A New Solution for Data Extraction: GENE/LONE Method.

Benlahmar El Habib,† and Doukkali Sdigui Aziz ††, El ouerkhaoui Asmaa†††

University of Mohamed V souissi - Higher national school of data processing and analysis of the systems, Rabat Morocco

Summary
Although the Internet presents a wealth of information, it is not totally usable. In fact, given that the Net is large in size and not enough structured, localizing the relevant information is time consuming.
Therefore, extracting data from the Internet, especially in public sources and making it available for final users, programs or applications is very much motivated, this paper is a contribution to resolve and improve the extraction systems of web data.
Key words:
Data retrieval, XML, Gene/clone.

Introduction

In this paper, we propose a new approach based on the analysis of the structure of results pages. We reduce human intervention in the wrapper generation process to a simple cut/paste operation.
The idea consists of extracting the smallest structure that generates the global structure of the zone which contains the pertinent information.
The smallest structure will be structured as the document's gene. Then, we use this gene to reproduce a structure and to superimpose it to the results documents in order to extract relevant data. In other words, from a gene we clone a structure that uses an XSLT filter so as to get pertinent data. Thus, the approach we adopt here is called "extraction by the cloning method"
The gene structure is represented as a set of relevant data characteristics, especially its textual and structural context. This is done for the purpose of extracting the textual parts of all the occurrences that have the same structural and textual context as that of the "gene" structure. The idea is to XMLisate (i.e. to convert an HTML format into an XML one) results pages so as to give the document a standard format [1] [2]. This, actually, makes the document legible, comprehensible and useful. The user gives as an input a set of example instances to extract. The context of each example value (called context_value) and the context of each instance (called context_value) are sought in a page issued from the source, and from the context_value and context_instance, the relevant nodes will be detected to get to the relevant information in a search session.
Therefore, instead of seeking relevant information, we now look for relevant nodes, namely those which include the desired information

Manuscript received June 5, 2006.
Manuscript revised June 25, 2006.

2 work context

In the year of 2000 our researcher team in the ENSIAS started with extraction data. First of all, there was the method of FDA (Final and determinist automat) which uses definite robots [1], [2]. This method has been approved but shows a lot of weakness, the most current is the fact of the page's modification taking from web sites that affect our application in the way that it fell. In the year of 2002, we proposed the GENE/CLONE method [6], [7], [8], which work out the weakness of the ancient method, but it also bring some self weaknesses, however it brings better results that the FDA methods
The most popular of the developed application that we have done is the BERG project:
In the Berg Project we extract data from on-line telephonic directory Meta motors.
Actually, there is today two laboratories versions for WBerg : 1.0 and 2.0 [1], [2], and two versions declined on a moving support: 1.0 and 2.0 that is presented in this paper. [3].
Berg 1.0: This version of Berg uses definite robots to display structures. For each target research we use a specific code.
Berg 2.0: This version use XSLT filter to obtain relevant information instead of using definite robots.
Wberg: After the Web versions of Berg, we proposed a moving version Berg2.0, it is also based on XML technology. In fact, we use this technology to standardize the document format.

3 State of art

3.1 Wrappers based on labeled pages

Wrappers based on labeled pages are wrappers whose input is the pages where the needed information is identified by the user. Two of these wrappers are WIEN and STALKER [11]; [12]; [13],

3.2 Final extraction patterns starting with the analysis of documents

The appearance of semi-structured Web pages gives regularity to these pages. The objective is to exploit this regularity by analyzing the structure of the document. This technique is generally called extraction of patterns starting with the analysis of documents. Many approaches have been proposed using this technique, two of which are the IEPad [9] system and ROADRUNNER [10] system. This technique is based on the idea of built patterns by exploiting the regularity of the structured web pages. And starting with these patterns we retrieve data

3.3 Relation extraction

This method relies on a duality between the patterns and relations explained below. According to a set of relations we find an occurrence in the document used in order to build patterns. The correspondence between the relation and patterns resides in the fact of generating a new tuple that allows us to extract patterns from a text[3]; [4]; [5].

4. Approach

4.1 Presentation

The first step in the GENE/CLONE approach is to fill a form in order to obtain information from the chosen source (generally a web site) the form allows the construction of a request which gives a response. The received information is in an HTML format. These pages are not well structured that's why we proceed to an XMLisation of these pages (transform pages from an HTML format to XML). The third step is the most important, in this step we take an instance from the chosen source in order to get its XPATH, this will allows us to know where the relevant information are, Our purpose is to get the sub-structure that contains the relevant information. This structure will be called "target structure". And to get the smallest sub-structure which gives the whole relevant structure, this sub-structure will be called "generic structure". As from this, we can have for every required information an XPath, therefore instead of looking for an information we look for a node where is the relevant information.

