2].

```
. <html>
<body>
<p><b>
----------------------  ----------
</b> </p>
<p><big>
-----------------------------------
</big></p>
<p><i>
-----------------------------------
<ul><li>----------------------</li>
<li> ----------------------</li></ul>
<u>----------------------------</u>
</i></p>
</body>
</html>
```

(a) Initial Version

```
<html>
<body>
<p><b><font ---------->
---------- ----------
</font> </b> </p>
<p><big>
----------
</big></p>
<p><b>
----------
----------
----------
</b></p>
</p>
</body>
</html>
```

(b) Changed Version

Figure 2.1 structural change in web page

2.2. Content or semantic changes

A web page is changed with respect to its content means the content in the web page changes time to time from reader's point of view. For example: a news website changes its contents when fresh news arises [2].

```
<html> <head> <title> this is page title
</title> </head>
<body> <center>Times of India News 5-
Oct-2015</center>
<ul>
<li>Germany ready to support "Make in
India".
<li>Bulk of black money within India: Jaitley
<li>Utter Pradesh starts online registration
of hindu marriages
</ul>
</body></html>
```

```
<html> <head> <title> this is page title </title>
</head>
<body><center>Times of India News 5-Oct-
2015</center>
<ul>
<li>Markets rush full steam; sensex up over 500
points
<li> Bulk of black money within India: Jaitley
<li> Germany ready to support "Make in India".
</ul>
</body></html>
```

(b)Changed Version

Figure 2.2 Content changes in a web page

2.1. Presentation changes

In this category only the appearance of a web document changes and the contents remain unchanged .means by changing the HTML tags the appearance of a web page can be changed [2].

```
<html>
<body>
<p>India-Lanka 1st Test ends in a draw </p>
<p>Maya asked to increase Rahul's security
</p>
<p> Ties with China not at India's expense:
US
</body>
</html>
```

(a) Initial Version

```
<html> <body>
<p style="background-
color:#00FF00"><u>India-Lanka 1st
Test ends in a draw</u></p>
<p style="background-color:
rgb(255,255,0)"><u>Maya asked to
increase Rahul's security</u></p>
<p style="background-
color:yellow"><u>Ties with China not
at India's expense: US
</u></p>
</body> </html>
```

(b) Changed Version

Figure 2.3 Presentation Changes in a web page

2.4Behavioral changes

Behavioral changes refer to modifications to the active components present in a document. For example, web pages may contain scripts, applets etc as active components. When such hidden components change, the behavior of the document gets changed [2].

So, in order to make search engines up-to-date it is mandatory to detect the changes that take place in web documents. Further in this paper we will discuss the structural changes in a web document.

3.METHODS TO DETECT STRUCTURAL CHANGES

There are a number of methods to detect structural changes. Here document tree based method for the detection of structural changes in a web document [3][4][5] is discussed in detail.

Document tree based structural change detection method

This method works in two steps. In the first step document tree is generated for the downloaded web page while in the second step, level by level comparison between the trees is performed.

The downloaded page is stored in the database in search/insert fashion as given below [3]:

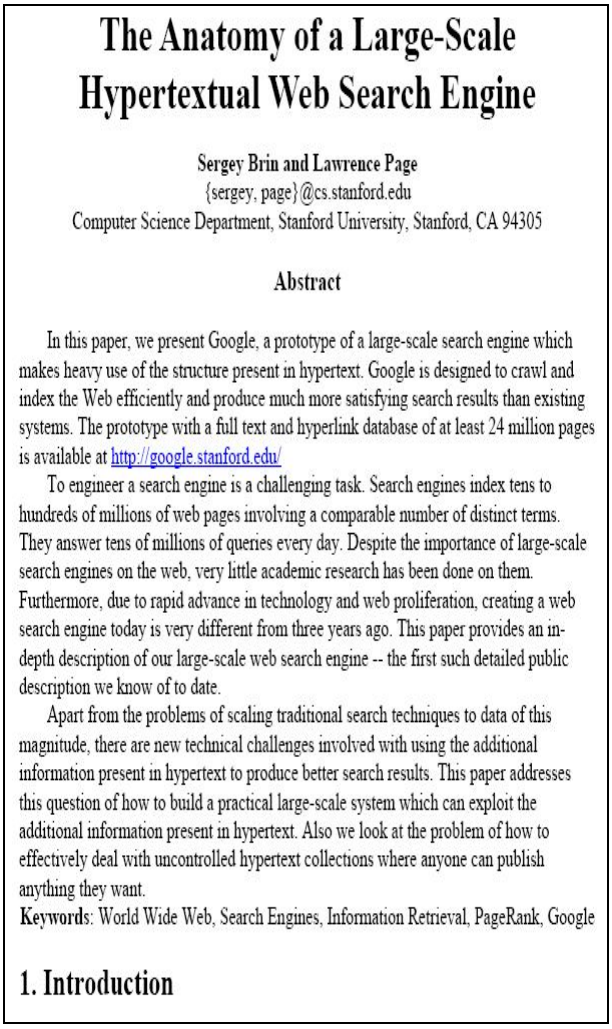
Step 1: Search the downloaded document in the database.

