PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

Unsupervised information extraction from Web sites using bioinformatic techniques

CARLOS ANDRADE INDO

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
JAIME NAVÓN C.

Santiago de Chile, January 2011

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

Unsupervised information extraction from Web sites using bioinformatic techniques

CARLOS ANDRADE INDO

Members of the Committee:
JAIME NAVÓN C.
ROSA ALARCÓN
SERGIO OCHOA
GONZALO PIZARRO

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, January 2011

Dedicado a mi familia.

ACKNOWLEDGEMENTS

This work would have been very difficult to complete without the support i received from my family and friends during my research. I particularly I would like to thanks to those who were of great importance to complete my work.

First I want to thank my parents for their unconditional support in the decisions i made in my life. My sister Verónica and his fiancé Marcos for their support and help.

I thank my professor Jaime Navón by their guide, help and motivation not only for the preparation of this work, but my entire college life.

My lifetime friends Leonardo Luarte for their advice, support and feedback during the entire process and Natalia Corominas for his support during the final stage.

Finally, I want to thank my Lab buddies: John Owen, Raúl Montes and Rodrigo Alvarez for they great help during the process while tolerating my daily-basis crazyness.

Contents

List of Figures

ABSTRACT

Integration has been a recurring theme in the context of Web sites and Web applications in these last years. Web sites are seen as important sources of information and they are expected to provide public access through web services, so many new sites and applications provide APIs to fulfill these expectations. Nevertheless, the vast majority of existing web sites do not provide a standardized interface to access the information, turning the integration of these sites into a daunting task.

Important efforts have been made in information extraction oriented towards facilitating the process of retrieving relevant information from web sites in an automated manner. The focus, however, has been on the information itself, leaving the meaning of the information gathered up to the user. This is clearly not sufficient for the automatic synthesis of web services or APIs, and a contextual model for the collected information must be added. We propose BIOIE, an unsupervised system to extract an information model from a web site that could be used for the automatic generation of associated web services. To build this model, we used properties available within the sites and an approach inspired in bioinformatics to recognize the information patterns.

We implemented and tested the system with a variety of web sites to prove technical feasibility. In all cases, it was possible to extract a rich, useful model, which was completed in a short time period without supervision. These results are very promising since from here to actually generating the web services, we do not anticipate important difficulties.

Keywords: information extraction, Web Services, sequence alignment, software integration

RESUMEN

La integración de la web ha sido un tema recurrente en estos últimos años que toma cada vez más fuerza en este mundo de grandes fuentes de información. Muchas sitos de la web han comenzado masivamente a incluir estos conceptos dentro de sus productos incluyendo acceso público a servicios web o Apis. Sin embargo la gran mayoría de los sitios no proveen de interfaces estandarizadas para acceder a su información, haciendo la integración con estos sitios una tarea muy compleja.

Grandes esfuerzos en information extraction se han producido para lograr obtener información relevante de distintos sitios en la web de manera automatizada, usualmente enfocándose en la información misma que se quiere extraer, dejando el propósito de esta información recolectada al usuario. En el caso de buscar generar servicios o Apis automatizado esto no es suficiente, ya que se vuelve necesario también un modelo contextual de la información obtenida para dar un paso más allá.

Proponemos un sistema no supervisado que extrae un modelo de la información de un sitio en miras a construir a futuro un sistema para construcción de servicios web automática. El software utiliza para la construcción del modelo propiedades presentes en los sitios web junto con conceptos traídos de bioinformática para obtener patrones de información. Este trabajo describe la perspectiva global de nuestro objetivo, retos enfrentados, una implementación prototipo que muestra la factibilidad técnica y objetivos para el futuro.

El software fue probado utilizando información no etiquetada de distintos sitios de nuestro interés, obteniendo buenos resultados en términos de calidad y tiempo de ejecución, logrando además de extraer la información obtener un modelo rico para los pasos siguientes de nuestro gran objetivo.

Palabras Claves: extracción de información, Servicios Web, alineación de secuencias, integración de software

Chapter 1. INTRODUCTION

The web has become not only a place to share documents and images but also a platform to expose, communicate, and integrate applications and information. Due to the large amount of information available on the Web there has been a great emphasis on applications and services that can integrate and coexist with these massive data sources, and standard tools that provide easily integration and exposition of functionality and resources easily are a necessity.

Many of the most modern web sites provide web interfaces to expose their capabilities to other third party applications and most new software projects use the web protocols and standards leaving behind other integration tools and standards such as CORBA (Group, 1999), COM [1] among others.

The rise of Web Services formalizes the idea of using Web interfaces for information exposure. They are usually separated in 2 groups: Big Web Services and REST Web Services(Pautasso, Zimmermann, & Leymann, 2008). Although with different approaches and methodologies, they both share the goal of using web standards for application communication. Great progress of standardization and ease of use has occurred in the last years, which has resulted in mass adoption of this technology to become a de facto standard for communication applications.

However, while the Web is evolving and companies and institutions are updating their sites according to current technological trends, many other sites are not updated, leaving a large amount of useful available information outside of this new world of easy integration. This forces the users who need to connect to these sources to build complex hard-to-maintain extractors that capture information on non-standard ways generating additional barriers to the difficult task of making a more intelligent and integrated Web.

[1] See http://www.microsoft.com/com/

Before getting into the integration problem itself and particularly into a proposal for a solution to the problem we present an overview of web services, some information extraction tendencies, a little introduction to sequence alignment and bioinformatics, and other topics that relevant to better understand this work.

