KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 81–88

ENHANCING THE INVISIBLE WEB

LUCIAN HANCU[(1)]

Abstract. In recent years, a large amount of information has been placed in databases across the globe, and published through dynamically generated Web pages. The evolution of the so-called Invisible (or Hidden) Web constitutes both an opportunity and an issue for Web-based information extractors. This article describes the architecture of an Invisible-Web Extractor, whose primal goal is to enhance the value of the hidden Web data. We consider three main issues of the tool: how to access the Invisible Web information, how to extract information from the gathered data and how to create new knowledge from it.

1. Introduction

During the last decade, the Web has become a primal source of information, which exhibits various forms of content: personal or business Web pages, news aggregators, large collections of music or videos. The more its content evolves and varries, the most difficult becomes the design and implementation of automatic tools that discover it, in order to index it and make that content available by use of search interfaces or to extract page snippets with the purpose of creating a more valuable Web material.

In [10], the authors classify the various portions of the Web, by considering two dimensions: whether the pages are public or private and whether the pages are static or dynamic (automatically generated by a script). Today's search engines index only public static pages and public dynamic pages, whose parameters are known or not required, thus leaving undiscovered a large amount of potential indexable information.

The total amount of indexed material is only a small fraction of the entire available Web data. As mentioned in [Table 1], private pages are not easily trackable and indexable, as they require login credentials [10]. Discovering the appropriate parameters (where they are required to correctly gather the Web pages) is a crucial task, as the missing or misleading of only one of the expected parameters can

2000 *Mathematics Subject Classification.* 68U15, 68U16.
Key words and phrases. Data Mining, Information Extraction, Invisible Web.

Page availability	**Page producer**		
	Static pages	Dynamic pages	
		Params known	Params unknown
Private	*Requires login*		
Public	Indexable by search engines		Requires domain-specific data

TABLE 1. Indexable Web - a small fraction of the entire Web contents

cause undesirable behaviour of the script which materialize the Web page, making impossible its correct collecting by the Web agent.

In this paper, we discuss all the steps of the roadmap to the exploitation of the Invisible Web material. We begin by describing various approaches to the discovering of the information hidden behind search forms and present our technique together with the motivation of applying it. The third section investigates the structure of the collected information and propose a solution to extract valuable information from the Web pages. The fourth section examines the extracted material and describes how to create new knowledge based on the hidden Web data and its practical usage. We conclude by presenting a number of issues we found during our experiments and propose alternative methods to be explored in future work.

2. Discovering the hidden material

In the previous works [1, 3], we have investigated two approaches for discovering and providing the appropriate parameters to be filled in a Web form: the first is a semi-automatic tool for specifying input parameters for the URLs of the dynamic pages that need to be downloaded and indexed, which relies on the information from local databases to instantiate the parameters and produce the pages [3]. A second approach consists in applying program analysis techniques on the source code of the scripts which generate Web pages in order to derive dependencies between Web page's input parameters and columns from the data repositories. After extracting these dependencies, an automated tool could simply collect all the possible values for each input parameter - as a finite set of values - and materialize all the possible Web pages obtained by instantiating the parameters with those values [1].

These two approaches illustrate a participatory vision, in which the tool responsible with the gathering of the Web material has access to local databases and the source of the scripts in order to retrieve information from them. In contrast, the black box model considers that the Web agent which collects the Web pages does not have the credentials to access local databases or the source of the scripts. Theese black-box Hidden Web crawlers apply form analysis tools, discover

common filling patterns or make use of heuristics to collect pages hidden behind search forms [5, 7, 10].

In this article, we consider the case of both private and public dynamically generated Web pages which can be gathered by the use of background knowledge. Our model consists in extracting information from an Invisible Web source, then using that information for instantiating parameters to a second source. Both sources of information are not indexable (i.e. invisible) by classical Web agents.