This is a sort of cloning structure technique. To localize the generic structure, we need to know the relevant information. This is, actually, why we have decided to use examples. Thus, the users make use of some examples to express their need in terms of the pertinent information that is to be extracted. Each example is called instance.

However, a relevant node can have one or more occurrences that do not contain relevant information. Therefore, having an idea about the node does not solve the problem. For this purpose, we have to know the node's parents, or its entire paths.

Instance values do not always constitute the entire textual content; they can present part of the textual content. For example, the phone number in an on-line telephone directory can be displayed after a "phone" string, or the price in on-line sale sites can show up after the "price" string and before "EUR" or "Dollars" strings. Hence, we note the prefix t_i the textual sub-string that comes after t_i and the t_i suffix the sub-string that comes before t_i.

4.2 Generation of the textual context

We define the suffix (idem. prefix) of a ti value as a sub-string that comes after (idem. before) ti. However, this definition is insufficient to generate suffixes and prefixes. For example: let's consider the string: "Tel.: 037998899 fax.: 037558898". The suffix of the fax number is: "Tel: 03755889 Fax.: and the prefix of the phone number is: "fax. : 037558895". We note that the suffix and the prefix contain instance values. We can not extract information since these values do not figure in the suffix and prefix in the other instance. For this reason, we define the valid prefix and suffix.

Compared to existing methods, this approach has several advantages. It needs no labeled pages, and it is adaptive. Indeed, the rebuilding of the extraction rules following a change of the source's format include no labeling and the data extracted before the format change can be used to build a new extraction rule.

Let's consider the previous example: "Tel.: 037558899 Fax.:037558898", the initial prefix of the "037558899" value is "fax.: 037558898". Since it contains the "037558898" value, and the prefix will be the suffix of the "037558898" value, thus, the sub chain "fax.:".

The initial suffix of the "037558898" value is "tel.:037558899 fax.:" which contains the "037558898" value. Therefore, the suffix will be replaced by the prefix of this value which is the sub chain "fax.:".

Often, the presence of suffixes and prefixes is used to give the user the semantic meaning of the data appearing in the page. For example, if the user finds the "tel.:" chain before its number, he concludes that it is a phone number. The suffix and prefix are obligatory, especially if a lot of information that has the same format (phone, fax, mobile numbers) is proposed. Thus, the prefix and suffix are used as a label. Henceforth, we can use them to increase the precision rate of our data extraction approach.

However, a ti value of a t instance can appear several times in documents. These values are called occurrences.
A ti value of t instance has a context; it is, for instance, the prefix and the suffix of the value and the node which contains the t_i value in the textual content.
Therefore, we can divide the context of the value into two contexts: the textual context by the couple (suffix, prefix) and the structural context defined by the path which is the XPath of the relevant node.

However, a value can have several occurrences and each occurrence has its context. We defined the context of a t_i value as the set of contexts of different occurrences of t_i.
Formally, a context_value can be considered as a C function, which associates a ti value to a set of triples made of (αpathij,suffixij,prefixij) where I presents the index of the I value and j presents the index of the O_j occurrence.
An instance has a context which is in a larger context. It is the global context where all the relevant information is. To determine this context we use a set of example instances (two or more).
Finally, we defined the generic contexts as the union of the values context as well as the instance and global contexts.

4.3 The approach

Let's consider a D document from an S source search, and t=(t1 , Λ , tn) and t=(t'1 , Λ , t'n) two example instances.
To build extraction rules for relevant information on S, our approach goes through the following phases:
1- pre-treatment of the document.
2- Seeking the occurrence of instance values.
3- Extraction of value contexts.
4- Extraction of instance contexts.
5- Construction of extraction rules.
The extraction rule construction process is defined by the following functions:
- **XMLisation**: From the raw string of a document body, we build an XML document or simply an XML tree.
- **Seek**: Seeking the set of occurrences of instance values in the document.
- **V_context**: Extracting the values contexts of example instances.
- **I_context**: Determining the contexts of example instances.
- **R_construction**: Constructing extraction rules.
- **Extract**: Applying extraction rules to the document or to the XML structure.