1.1. Introduction to Web Services

1.1.1. Web Service Definition

There are several possible definitions for such a broad concept as Web Service. According to W3C, "A Web Service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols." (Schlimmer, 2002).This definition includes some key elements such as the machine-machine nature and the use of internet protocols but relates more to the "traditional" web services, usually built using SOAP (Gudgin et al., 2002) giving no much room to incorporate REST Web Services (Fielding, 2000).

Stencil Group (Sleeper & Robins, 2001) gives broader definition: "Web Services are loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols". This brief definition considers all the key elements of a web service from this research perspective:

Reusable software components: We want our service to be reusable. In fact, this is one of the service's main characteristics considering that our objective is to generate interfaces that exhibit some functionality to other services for which it could be useful.

Programmatically accessible: A web service must be oriented to be consumed not by humans but rather by other applications. This is the main reason why we will not consider a web site as a web service.

Distributed over standard Internet protocols: A web service must exhibit its functionalities using internet standards and not proprietary technologies. This way we can assure an easy integration using mass accepted and neutral technologies and, by not specifying what standards are, leave the possibility to build services under SOAP, REST or any other approach that could present in the future.

Encapsulate discrete, loosely coupled functionality: A web service provides specific functionality in the same way a web site offers delimited functionalities and information to a particular topic. This functionality should be used as a loosely coupled component, without the need of external needed dependencies to use the exhibited features.

So in the rest of the document we use the Stencil Group definition of a web service and the points presented before as its needed characteristics.

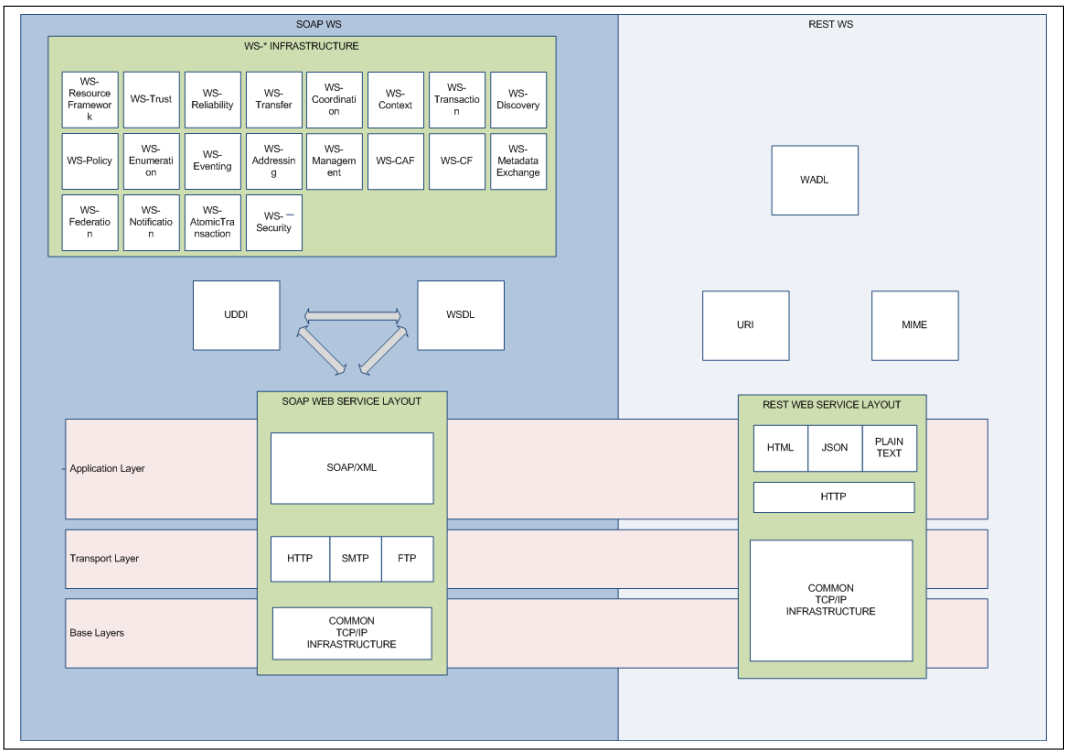

FIGURE 1.1. Diagram showing the different technologies and standards in each of the Web Service approaches

1.1.2. Web Services Aproaches: SOAP vs REST

There are two main approaches for the implementation and exposure of Web services: "Big" Web Services,which use SOAP and all its related technologies(Gudgin et al., 2002) and REST Web Services, based on the REST architectural style presented by Fielding (Fielding, 2000). We present next a very brief introduction of each of them.

1.1.2.1. SOAP Web Services

According to W3C "SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics."(Gudgin et al., 2002)

In this way, SOAP is concerned about how messages are structured and provides the facilities to build the semantics of information, delegating other tasks such as transmission or format issues to other web standards like XML, HTTP, XML-SCHEMA, etc. SOAP alone however, is not enough to build services because a way to describe the service and the exhibited functionalities is still required. The Web Service Description Language (WSDL) covers exactly this need.

According to W3C "WSDL addresses this need by defining an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication" (Chinnici, Moreau, Ryman, & Weerawarana, 2007). In a WSDL document, a service is considered as just a set of endpoints or ports, separating thus the specific implementation details from the service definition. This definition specifies and describes each service operation, leaving to SOAP the communication and the rest of the Web standards.

