Deep Web Data Extraction Using Visual Features

Harshali P. Patil,J. W. Bakal, Sharvari Govilkar
(Department of Computer Science, Thakur college of engineering and technology,Mumbai University, India Email: harshali.patil9@gmail.com)
(Principal,Shivajirao S. Jondhale College of Engineering , Mumbai University, India,Email: bakaljw@gmail.com)
(Department of Computer Science, pillai's institute of information technology,engineering, media studies & research,Mumbai University, India,Email: g_sharvari@rediffmail.com)

ABSTRACT

World Wide Web has huge online useful Web databases which are difficult to extract relevant data from various sources. The number of Web databases has reached 25 millions according to a recent survey. Web databases can be searched through their Web query interfaces and the results are wrapped in dynamically generated web pages which are called as deep web pages. However, due to the heterogeneity and the lack of structure of Web information sources, access to this huge collection of information has been limited to browsing and searching. Searching for personal information is sometimes even more complicated. Below are several common problems we face when trying to get personal details from the web Information is distributed between different sites. There is no uniform format for personal information. It is not updated. Name ambiguity, two or more people with the same name. Our goal is to build an easy, fast and reliable tool that will find personal information on the web, and present it in a friendly, easy-to-read format. The output of our application will be a HTML page, which looks like personal web-pages which are found in the internet. In our paper we focus on searching for personal information of scientists and researchers,because these people have more relevant information in the web.The method we use in order to extract the information based on publications.

KeyWords— **Deep web data, Google API, Information Extraction, Search Engine, Yahoo API**

1. INTRODUCTION

The rapid expansion of the web is causing the constant growth of information, leading to several problems such as increased difficulty of extracting potentially useful knowledge. Deep Web, as a rich and largely unexplored data source, is becoming nowadays an important research topic. In the paper we propose different approach based on deep web data extraction.

Finding information about people in the World Wide Web is one of the most common activities of Internet users. Person names, however, are highly ambiguous. Person names, however, are highly ambiguous. In most cases, the results for a person name search are a mix of pages about different people sharing the same name. The user is then forced either to add terms to the query (probably losing recall and focusing on one single aspect of the person), or to browse every document in order to filter the information about the person he/she is actually looking for. In an ideal system the user would simply type a person name, and receive search results clustered according to the different people sharing that name. One particular case of this people-document association task is referred to as personal name resolution. The task is as follows: given a set of documents all of which refer to a particular person name but not necessarily a single individual (usually called referent), identify which documents are associated with each referent by that name. Different methods have been used to represent documents that mention a candidate, including snippets, text around the person name, entire documents, extracted phrases, etc.

The project is basically a research work in the field of search engines. Attempt will be made to design a much better system to overcome the flaws of the existing system. But due to time and resource constraints, the research work will be more focused for IEEE documents only. However, our algorithm should not be limited for IEEE documents and can use other papers. The resultant of this is customized advance search and download engine.

This paper is structured as follows. In section 2, we describe what is the objective of the system, in section 3 we discuss about the existing system for this function. In section 4 we present a new concept which is used for deep web data extraction. In section 5 we discuss some output of related work and finally we conclude in section 6. We

conclude with the resultant, customized advance search and download deep web data.

2. OBJECTIVE

Web Search machines are absolutely not new concepts, but typically they've included strict terms of service regarding the rearranging and presentation of results, and provided little or no opportunity for monetization. These constraints have limited the innovation and commercial viability of new search solutions. The information originates from multiple sites, which sometimes are not updated and due to that ,Maximum of the information mining is time consuming as Searching for two people with same name adds to complexity while search.

3. EXISTING SYSTEM AND ITS EFFECT

Most of the Information is spread over the different sites.It is not updated. Two or more people with the same name called as Multi-Referent ambiguity.Multi-morphed ambiguity which is because one name may be referred to in different forms. In the most popular search engine Google, one can set the target name and based on the extremely limited facilities to narrow down the search, still the user has 100% feasilibility of receiving irrelevant information in the output search hits. Not only this, the user has to manually see, open, and then download their respective file which is extremely time consuming. The major reason behind this is that there is no uniform format for personal information.Maximum of the past work is based on exploiting the link structure of the pages on the web, with hypothesis that web pages belonging to the same person are more likely to be linked together.

4. PROPOSED SYSTEM

Our project can be easily presented as a chain with 4 separate sections. However, it is only an abstract description. In fact, these stages are distributed differently between the classes. Stage 1st is the user input. User inserts the target name. We don't have any reliable method to identify personal details in a text. The solution for this problem and perhaps the idea behind our project is to get the details from scientific publications.

The structured layout of these articles helps us to detect the information. HomePage is Java Desktop application for Windows XP/ Windows Vista and requires internet connection. The input for the search process is the "target" name. It can be a combination of first name and last name, or the last name alone (in this case there is a greater chance for name ambiguity).

In the first develelopment cycle we focus on finding the personal details like E-mails, Publication titles, Publication short description (abstract), Links to the full document in the web and Pictures.

In the end of the search process a HTML page is built automatically containing all the above mentioned. The default system browser is opened with the generated HTML page. A report with the search statistics is shown to the user.