Step 2: If the document is found then compare both the versions of the document for structural changes using level by level comparison of their respective document tree.

Step 3: Else store the document in the database.

After parsing the downloaded web page, parser extracts all the tags present in it. Then tags are arranged in the order of their hierarchical relationship and finally document tree is generated with the help of tree generator. All the tags which are nested at the same level in the web page should be at the same level in document tree too. The structure of document tree contains following fields [3]-[5]:

- **Tag_name:** This field of node structure stores the name of HTML tags.
- **Child:** This field contains information about the children of each node.
- **Level_no:** This field contains the level number at which the nodes appear in the constructed document tree.
- **No_of_siblings:** This field in a node structure contains total number of nodes present at that level.



The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page
{sergey, page}@cs.stanford.edu
Computer Science Department, Stanford University, Stanford, CA 94305

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/

To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago. This paper provides an in-depth description of our large-scale web search engine -- the first such detailed public description we know of to date.

Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. This paper addresses this question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want.

Keywords: World Wide Web, Search Engines, Information Retrieval, PageRank, Google

1. Introduction

Figure 3.1 Initial version of web document

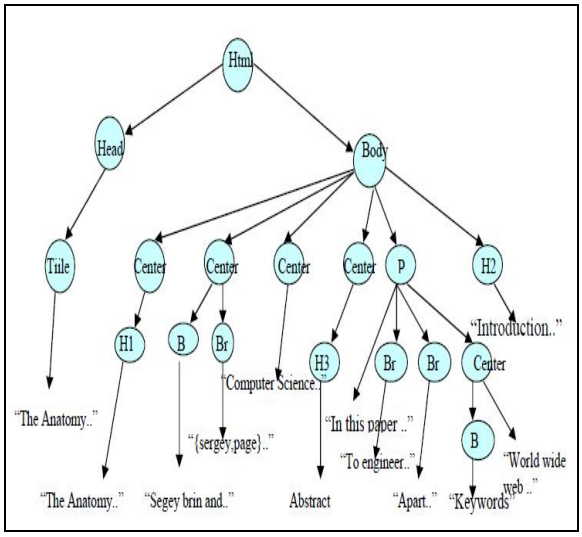
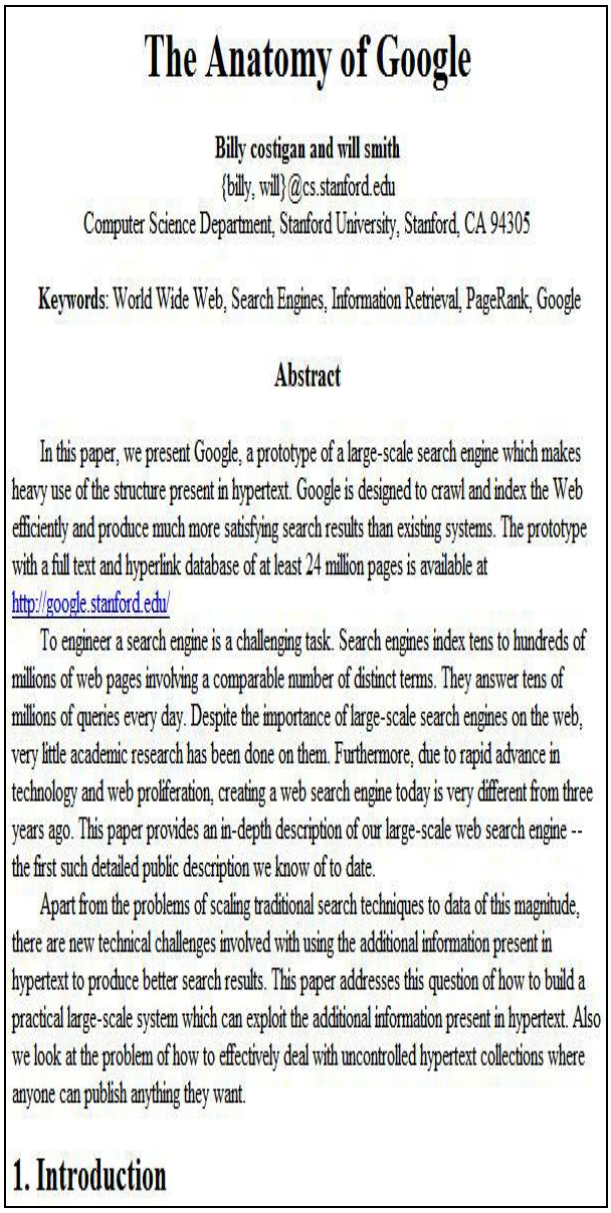


Figure 3.2 Document tree for initial version



The Anatomy of Google

Billy costigan and will smith
{billy, will}@cs.stanford.edu
Computer Science Department, Stanford University, Stanford, CA 94305

Keywords: World Wide Web, Search Engines, Information Retrieval, PageRank, Google

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/

To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago. This paper provides an in-depth description of our large-scale web search engine -- the first such detailed public description we know of to date.

Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. This paper addresses this question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want.

1. Introduction

Figure 3.3 Changed version of web document

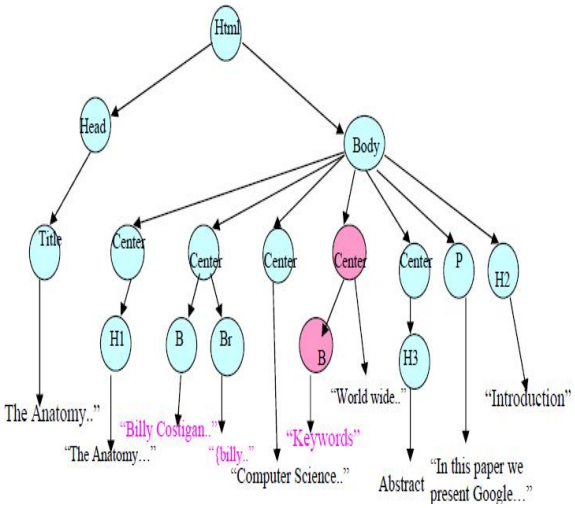


Figure 3.4 Document tree for changed version

```
<html><head><title>----------
</title></head><body>
<center><h1>----------
</h1></center>
<center><b>----------
</b></center>
<center>----------</center>
<center> <h3>----------</h3>
</center>
<p>----------<br>--------
-
<br>----------
<center><b>----------</b>
----------
</center>
</p>
<h2>----------</h2>
</body></html>
```

Figure 3.5 HTML tag structure of initial version

```
<html><head><title>----------
</title></head><body>
<center><h1>----------
</h1>/center>
<center><b>----------
</b></center>
<center>----------<br>------
</center>
<center><b>----------</b>
</center>
<center><h3>----------</h3>
</center>
<P>----------</P>
<h2>----------/h2>
</body></html>
```

Figure 3.6 HTML tag structure of changed version

Table 3.1: Node wise attribute details of the initial and the modified tree using level order traversing

Attributes	Initial version	Changed version
Level_no	1, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5	1, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4
Tag_name	html, head, body, title, center, center, center, center, p, h2, h1,b, br, h3, br, br, center, b	html, head, body, title, center, center, center, center, p, h2, h1, b, br, b, h3
Child_no	2, 1, 6, null, 1, 2, null, 1, 3, null, null, null, null, null, null, null,1, null	2, 1, 7, null, 1, 2, null, 1, 1, null, null, null, null, null, null, null
No_of_Siblings	1, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 1	1, 2, 2, 8, 8, 8, 8, 8, 8, 8, 8, 5, 5, 5, 5, 5

We can draw the following inferences on the structural changes:

- When no sibling added/deleted at any level it means there is no structural change between two versions of the web document.
- When leaf siblings added/deleted it means there are minor changes between the two versions of the web document which may be non-noticeable sometimes.
- When siblings having children added/deleted it means major changes have occurred and these changes are used to detect the structural changes in two versions of web documents.

4. CONCLUSIONS

Although the document tree based structural change detection method is efficient to detect the structural changes perfectly if the constructed tree represents the true hierarchical relationship among tags. While constructing the tree problems arises in presence of optional tags or if the tags are misaligned, it becomes difficult for document tree constructor to maintain the correct hierarchical relationship among the tags while constructing the tree. If nesting of tag structures is misaligned then it becomes very difficult to handle the situation while constructing the document tree

References

[1] S.Goel, R.R Aggarwal "An Efficient Algorithm for Web Page Change Detection "International journal of computer applications (0975 – 888) Volume 48– No.10, June 2012.

[2] Yadav D. 2009"Design of A Novel Incremental Parallel web crawler" PhD thesis, Jaypee Institute of Information Technology University, 2009.

[3] Available at shodhganga.inflibnet.ac.in/bitstream/10603/.../14_chapter%205.pdf

[4] N.K.Varshney, D. K. Sharma "A Novel Architecture and Algorithm for Web Page Change Detection.

[5] S. Flesca, E. Masciari " Efficient and effective Web change detection" Data & Knowledge Engineering 46 (2003) 203–224