Thus WSDL with SOAP, XML and HTTP build a highly specific and neutral platform for building Web services using the Remote Procedure Call and Message Integration architectural styles. Due to the complexity of SOAP, big companies like Microsoft, IBM or SUN (now Oracle) have built powerful tools to simplify the process of building a Web Service and and encourage the use of this technology for business.

1.1.2.2. REST Web Services

REST web services approach emerged only in the last few years. Unlike traditional Web services that use RPC as the main architectural style, REST services use the REST architectural style introduced by Fielding (Fielding, 2000). REST is a hybrid style derived from several network-based architectural styles combined with additional conditions that defines a uniform connector interface. Particularly, REST is a style derived from:

Null Style: The starting point Style

Client-Server Style: REST includes Client Server constraint

Stateless Syle: We add the Stateless Connection constraint to the Client Server Interaction

Cache: We add Cache feature to our Stateless Client-Server Interaction for network efficiency and scalability purposes.

Uniform Interface: REST add the constraint of a Uniform Interface for components interaction. This improves the visibility of the system and at the same time simplifies the process. As a tradeoff, the Uniform Interface degrades efficiency by forcing standard interfaces instead of custom ad-hoc interfaces specially designed for accomplish the goals of the system.

Layered System: To the actual constraints,we add a hierarchical layered system constraints in order to achieve internet scalability needs.

Code on Demand: Finally we add to the model the ability to download and execute code for extends client features on demand.

To ensure the uniform interface, REST defines 4 interface constraints:

Identification of Resources: A resource is the fundamental element of information in REST, anything that has a name can be a resource.

Manipulation of Resources through representations: According to Fielding " REST components perform actions on a resource by using a representation to capture the current or intended state of that resource and transferring that representation between components. A representation is a sequence of bytes, plus representation metadata to describe those bytes. Other commonly used but less precise names for a representation include: document, file, and HTTP message entity, instance, or variant." In this way we obtain a particular representation of a resource and with well-defined operations passes from one representation to another.

Self-descriptive messages : Acording to Fielding "REST enables intermediate processing by constraining messages to be self-descriptive: interaction is stateless between requests, standard methods and media types are used to indicate semantics and exchange information, and responses explicitly indicate cacheability". In other words, the response should contain all the information required to process the message.

Hypermedia as the engine application state: This is one of the least understood constrains in REST. This constraint just say that every response from the server, must include the set of operations (or URIs) that can be perform over the actual state of the application, making the available operations dependent to the actual state of the application. This is completely different to SOAP that expose all the operations in the WSDL description.

In practice, REST Web service are resource-oriented services that use hypermedia web standards to represent the states of the application, URIs to identify resources and HTTP operations like GET, POST, PUT or DELETE to map the set of operations available, making the HTTP protocol the application layer instead of the transport layer like SOAP Web services.

1.1.3. Pros and Cons for Web Services Aproaches

There has been a long discussion about REST vs SOAP Web Services. On the one hand SOAP services are often criticized for having too much complexity and being too verbose. On the other hand, REST services are criticized by its lack of tools and standardization. We present below a summary of the advantages and disadvantages SOAP and REST as mentioned in (Pautasso et al., 2008).

1.1.3.1. SOAP Strengths

One of the advantages of SOAP approach is its transparency and protocol independence. As a layer of application, the SOAP message can travel in different protocols and

can be understood by a large number of different middleware systems, whether HTTP or not, being able to change in transit.

Another important feature is the advantage of WSDL. With an abstract definition of the interface we are able to separate service specification from communication protocols, specific serialization and implementation details.This way the particular service implementation, or any of the previous layers, can change without affecting the service specification, a big advantage in a world where technology advances and changes rapidly.

Finally one of the greatest advantages of SOAP-WS is the maturity of its tools to build this type of service. Tools built as JAX-WS, Apache Axis[2] or .Net [3] have facilitated the construction of these services in such level that developers can ignore how the SOAP-stack specs are built to simply focus on the construction of the service itself.

1.1.3.2. SOAP Weaknesses

Since powerful tools have facilitated the construction of these services, developers ignore how the SOAP-stack specs are built resulting in incorrect or less than acceptable solutions. (Sessions, 2004).

Given the high expressiveness of the standards WS-* needed for a disciplined and neutral extension of SOAP web services, early implementation had serious problems of interoperability between platforms, particularly the well-known problem of embedding XML into native objects in a programming language, known for being one of the main sources of inefficiency in terms of performance.(Florescu & Kossmann, 2002).

These complexities were introduced to abstract functionality and to obtain a richer and more controlled description of what was happening in every stage of the process, “taking distance” from the web to a complex and highly coupled system(Goth, 2004). Apparently most developers do not need this complexity preferring instead the intrinsic web simplicity exemplified in the famous case of 80/20 rate of use of the Amazon Web Services(Goth, 2004).

[2]See http://axis.apache.org/axis2/java/core/
[3]See http://www.west-wind.com/presentations/dotnetwebservices/DotNetWebServices.asp for a tutorial

1.1.3.3. REST Strengths

According to Pautasso (Pautasso et al., 2008), REST-based services are perceived as simpler than SOAP because all its associated technologies (HTTP,XML,MIME,URI,etc) are well known and ubiquitely supported in a large number of platforms, architectures, programming languages , and systems. In addition, REST uses the default HTTP port which is usually open in the vast majority of locations to access the web. Furthermore, due to low entry barriers, the development can be done with a minimum of tools and the product can be tested simply with a browser.