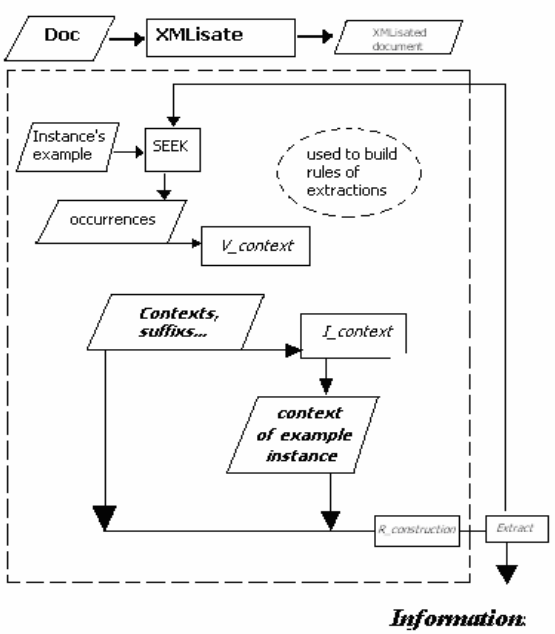

Fig. 1 Schematic of extraction.

Pre-treatement of the document (XMLisation): In this phase, we transform the structure of the HTML document into an XML structure or XHTML. We do not consider the entire document but only its body since the useful information is there. But before making this transformation, we delete some useless information that is often low level markups. HTML markups are divided into two classes: High level markups and low level ones.
Low level markups allow us to modify the appearance of a text sub-string but they do not contain structural information such as (,,<b>) and script markups. High level markups such as (<p>,<td>,<li>) give information about the structure of the data representation. It is these markups that we keep.
In a search session within a search source, we only consider the part of the document that is determined by the global context.
Seeking occurrences: Let's consider two examples of instances (or more) t=(t1 , Λ ,tn) and t=(t'1 , Λ , t'n), $\forall i \in [1,n]$ Once we have found ti occurrences, we represent them by triplets in the following formats (Xpath, Suffix, Prefix), where Xpath is the Xpath of the i occurrence, and suffix and prefix are respectively the textual prefix and suffix of the i occurrence.
The search of all the occurrences of a tuple that have a t value (t=(t1 , Λ , tn)) consists of seeking all the occurrences of each ti.
Let's consider occ(t_i) the set of ti occurrences. The occurrences of the t tuple correspond to the Cartesian

product occ(t1)xΛocc(tn). The number of these occurrences can be large if one of these values has many occurrences in the page.

To reduce the number of generated occurrences for the tuple of a value, we can define some heuristics. For example, we consider that the more values belong to the smallest sub-tree, the more they can belong to the same tuple.

The search phase of occurrences consists of seeking the set of occurrences of each instance in the set of the source documents.

From a tuple in the document, the occurrences of t1 are sought. The set of occurrences of the sub-tuple (t1,…,tn) are looked for in a recursive way. In this case, any classical algorithm for the search of strings either in a document or in a tree can be used.

Extraction of context values: This function is used to generate the context of the relevant information to be extracted. For this purpose, we use two instance examples (or more), and we determine the suffixes and prefixes of each occurrence of ti and t'i values. Then we favor the values that have the same structural context and the same textual context.Let's consider two instance examples. For each i value, the t'i and t'i values have the same structural and textual contexts. This proposal entails that the contents of the pages issued from a web data source have been taken from structured data belonging to a database and formatted so that the same information of the same type can be presented similarly.For example, in the on-line telephone directory, the addresses' context is identical in all instances. This actually holds true for on-line selling websites. The prices of products are displayed in the same way in the document. Thus, they have the same structural context for all relevant information instances. We note occ(ti) the set of the ti values occurrences. The generation of the context of values brings about a lot of problems, especially in cases in which the ti and t'i values have one or more occurrences that have the same textual and structural contexts.

Let's consider, for example, the case of on-line telephone directories, if two addr and addr' addresses have two occurrences in such a way that addr1 (addr2, respectively) has the same context as the addr'1 occurrences (addr'2, respectively). Once we compare the context of both addresses, we encounter the following problem: "Which occurrence is to be kept?" Following the definition of the value context, two values of two instances of the same position have the same context. In this case, however, addr2 and addr'2 have the same context and so do addr1 and addr'1. In such as a situation, we consider both occurrences.