In our model, the first Web source contains identification data on Romanian business entities (such as fiscal ID, name of the entity, location, status), whereas the latter Web source contains financial data (financial statements on the last financial years). The sources are invisibles to the search engines, as the accession of the first one requires login credentials (we use a limited guest account which displays the minimum required information to be used in gathering content from the second Web source), while the accession to the latter Web source expects the input of the financial ID of each entity. The purpose of performing these steps is building new knowledge based on the data extracted from the Invisible Web sources.
Here are the basic steps our tool performs:

1. Collect and extract information from the first source
   1.1. Authenticate to the Web server using background knowlegde (login credentials)
   1.2. Extract the first page (comprising the total number of results)
   1.3. Navigate through the results of the query
   1.4. Extract information from the Web pages collected during the previous step (1.3).
2. Apply extracted information to the instantiation of the second source parameters, then gather data.
3. Create new knowledge from both invisible Web sources.

In the discovery process, the access to the pages is crucial, thus we use a semi-automatic approach, which consists in: a manual visit of *the login page* (for extracting login credentials), of *the query page* (for configuring the discovery tool) and of *the first result page*, which contains the number of results and links to the subsequent results pages, followed by the launch of the tool. The manual configuration of the tool is preferable to any automatic approach, as the information extracted from the first (or *base*) source shall be used in gathering Web material from the second (or *target*) source.

A fully automatic composition of input parameters could imply errors in downloading information from the first Web source, then missing downloadable pages in the case of the *target* Invisible Web site. The *error propagation* comes out as we apply information captured from the base Web source, then build the required

list of input parameters for the downloadable scripts, and finally gather the pages from the target Web source.

3. Extracting Information from the Invisible Web

Once a large amount of information is collected in a local data store, automated tools index it and republish it with the purpose of easily find that information as response to user queries through Web search forms. Instead of only index it (as in [1, 3]), we intend to extract data from both sources and create new knowledge that would enhance the value of the Invisible Web.

The aim of information extraction is to find relevant text in a document, that is a text segment and its associated attributes [6] or to find relationships between two distinct items of text [2]. As suggested above, this comes in contrast with the aim of information retrieval, which deals with the issue of finding relevant documents in a collection [4]. While multiple difficulties arise when extracting text from unstructured text, Web data has the advantage of comprising HTML tags which can be treated as text separators or can provide us with additional information on the data (for instance, tags like *<B>*, *<H1>* .. *<H6>* usually contain data as article titles, section of articles).

Invisible Web pages have an additional advantage of being automatically generated by a Web script, with useful material from columns of databases. We have manually investigated the material extracted from the two Invisible Web sources mentioned in the previous section and classified two different situations, in which the source renders *a single row* of the database and *multiple rows of the database*.

In the former case, the Web page is structured as follows:
$< TR >$
$< TD >$ Description of first column $< /TD >$
$< TD >$ Content of first column $< /TD >$
$< /TR >$
...
$< TR >$
$< TD >$ Description of n-th column $< /TD >$
$< TD >$ Content of n-th column $< /TD >$
$< /TR >$
whereas, in the latter case,
$< TR >$ *[Header row with description of columns]*
$< TD >$ Description of first column $< /TD >$
...
$< TD >$ Description of m-th column $< /TD >$
$< /TR >$
$< TR >$ *[Content of the first row]*
$< TD >$ Content of first column $< /TD >$
$< TD >$ Content of 2nd column $< /TD >$

...
$< TD >$ Content of m-th column $< /TD >$
$< /TR >$
...
$< TR >$ *[Content of the n-th (last) row]*
$< TD >$ Content of first column $< /TD >$
...
$< TD >$ Content of m-th column $< /TD >$
$< /TR >$

The discovery that Web pages from the same site share the same structure conducted us to applying pattern matching for extracting useful data from the documents. The patterns are manually constructed and make intense use of $< TD >$ and $< /TD >$ tags to delimit two columns of the table, or to delimit the description of one column from its contents.