Because REST does not need a centralized entity to describe the service and because it can take advantage of the native capabilities of cache, clustering and load balancing from mature HTTP technology, REST-based services often have good scalability properties. Finally, REST-based services are not forced to use verbose XML, and can use more compact formats like JSON.

1.1.3.4. REST Weaknesses

One of the most criticized issues of REST-based services is the lack of clear reference to the best practices for development. The ideas behind a "real REST" architecture are sometimes unrealistic in practice and the developers need some way to fix these problems, often generating different views of how it must be implemented. A good example of this is the recent Hi-REST vs Lo-REST discussion (Pautasso et al., 2008).

Unlike SOAP and other standards backed by the W3C, the REST-based web services do not have a comparable level of specification so some issues are still source of discussion. It is important to notice however that REST is still a new concept and the lack of specification and best practices is being addressed. For example last year the first workshops on RESTful Design was carried on with great success.(Pautasso, Wilde, & Marinos, 2010).

1.2. Introduction to Information Extraction

Thanks to the Web, humanity has more information available than ever before in human history. It is impossible for a human beings to read, analyze and process this vast amount of data so we are forced to discard or ignore much of it. This has motivated some researchers to explore techniques for obtaining data from the Web in a more automated manner. These techniques can be grouped into Information Retrieval, Information Filtering and Information Extraction(Cowie & Lehnert, 1996). We will briefly describe these three concepts to digg later more deeply into Information Extraction.

Information Filtering (IF): Filtering Information Systems has several of the characteristics of Information Retrieval, from the data point of view that interacts with them. But information filtering is related to discard data from a major source of information instead of searching on them. The spam filter systems present in most email systems is one of the most popular examples of information filtering systems.

Information Extraction (IE): A more specialized kind of Information Retrieval, searches and extracts structured information from unstructured sources, often using techniques of natural language processing, statistical models and machine learning techniques. According to (Baeza-Yates, Ribeiro-Neto, et al., 1999) "an IE system can then transform the raw material, refining and reducing it to a germ of the original text".

1.2.1. Information Extraction Aproaches

Within Information Extraction there are several approaches to find and organize information. At the beginning, they focused on obtaining structured information from plain text paragraphs using primarily natural language processing tools (NPL) but nowadays much of the information sources come from the web in hypertext format that adds some structure to the documents. In these cases we speak of "unstructured html text" referring to the lack of semantic metadata in the document domain. For these particular cases we

don't always analyze the text at word level but rather tend to look for larger items such as paragraphs, relevant html nodes or components html that capture a concept that help to structure the document.

We present below a short description of various approaches. They are explained in detail in (A. Laender, Ribeiro-Neto, Silva, & Teixeira, 2002).

1.2.1.1. HTML-aware Tools

Information extractors that are based on specific knowledge of the HTML standard to construct extraction rules which takes advantage mainly of the inherent tree structure. The construction of these rules are usually assisted by expert users who include their knowledge of the document through the use of tools. Examples of algorithms are W3F (Sahuguet & Azavant, 2001), XWRAP (Liu, Pu, & Han, 2002), RoadRunner (Crescenzi, Mecca, Merialdo, et al., 2001), Lixto (Baumgartner, Flesca, & Gottlob, 2001).

1.2.1.2. NLP-based Tools

These systems use Natural Language Processing techniques to obtain relevant information from texts written in traditional languages. They include word filtering, part-of-speech tagging , lexical semantic tagging using several strategies from machine learning, expert knowledge in languages, etc. Examples are RAPIER (Califf & Mooney, 1999),SRV (Freitag, 2000),work presented in (Noah, Zakaria, & Alhadi, 2009), etc.

1.2.1.3. Modeling-based Tools

These systems try to extract information from the sources, getting pieces of the structure of information that we sre interested to extract. One must specify first a structural model of the information to be extracted information and then use techniques similar to Wrapper Induction to induce how this structure is presented.The generation of the model is usually made by the user and assisted by GUIs. Examples of work are NoDoSE (Adelberg, 1998),(A. H. F. Laender, Ribeiro-Neto, & Silva, 2002),etc.

1.2.1.4. Ontology-based Tools

Unlike all other previous approaches, these systems, usually called OBIE (Ontology Based Information Extraction), analyze the semantics of the information contained in the documents rather than simply analyze the structure or the format. To analyze the information particular domain ontologies must be provided to infer the objects within the document. Examples of this emerging approach are (Wimalasuriya & Dou, 2009; Li & Bontcheva, 2007; Su, Wang, & Lochovsky, 2009), etc.

1.2.1.5. Wrapper Induction Tools

According to (Fiumara, 2007) a wrapper is a procedure that is designed to access HTML documents and export the relevant text to a structured format. It uses a set of examples to infer the template with which the documents were probably built and then extracts the relevant information. Unlike NLP-based systems these systems use format or information pattern in the documents instead of linguistic constraints, making them more suitable for structured Web documents or documents generated from a database. Examples are (Zheng, Song, Wen, & Wu, 2007; Kushmerick, 2000; Wang & Lochovsky, 2003),etc.

Within Wrapper Induction there is a pattern generation technique using structural alignment of text (used for example in (Chuang & Hsu, 2004)), which is used in bioinformatics for the analysis of nucleotide or protein chains. We present here a short introduction to this subject that we consider important to fully understand our work.