Let's consider two values t_i and t'_i of the same position of two instances t and t'. Every occurrence O_i of ti has to be compared with each O_k occurrence of t'_i. They have the same textual and structural context. Therefore, we add the context of the occurrence in the set of values context and we delete the O'_k occurrence from the set of t'_i occurrences. The same procedure is repeated until the set of t'i occurrences is empty.

Extraction of the instance context: So as to generate the smallest sub-structure that generates the generic structure (i.e. which generates the sub-structure that includes all the ti values), we determine in this phase the structural context of all the relevant nodes.

The definition of the instance context states that it is the smallest structure which contains all the t instance values. However, the value_context of a ti value is not unique. In fact, a value can have several occurrences and a set of value_contexts.

Let's consider Pi the set of structural contexts of the ti value of a t instance. If card(Pi)=k, the ti value will appear k times in the document and the t'i value of the second example instance will also appear k times in such a way that the textual and structural contexts of ti and t'i occurrences will be equal two to two.

In this case, we have to choose the occurrence that represents the relevant information. To solve this problem, we opt for the closest occurrences. Therefore, we define the distance between two nodes (two structures, respectively).

Construction of extraction rules: From the global context we can localize the relevant structure, which is the sub structure which contains all the pertinent information. Then, from the context's instance we can gain access to the sub-structure which includes the entire relevant information block, and for each block we access, via value contexts, to the relevant information.

The instruction rules can be seen as an XSLT file. This means that from the structural information that we have generated, we can easily build an XSLT file and apply it to the source documents after pre-treatment.

5. Experiment and results

To evaluate the performance of our system, we used more than 500 downloaded pages from 10 sites. One of the first fields on which we tested our approach is on-line telephone directories. We made use of two largely widespread parameters in the extraction: recall and precision.

Recall (R) is the percentage of data to be correctly extracted from the corpus. This highlights the proportion of correct extractions. Precision (P) is the percentage of data correctly extracted by the wrapper, and this is what evaluates the quality of the extractions.

Formally, considering TP the number of correct extractions, FP the number of incorrect extractions, and FN the number of data to be extracted but actually it is not. We can define Recall and Precision as follows:

R =TP/TP + FN and P =TP/TP + FP

The results obtained are satisfactory. In all the experiments on the directories, we used two authority examples for each

directory. The table below shows the results obtained from some web sites:

Table 1: the recall and precision results

	TP	FP	FN	R	P
www.google.com	10	1	1	90%	90%
www.kelkoo.com	20	0	0	100%	100%
www.pagesjaunes.ch	10	4	4	74%	74%
www.froogle.com	1	0	9	10%	10%
www.altavista.com	10	0	0	100%	100%
www.yellow.ca	19	0	0	100%	100%
www.bizrate.com	20	0	20	100%	100%

6. Application

6.1 WBerg application

We start apply the GENE/CLONE method in web application, the first one namely BERG deals with on-line directory. We entend from this apllication to extract some information from entreprise Meta motor search engine such as (phone, name of entreprise, fax number...). This application gives satisfied results for a lot of countries.

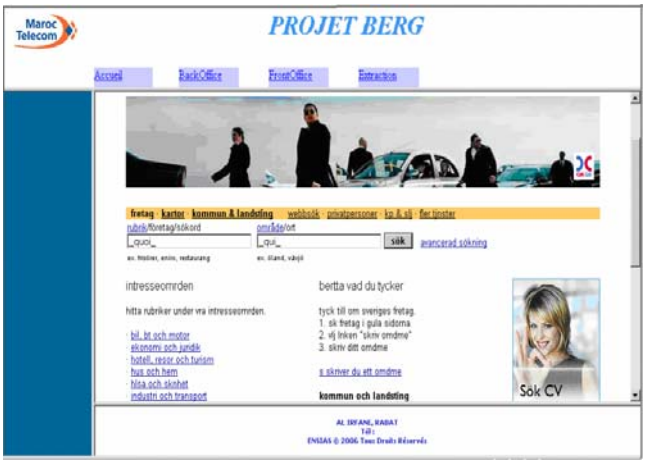

Fig2. Configuration page for request module.

The first thing to do to configure the BERG application is to enter the three required parameters: '_qui_', '_ou_',and '_quoi_'.
'_Qui_': refers to the seeking enterprise name.
'_Ou_': refers to the seeking enterprise location.
'_Quoi_': refers to the seeking enterprise category.
Once entered, we launch the research, in the back office; our application use on-line directories to find the seeking enterprise.
The first thing to do is to configure on-line directories for every country we want to add in our BERG application. In back office we fill the form proposed by this on-line directory in order to build an XSLT filter so that we can use it after to get similar information as from our own form.