One of the key challenges that needs to be overcome to make the project functionality a reality, is to build an advance query system that is capable of reaching high disambiguation quality. The project work is targeted to design an advance version of the search engine using Yahoo Search BOSS and Clustering Algorithm. In this research work, the focus is mainly on searching for personal information of scientists and researchers. The user has to set the proper target name for search, which when completed, the user will receive complete PDF and image files based on the key (e-mail) of the search. Each group of information items (cluster) will be defined by its key (email) and the user make the choice. The result page will be produced from the chosen clusters For making the search operationally accurate, we will assume the usage of IEEE doc files as they carry a standard format of name, e-mail ID, publication, images, and links to the full images.

This project is based on Yahoo! Search BOSS (Build your Own Search Service). BOSS is a Yahoo! Developer Network initiative to provide an open search web services platform. The main goal and idea of BOSS is to give developer's free access to the Yahoo! Search index. The results can be supplied into the developer's website or program so that they can manipulate the resources according to their product's requirements. BOSS will supply the internet search that we need for our application.

5. STRATERGIES

The project work is broadly classified to 4-different parts for the development purpose as shown in Figure 2 above.

1. First, we need to find and download PDF files suspicious in being articles containing the target name using crawlers.
2. second, the indexer will extract the text from the PDF document, filter documents that not seem to be relevant papers
3. Third, 5 clusters with the biggest amount of related document are presented to the user.
4. Fourth, the page builder also shows a summary screen of the search process.

5.1 Crawler

We need to find and download PDF files suspicious in being articles containing the target name. The search is done with BOSS API that provides the possibility search for PDF documents only. The Crawler will get the URLs from BOSS's reply (which contains many other fields), and download the documents to the local disc. The application will work with configurable "maximum" of documents. The default is 50 (i. e. maximum 50 PDF documents will be downloaded). The crawler performs second search for finding image results for target name. First N pictures will be taken to the indexer (N is configurable).

A web crawler is a program or an automated script which browses the World Wide Web in a methodical automated manner. A Web crawler also known as a web spiders, web robots, worms, walkers and wanderers are almost as old as the web itself. The first crawler, Matthew Gray's wanderer, was written in spring of 1993, roughly coinciding with the first release of NCSA Mosaic. Due to the explosion of the web, web crawlers are an essential component of all search engines and are increasingly becoming important in data mining and other indexing applications. Following is the process by which Web crawlers work:
1.Download the Web page.
2. Parse through the downloaded page and retrieve all the links.
3. For each link retrieved, repeat the process. The Web crawler can be used for crawling through a whole site on the Inter/Intranet. You specify a start-URL and the Crawler follows all links found in that HTML page. This usually leads to more links, which will be followed again, and so on.

5.2 Indexer

The indexer will extract the text from the PDF document, filter documents that not seem to be relevant papers. Indexer will analyze the text and find the suitable information pieces. As to the images, the indexer will download them to the local disc. Another problem is avoiding ambiguous names. In order to solve this problem we will need user interface. Indexer will divide the collected information to "clusters".
Cluster will be identified by information pieces types that were defined as the key. In this version the key info is the email address. Each cluster will contain the key e-mail and the rest of the information pieces found in the same documents as the key.

5.3 Cluster Module

Information from the documents is inserted into the clusters according to the emails(keys) in the source document.5 clusters with the biggest amount of related document are presented to the user. The user will choose the clusters and that are likely to belong to the target person. The chosen clusters are passed to the page builder. If the indexer produced only one cluster – the application will skip this step. In case that no clusters were produced – message will be shown to the user.

5.4 Page Builder

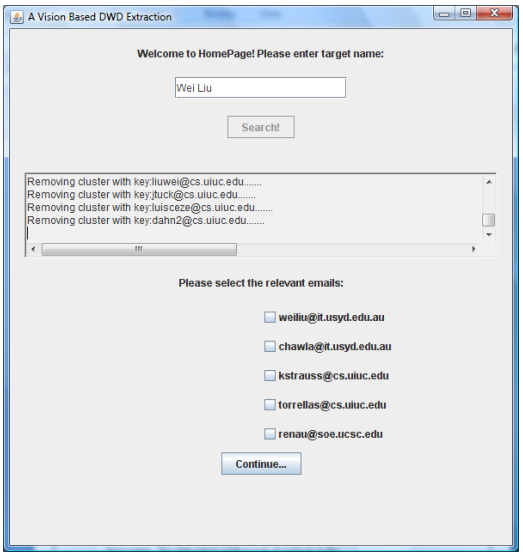
Gets the clusters and images and creates HTML page containing all the information from the clusters. The sections in the HTML page are:

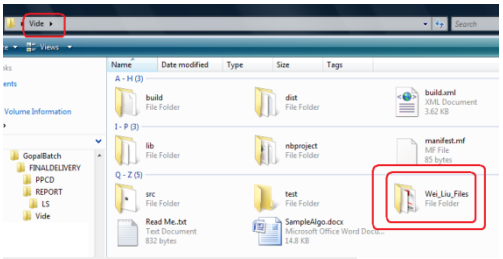
Header part – notification ("This page was automatically generated etc.), the target name.
Images – up to N images which were found and can be successfully shown; path to the images in the local disk.