1.3. Bioinformatics and Sequence Alignment

1.3.1. Introduction to Bioinformatics

According to the National Center for Biotechnology Information:“Bioinformatics is the field of science in which biology, computer science, and information technology merge to form a single discipline. The ultimate goal of the field is to enable the discovery of new biological insights as well as to create a global perspective from which unifying principles in biology can be discerned”(Biotechnology Information, 2004).

A definition more appropriate to our needs is given by Luscombe “Bioinformatics is conceptualizing biology in terms of macromolecules (in the sense of physical-chemistry) and,then, applying ’informatics’ techniques (derived from disciplines such as applied math, computer science, and statistics) to understand and organize the information associated with these molecules, on a large scale” (Luscombe, Greenbaum, & Gerstein, 2001).Because the role of macromolecules in the definition is not clear we complement it with the definition of Tekaia: “The mathematical, statistical and computing methods that aim to solve biological problems using DNA and amino acid sequences and related information” (Tekaia, 2003)

In other words, Bioinformatics is a research area that applies multidisciplinary techniques from computer science and information technology to solve issues in the biological area, mainly oriented to problems of nucleotide sequences which, due to its high complexity or size, are impossible to analyze with traditional biology techniques requiring the use of heuristics and automated calculation.

In particular, the alignment of sequences is a bioinformatics topic whose purpose is to analyze and compare structurally chain sequences to find patterns that help understand their structure and functional behavior.

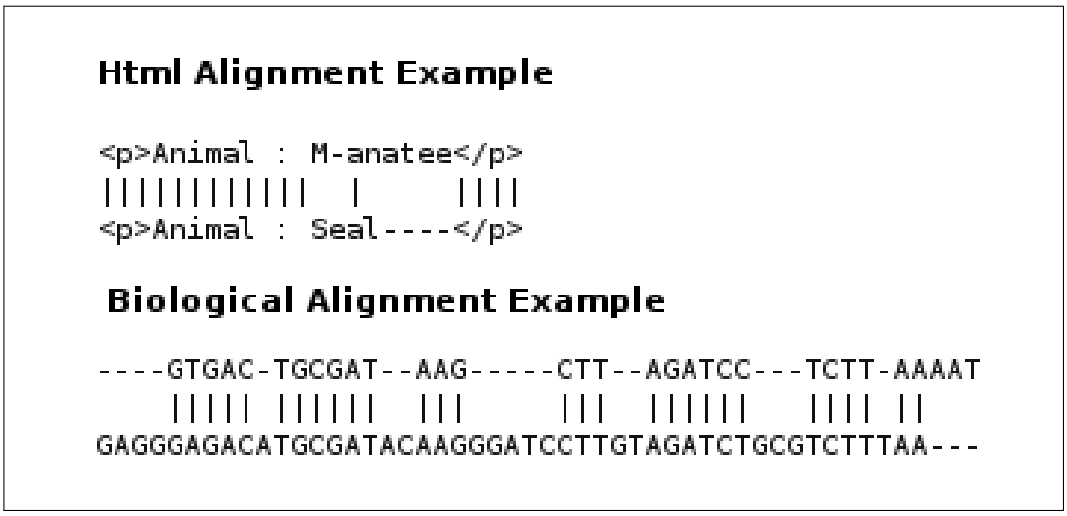

FIGURE 1.2. Image showing example of possible pairwise alignments in the case of information of a web page, or aligning nucleotide

1.3.2. Sequence Alignment and Multiple Sequence Alignment

1.3.2.1. What is Sequence Alignment and Multiple Sequence Alignment

The sequence alignment consists in rearranging two or more sequences (usually represented as characters) in order to find similar areas inside them. When it comes to align two sequences it is usually called “Pairwise Sequence Alignment”. whereas "Multiple Sequence Alignment“(MSA) refers to the alignment of 3 or more sequences.

The alignment is done by adding fill characters or “gaps“ to some of the particular sequences, so that when comparing each ith element with the ith element of the others sequences, a score function that favors the matches and punishes the mismatches or extra gaps can be maximizes. Because of the gaps insertions, both sequences are adjusted to the same size and the similar areas areas are notoriously exposed (Figure 1.2).

We define next the above mentioned concepts in a more formal manner.

1.3.2.2. Sequence Alignment Definition

To formally define a sequence alignment we define first a sequence and some some associated operations.

Definition 1.1. *Let n be an integer, N a finite set of symbols, we define a* ***sequence*** S^n *as* $S^n = \{S_i, S_i \in N \, \forall i, 0 \leq i < n\}$

Definition 1.2. *Let* S^n *and* R^m *be sequences of n and m elements respectively.We define* ***sum of sequences*** $K^{n+m} = S^n + R^m$ *such that,*

$K^{n+m} = \{K_i, K_i \in N \, \forall i, 0 \leq i < n+m\}$

and K_i *is defined as*

$$K_i \begin{cases} S_i & \text{if } 0 \leq i < n \\ R_i & \text{if } n \leq i < m \end{cases}$$

Definition 1.3. *Let N be a finite set of symbols,* S^n *a sequence in N and m, p belongs to the integers such that* $0 \leq m \leq p < n$ *,we define* $S^n_{m,p}$ *a* ***subsequence of S*** *between m and p such that,*