4. Creating new Knowledge

Building new knowledge is the subsequent step after extracting useful material from the Invisible Web pages. We have investigated the Web sources from which to collect the pages and figured out that interesting information could be generated after inspecting related data.

We consider two Web pages P_A and P_B that contain financial information on companies C_A and C_B. We say that P_A is related to P_B if C_A *competes* with C_B, that is C_A and C_B have the same activity code (described in [8]). The *competition* is either *local* (when the two companies also share the same county of residence) or *national* (when the two companies do not share the county of residence).

Our goal in creating new information is to build the list of the first competitors (either local or national) which share the same activity code. The list also varies on a criteria like the total number of company's employees or the turnover on a specified year. Creating such synthetical information is similar to the classical Strengths, Weaknesses, Opportunities and Threatenings analysis [9]. This type of analysis can be useful for competitors in discovering the tough and weak points of the companies in the same activity domain; it can also be useful for clients of those companies in analysing their position on the local or national market, and it also provide the entrepreneurs with interesting investing opportunities (like finding sectors with weak competition, or counties with available work force). We present an example of such an analysis that is automatically generated after extracting the useful information from the available Invisible Web pages.

In [Table 2] we outline the results of quering our tool with the keywords *7221*, *employees*, *Cluj*, which returns the first 10 entities from the Cluj county whose activity code is 7221 (*The editing of software programs*). We obtain the list of the entities in descending order by the number of employees and render it considering

Entity	*Em*	Aim	Ac	Ct	Dat	Ca	Ve	Pb	Sal	Rpr	Pca
Intellisync	1.00	6			3	5	5		1	9	
Nivis	0.70	3	3	4	1	2	2	2	2		
Transart	0.65		4	5		6	6	8	3		
EBS	0.60	5	9		4	4	4		4		
ISDC	0.45	9	5	8	8	8	8	6	5		
Alfa Global	0.40		6	6	5	7	7	4	6		
Montran	0.38					1	1		7		
Recognos	0.33					9	9		8		
Arobs	0.31								9		
Ro planet	0.29										
Fortech	0.22		7	7				3			
Nethrom	0.15										
Api	0.09	7			7						
Vectorsoft	0.08										
Transylvan	0.08										
Q soft	0.08										
BNW	0.08										
Depart	0.08										
I I Studio	0.07										
Arxia	0.07										

TABLE 2. Information obtained from the Invisible Web sources

the relative number of employees (the number of employees of the current entity divided by the number of employees of the top entity).

We also render the positions of each entity by considering ten distinct criteria: *AIM* (Tangible assets), *AC* (Intangible assets), *CT* (Total capitals), *DAT* (Total debits), *CA* (Turnover), *VE* (Total income), *PB* (Gross profit), *SAL* (Employees), *RPR* (GrossProfit/TotalCapitals) and *PCA* (GrossProfit/Turnover).

This second classification orders the top nine entities on each one of the mentioned criteria. For instance, the *SAL* column points out all the nine positions of the top, whereas the other columns do not necessarily display all positions. This happens because entity's strength in a category does not guarantee a good position in any other category, with the exception of the *Turnover* and *Total income* columns.

In the illustrated example, the *Turnover* and *Total income* columns generate the same order for the listed entities. The result is expectable, as the *total income* includes the *turnover*. Furthermore, companies in various sectors do not output financial or extraordinary income, making the two cited columns publish almost the same order on companies.

5. Issues

In this section, we discuss some of the difficulties we found during our experiments and propose solutions to be explored in future work.

Information freshness: We conducted our experiment on the companies' financial data found at the end of 2004, which were published on the Internet in the late 2005. There is almost half-year delay between the availability of the information at the companies and the publishing of that information on the Internet and almost one year delay between the end of the financial exercice. Even so, the results provide the user with interesting knowledge, such as *the top companies on a certain activity domain*, *the strengths and weaknesses analysis* [9] *of the top companies*. A solution to the freshness of data would be the implementation of a participatory system in which companies upload their financial results as soon as they release them. A success of such an architecture would require a very large number of collaborating participants (almost all of the active companies) to provide us with useful material.