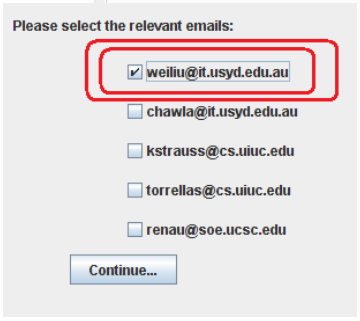
Publications – title; abstract; URL; local path.

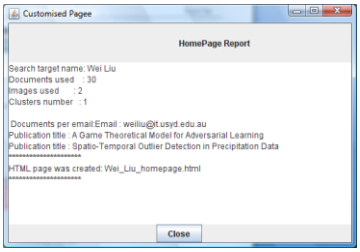
Contacts – emails: When the page is ready, the default system browser is opened, and the page can be visible. The page builder also shows a summary screen of the search process. The summary screen contains the number of the documents and images, the number of clusters and so on.

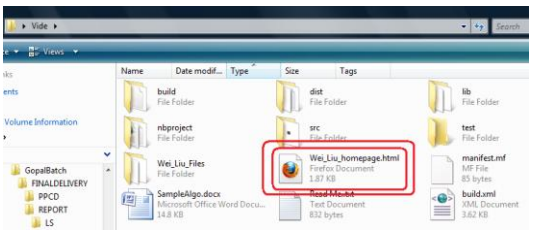
5.5 Output

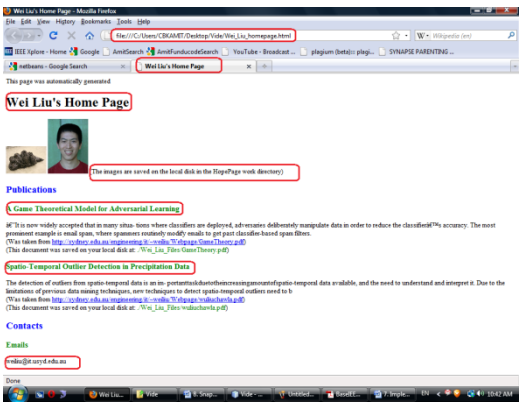












6. Conclusion

The project is basically a research work in the field of search –engines. Attempt will be made to design a much better system to overcome the flaws of the existing system. But due to time and resource constraints, the research work will be more focused for IEEE documents only. However, our algorithm should not be limited for IEEE documents and can use other papers. The resultant of this is customized advance search and download engine.

The main goal of the work is to build an easy, fast and reliable tool that will find personal information on the web, and present it in a friendly, easy-to-read format. The output of our application will be a HTML page, which looks like personal web-pages which are found in the internet.

The World Wide Web is a rapidly growing and changing information source.Due to the dynamic nature of the Web, it becomes harder to find relevant and recent information.. We present a new model and architecture of the Web Crawler using multiple HTTP connections to WWW.

The multiple HTTP connection is implemented using multiple threads and asynchronous downloader module so that the overall downloading process is optimized. The user specifies the start URL from the GUI provided. It starts with a URL to visit.

As the crawler visits the URL, it identifies all the hyperlinks in the web page and adds them to the list of URLs to visit, called the crawl frontier.

URLs from the frontier are recursively visited and it stops when it reaches more than five level from every home pages of the websites visited and it is concluded that it is not necessary to go deeper than five levels from the home page to capture most of the pages actually visited by the people while trying to retrieve information from the internet.

The web crawler system is designed to be deployed on a client computer, rather than on mainframe servers which require a complex management of resources, still providing the same information data to a search engine as other crawlers do

REFERENCES

[1] M. Kovacevic, M. Diligenti, M. Gori, and V. Milutinovic. Recognition of common areas in a web page using visual information: a possible application in a page classification. In Proceedings of ICDM-2002.

[2] Javier Artiles, Satoshi Sekine, Julio Gonzalo, Web People Search - Results of the first evaluation and the plan for the second – ACM portal, April 21-25, 2008 · Beijing, China

[3] Javier Artiles, Julio Gonzalo, Satoshi Sekine, The SemEval-2007WePS Evaluation: Establishing a benchmark for the Web People

Search Task, Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 64–69

[4] Ron Bekkerman, Andrew McCallum, Disambiguating Web Appearances of People in a Social Network, International World Wide Web Conference Committee (IW3C2), 2005

[5] Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka, Measuring Semantic Similarity between Words Using Web Search Engines, Copyright is held by the International World Wide Web Conference Committee (IW3C2)., 2007

[6] Nguyen Bach & Simon Fung, Co-reference Resolution for Person Names,

[7] Dmitri V. Kalashnikov, Rabia Nuray-Turan, Sharad Mehrotra, Towards Breaking the Quality Curse. A Web-Querying Approach to Web People Search, ACM-2008

[8] Krisztian Balog, Leif Azzopardi, Maarten de Rijke, Personal Name Resolution of Web People Search, NLPIX2008, April 22, 2008, Beijing, China