$S^n_{m,p} = \{S_i, S_i \in N \, \forall i, m \leq i < p\}$

Definition 1.4. *Let N be a finite set of symbols, g a symbol* $\notin N$*,* S^n *a sequence in N, j a integer such that* $0 \leq j < n$ *and G a sequence of size 1 with only one symbol g, we define* $INSERTGAP(S, j)$ *a* ***gap insertion*** *at position j such that,*

$S^n + INSERTGAP(S, j) = S^n_{0,j} + G + S^n_{j,n}$

As a shorthand we´ll consider a $INSERTGAP(S, j) + INSERTGAP(S, k)$ as consecutive gap insertion in j and k, such that:

$INSERTGAP(S, j) + INSERTGAP(S, k) = INSERTGAP(S^n + INSERTGAP(S, j), j)$

Next we define a score function needed for alignment

Definition 1.5. *Let N be a finite set of symbols, g a symbol* $\notin N$*, n and m symbols* $\in N \cup \{g\}$*. k,h,l,g functions from* N^2 *to the real numbers,we define the* ***substitution function*** $F_{khlg}(m, n)$ *such that*

$$F(m,n)\begin{cases} k(m,n) & \text{if } m = n \text{ and } m \neq g \\ h(m,n) & \text{if } m \neq n \text{ and } m \neq g \text{ and } n \neq g \\ l(m,n) & \text{if } m \neq n \text{ and } m = g \text{ and } n \neq g \\ g(m,n) & \text{if } m \neq n \text{ and } m \neq g \text{ and } n = g \end{cases}$$

We will call k the match function, h the mismatch function, l the insertion function and g the deletion function respectively.

Definition 1.6. *Let N be a finite set of symbols,$S1^n$, $S2^n$ sequences of n elements, F a substitution function, we will define SEQUENCEALIGNMENTSCORE* ***function*** $SF_F(S1,S2)$ *as*

$$SF(S1,S2) = \sum_{i=0}^{n-1} F(S1_i, S2_i)$$

Finally, we have the foundation for defining an alignment:

Definition 1.7. *Let $S1^n$, $S2^m$ sequences of n and m elements respectively in N, F_{khlg} a SEQUENCEALIGNMENTSCORE function ,G_{1i} and G_{2j} boolean variables with $0 \leq i < n+m$ and $0 \leq j < n+m$, we will define a* ***Pairwise global alignment*** *as*

Argmax

$$\{SF_F(S1^n + \sum_{\forall G_{1i}, G_{1i}=1} INSERTGAP(S1,i), S2^m + \sum_{\forall G_{2i}, G_{2i}=1} INSERTGAP(S2,i)) \quad \}$$

subject to

$$\sum_{i=0}^{i=n} G_{1i} + n = \sum_{i=0}^{i=n} G_{2i} + m$$

$$G_{1i} \in 0,1 \forall 0 \leq i < n+m$$

$$G_{2i} \in 0,1 \forall 0 \leq i < n+m$$

In other words, we look for insertions in each of the sequences so that both are the same size and maximize the alignment score. The final sequences are given by $S1^n +$

$\sum_{\forall G_{1i},G_{1i}=1} INSERTGAP(S1,i)$ and $S2^{\mathrm{m}} + \sum_{\forall G_{2i},G_{2i}=1} INSERTGAP(S2,i))\}$ respectively. Its clear that this definition can be extended easily to multiple sequence alignment.

It has been proven that the optimal solution for 2 sequences alignment can be obtained using dynamic programming and that its complexity is of quadratic order(Needleman & Wunsch, 1970).However, obtaining the optimal solution for multiple sequence alignment is a known NP-HARD problem, exponential on the number of sequences(Konagurthu & Stuckey, 2006),making necessary to find heuristics to reach good suboptimal solutions in a prudent time.

1.3.2.3. Aproaches to Multiple Sequence Alignment

Among the different heuristics to solve the Multiple Sequence Alignment problem are several approaches to complete the task. Below the most relevant:

Dynamic Programming Methods: They do not use any heuristic and obtains the optimal solution by an efficient search on the lattice of possibilities. The Carrilo-Lipman method (Carrillo & Lipman, 1988) proposes a boundary search polyhedron to obtain close to the optimum solutions, improving the performance well enough. However this technique is still prohibitive for cases which are not simple enough to execute it.

Progressive Methods: Also called hierarchical or tree-based methods, consist of establishing a sequence hierarchy to then align sequences pairwise according to this order until the final alignment include all the sequences. The order or hierarchical tree is usually constructed based on phylogenetic trees obtained with techniques such as neighbor-joining(Saitou & Nei, 1987) or probabilistic methods such as MrBayes(Huelsenbeck & Ronquist, 2001). This is one of the most popular methods because it achieves acceptable results in polynomial time. One of the most representatives algorithms of this approach is ClustalW (Thompson, Higgins, & Gibson, 1994), which uses neighbor-joining and substitution matrices to obtain quickly a good quality result. Another algorithm quite used to give

better quality solutions at the expense of speed is T-Coffee(Notredame, Higgins, & Heringa, 2000).

Iterative Methods:] Unlike the progressive methods in which the solution often depends too much on the initial order that the sequences incorporate into the result (once included the resulting alignment is not rearranged), the Iterative Methods progressively include sequences to the solution while making iteratively changes on it, making the solution more accurate but also slower. A popular example of this approach is MUSCLE (Edgar, 2004),which first builds a progressive alignment to then refine it iteratively calculating subtree alignments and incorporate or discard them depending on a maximization function, until a local optimum is reached.