Fig3. Configuration page for data extraction module

The figure 3 shows our proposed form, where we enter the WHO (_qui_), WHERE (_ou_) and WHAT (_quoi_) parameters to get the target enterprise.

Fig4. Result page

The figure 4 shows the result page, this page contains specifically the entered parameters of the founded enterprise. .

6.2 WAP application

The Moroccan telephonic agency IAM is actually financing the BERG project. After using BERG on the web, the IAM agency was interesting about a mobile migration for using BERG on the WAP phone. The WAP application has given good results and the main results behind this are the use of XML documents that eased the use of extracted information.

In figure 5.1 we proposed the form used by every customer to enter his request, in figure 5.2 we show the response form that contains the founded enterprise.

Fig 5.1- Open wave Phone interface (filling form)

Fig 5.2- Open wave Phone interface (response form)

7. Conclusion

In this article, we proposed a new approach for data extraction from structured semi-documents.
The results we obtained have proven that our method allows, in the large majority of cases, to build a wrapper quickly using some authority examples.
Moreover, the format of the extracted instances is known beforehand because it is similar to that of the instance given. Thus, compared to the existing methods, our approach has several advantages. It does not require the labeling of a set of example pages, and it is adaptive.
In fact, the reconstruction of the extraction rules due to a format change of the source requires no labeling. Furthermore, the data extracted before the format change can be used to build a new extraction rule.
We are currently extending the shape of the gene so that it could also consider structural suffixes and prefixes. We believe that this gene extension will enable us to increase the performance of our Cloner system.

References

[1] El Habib BEN LAHMER, Abd Elaziz SDIGUI DOUKKALI, Mohammed OUMSIS. La Méta recherche générique: vers la génération des méta moteurs de recherche. CopStic'03 Rabat, Maroc.
[2] El Habib BEN LAHMER, Abd Elaziz SDIGUI DOUKKALI, Mohammed OUMSIS. Towards An Automatic Extraction Of Data from Half-Structured Documents. In ISCCSP2006.
[3] X. GAO et L. STERLING. Semi-Structured Data Extraction From Heterogeneous Sources. In Second International Workshop on Innovative Internet Information Systems (IIIS'99), Copenhagen, Denmark,1999.
[4] X. GAO et L. STERLING. AutoWrapper: automatic wrapper generation for multiple online services. In Proceedings of the Asia Pacic Web Conference, , Hong Kong, September 1999
[5] Heekyoung SEO, Jaeyoung YANG et Joongmin CHOI. Knowledge-based Wrapper Generation by Using XML. In IJCAI-2001 Workshop on Adaptive Text Extraction and Mining, Seattle,Washington, August 2001.
[6] BEN LAHMER El Habib, Abd Elaziz SDIGUI DOUKKALI, Mohammed OUMSIS, 2004, WBerg un méta annuaire WAP,in isivc'04 Brest France.
[7] El Habib BEN LAHMER, Abd Elaziz SDIGUI DOUKKALI, Mohammed OUMSIS. La Méta recherche générique: vers la génération des méta moteurs de recherche. CopStic'03 Rabat, Maroc.
[8] El Habib BEN LAHMER, Abd Elaziz SDIGUI DOUKKALI, Mohammed OUMSIS. Towards An Automatic Extraction Of Data from Half- Structured Documents. In ISCCSP2006.
[9] Chia-Hui CHANG et Shao-Chen LUI. IEPAD : Information Extraction based on Pattern Discovery. In Proceedings of the ACM WWW10 Conference. ACM Press, 2001.
[10] Valter CRESCENZI, Giansalvatore MECCA et Paolo MERIALDO. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In The VLDB Journal, pages 109–118, 2001
[11] Nickolas KUSHMERICK, Daniel S. WELD et Robert B. DOORENBOS.Wrapper Induction for Information Extraction. In Intl. Joint Conference on Artificial Intelligence (IJCAI), pages 729– 737, 1997
[12] Nicholas KUSHMERICK. Wrapper induction: Efficiency and expressiveness. Artificial Intelligence, 2000.
[13] Nicolas KUSHMERICK. Wrapper Induction for Information Extraction. Thèse de Doctorat, University of Washington, 1997.