Extracting data on entities: We have explained in the *Information Extraction* section our approach to obtain information on entities from both the base and target sources of Web material. This approach considers that the structure of a hidden Web page is persistent over different invokes with input parameters. From this point of view, we can easily apply regular expressions in order to obtain the needed information. The problem appears when pages change their structure, making impossible the extraction of the data by use of the initial regular expressions. Introducing named-entity recognition techniques would imply a correct extraction of the places where companies reside (easily found in dictionaries), leaving uncertain the possibility to extract the name of the companies (as there is a vast variety for those names).

Extending the model to a larger scale: The primal goal of our work was to build a semi-automated tool for extracting content from the Invisible Web, considering the value of the information hidden behind search HTML forms and the possibilities to enhance it. We have investigated several Invisible Web sites and built a model formed of two Web data sources. One of the directions for future work would be to extend our model to perform the extraction from a more complex Web structure, such as a group of tens of Invisible Web sites. To extract valuable information from them, these sites have to be related, that is some part of the data gathered from one site must be used in another site (for instance, the Unique Fiscal ID of one company or the Private Numeric ID of one person). A hypothetical extention of our architecture would be a third site having the list of employees for every active company, then another site publishing personal information on people (like complete address, phone numbers). This type of extention raises privacy concerns, but it also poses other questions like how to obtain access to such sources of information. The fact that we can easily find Invisible Web sites

which contain data on companies does not guarantee the success in finding hidden Web information on persons, nor the existance of the sources in the near future. We believe that information on persons should remain private, thus limiting the possibilities of extending our model. However, we intend to extend the depicted tool to periodically collect and extract information from the sources, by applying the same techniques for the gathering, the extraction of valuable information and the creation of new knowledge as the information on both invisible Web sources changes.

6. Conclusions

We have described a model of gathering, extracting and enhancing the information from the Web pages whose content is kept in large databases and that are automatically generated as response to user queries. The results highlight the value of the information hidden behind search forms and how new information can be generated by applying pattern matching techniques on already existing Web material.

The techniques we have experimented are easily applicable to other Invisible Web sources. We are investigating the possibility to extend our model to include several related Web sources whose content is not trackable by traditional Web agents. We are also examining various data mining techiques for discovering valuable patterns or correlations on the gathered material. This would significantly improve the value of the Invisible Web, contributing to the creation of (what we call) an *Enhanced Invisible Web*.

References

[1] G. Attardi, A. Esuli, L. Hancu, M. Simi, 2004, *Participatory Search*, Proceedings of the IADIS International Conference WWW/Internet 2004, Madrid, Spain.
[2] S. Chakrabarti, *Mining the Web, Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Publishers, 2003, pp 290-295.
[3] L. Hancu, *Discovering Hidden Web Content*, Babes-Bolyai University, Graduation Paper, 2002.
[4] D. Hand, H. Manilla, P. Smyth, *Principles of Data Mining*, MIT Press, 2003, pp. 456-470.
[5] K.I. Lin, H. Chen, 2002, *Automatic Information Discovery From The Invisible Web*, International Conference on Information Technology: Coding and Computing, Nevada, USA.
[6] Manu Konchady, *Text Mining Application Programming*, Charles River Media, 2006, pp. 155-182.
[7] S. Raghavan, H. Garcia-Molina, 2001, *Crawling the Hidden Web*, Proceedings of the 27th Conference on Very Large Databases, Rome, Italy.
[8] Romanian Registry of Commerce, Nomenclator CAEN, *http : //recom.onrc.ro/obco.htm*
[9] Wikipedia, *SWOT analysis*, *http : //en.wikipedia.org/wiki/SWOT_analysis*.
[10] Ricardo Baeza-Yates, Carlos Castillo, 2005, *Crawling the Infinite Web*, Journal of Web Engineering.

[1] SoftProEuro Cluj-Napoca