Machine Learning/Artificial Intelligence Methods: Machine intelligence techniques have been applied to achieve this problem with completely different approaches to the traditional ones. Among the techniques used are genetic algorithms(Notredame & Higgins, 1996), boosting (Parker, Fern, & Tadepalli, 2006), Hidden Markov Models (Eddy, 1995), etc.

Motif Methods: Consists of searching for patterns within the sequences either to improve the quality of alignment of another method or get more appropriate substitution matrices. Most of the techniques shown above have been enhanced through this search for patterns, being an example (Bailey, Williams, Misleh, & Li, 2006), which uses Hidden Markov Models.

1.3.2.4. Global Alignment vs Local Alignment

So far we have mentioned concepts of alignment considering only global alignment, which finds a solution that maximizes the score for the entire sequence. On the other hand there is the local alignment, which consists of obtaining the optimal alignment for a particular segment of each sequence, regardless of the rest.

For example in Figure 1.3 shows a example being aligned by global alignment first and then by local alignment. Using the former, the entire sequence is adjusted to obtain the best score on a global basis, and the second one finds only the more similar portion of the sequences to get the best score, ignoring the rest.

The local alignment is useful when searching for specific similarities in the sequence and when obtaining a common structure to find functional patterns throughout the sample is not expected. Instead, the global approach is more appropriate when we expect to find a common structure or ancestor inside the sequences, i.e a structural pattern. Representative local pairwise alignment algorithms are Smith-Waterman(Smith & Waterman, 1981) and BLAST (Altschul, Gish, Miller, Myers, & Lipman, 1990).

Following the same previous nomenclature, we define more formally the local alignment.

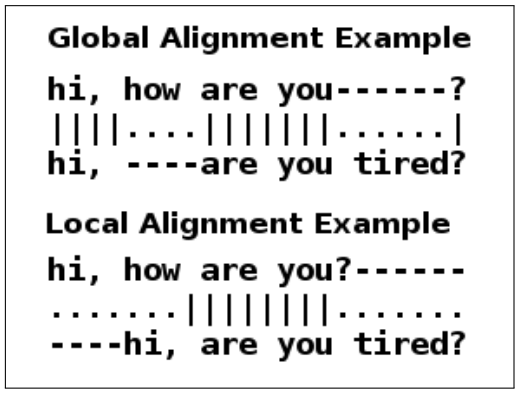

FIGURE 1.3. Image showing example of a possible global and local alignments for a pair of sequences

Definition 1.8. *Let $S1^n$, $S2^m$ sequences of n and m elements respectively in N, F_{khlg} an SEQUENCEALIGNMENTSCORE function,G_{1i} and G_{2j} boolean variables with $0 \leq i < n+m$ and $0 \leq j < n+m$, we define a* ***Local Pairwise Alignment*** *as*

Argmax

$$\{SF_F(S1^n_{a,b} + \sum_{\forall G_{1i}, G_{1i}=1} INSERTGAP(S1^n_{a,b}, i), S2^m_{c,d} + \sum_{\forall G_{2i}, G_{2i}=1} INSERTGAP(S2^n_{c,d}, i))\}$$

subject to

$$\sum_{i=0}^{i=n} G_{1i} + b - a = \sum_{i=0}^{i=n} G_{2i} + d - c$$

$$G_{1i} \in 0, 1 \forall 0 \leq i < n+m$$

$$G_{2i} \in 0, 1 \forall 0 \leq i < n+m$$

$$0 \leq a \leq b < n$$

$$0 \leq c \leq d < m$$

For the pairwise case, the optimal solution can be found in quadratic time with respect to the sequences(Smith & Waterman, 1981). However, the local multiple case is much harder to solve than its global counterpart, being shown as a APX-HARD problem. In other words, unless P = NP, there there is no polynomial time algorithm (Akutsu, Arimura, & Shimozono, 2000).

Finally, it is important to emphasize the role of substitution matrices and scoring functions to get a good alignment. In the study of amino acid or DNA sequences, there are matrices that determine the similarity between amino acids such as BLOSUM(Henikoff & Henikoff, 1992) or PAM (Dayhoff & Schwartz, 1978) which helps to improve the quality of alignments by adding bias that include knowledge about the domain in order to obtain meaningful biological results.

These tools, represented in our modeling in the definition 1.5, stand for a fundamental tool to add bias to the alignment and thus transmit expert knowledge to the result. For example, we can make symbols behave like wildcards, having a good score with a several

different symbols, or we can align other symbols only when strictly matches, penalizing otherwise.

Although they are usually represented by matrices, these tools can be modeled with any function even including other parameters such as position in the so-called position-specific scoring matrices (PSSM),which may give different scores of substitution depending on where the symbol is located within the sequence, making able to model more complex knowledge, such as scoring at the level of words, regions, etc. An example of using PSSM is PSI-BLAST(Altschul et al., 1997),where BLAST is improved by including this type of heuristics.

1.4. Proposed approach to Information Extraction for Web Service Automatic Generation from Web Sites

As mentioned at the beginning of this document, there are many sites that are not updated to the latest trends of integration and do not provide formal interfaces to access their services. On the other hand, access to the information without standard interfaces usually require complex software to build and very difficult to maintain. An easy way to create web services for these sites would encourage and facilitate the integration of applications and information on the web.

The objective of this work is to obtain a general model of information extraction using only the information that can be obtained by browsing the site, in order to be closer to the fully automatic construction of web services. We will present a general framework that includes the different stages in the construction of this system and then we presenta a prototype that implements the first stages of the process to demonstrate de feasibility of the proposal.

In the chapter 2 the work is presented in detail, including scope, methodology and finally the results we obtained with our implemented prototype. Finally, chapter 3 provides further analysis and discussion from a general perspective and the directions of future work.

Chapter 2. Unsupervised information extraction from Web sites using bioinformatic techniques

The following chapter is a paper, submitted for publication in the Journal of Web Engineering.

2.1. Introduction

The web has become more than a place for sharing documents and images. It can now be seen as a platform to expose, communicate and integrate applications and information. At present, due to the large amount of information available on the Web, there has been a greater emphasis on applications and services that can coexist and integrate themselves with these massive data sources.This has made that providing integration tools becomes more a need than just a desired feature.

Many modern web sites provide web interfaces to expose their capabilities to other third party applications that require it, and at the same time new software projects use the web as the official channel to communicate with other applications in an environment, leaving behind complex integration tools and standards such as CORBA (Group, 1999), COM[1], etc.

Web Services rise as the response to the use of Web interfaces for information exposure; these services are usually categorized into two groups: Big Web Services and REST Web Services (Pautasso et al., 2008). Both categories, although with different approaches and methodologies, share the goal of using web standards for application communication. A lot of progress in standardization and ease of use has occurred in this area in recent years, which has resulted in the widespread adoption of this technology, turning it into the de facto standard for communication between applications.

But in spite of the fact that the Web is evolving, while some companies and institutions are updating their sites with current technology trends, many sites have not been updated,

[1] See http://www.microsoft.com/com/

leaving a large amount of useful information beyond the reach of this new world of easy integration and use. This has forced users who need to connect to these sources to build complex, hard-to-mantain extractors that capture information in non-standard ways.These represent additional barriers to transforming the Web into a more intelligent and integrated one.

It thus becomes important to build automated tools for building Web services. Significant integration efforts have been made to collect and extract information in an automated way (Fiumara, 2007; Wimalasuriya & Dou, 2010; C. Chang, Kayed, Girgis, & Shaalan, 2006). These systems,however,focus on getting the information itself rather than the data model behind that information; this makes the task of building the associated service almost impossible, since for this purpose it is not just the information that matters but also the entire context.

We propose a model and methodology for information extraction that, besides obtaining the wrappers and the pertinent information, builds a model of the website itself, preparing for future stages of automated Web service building. This new approach combines properties present in hypertext documents, with techniques borrowed from bioinformatics, to obtain structural patterns and to detect the hidden features.

The rest of the paper is organized as follows: In Section 2.2 we provide some context and background material including the main stages needed to generate a web service and the rationale behind the decisions we made for our system. Section 2.3 describes the details of a prototype implementation, and then in section 2.4 we present the results obtained after running a series of tests over the implemented system. Finally, in Section 2.5 we present the conclusions and future work.

2.2. Context

2.2.1. From a site to a service: Theoretical framework

Automated synthesis and generation of web services from a web site is not an easy task. The process can be separated into the following stages: information gathering, data

analysis and model building, analysis of relationships or methods, and finally construction of the service itself. The figure illustrates the different activities in each stage.

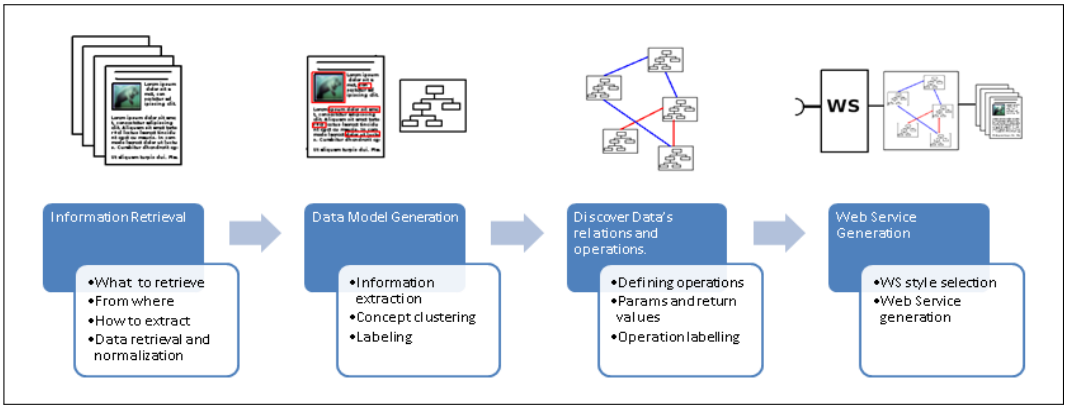

FIGURE 2.1. Main stages of the building process

2.2.1.1. Determining the information to be exposed.

A Web Service is an interface that presents information and functionality to be consumed by automatized processes. This is different from web applications or web sites, where the appearance and presentation of information is also added, so human users can more easily comprehend the content. Since a web site provides much more information than is required by the equivalent web service, it is necessary to determine and decide what information the service will expose.

We can make a web service that exposes only standardized data, or one that also provides the functionality (operations) available for that information. It is even possible, although not very widely used, to build a service that also exposes the presentation of the information, such as colors or styles. We might also want to avoid advertising information, which is usually present in headers and footers, so as to focus just on the core information.

The degree of completeness of the information required to build the web service depends on the needs of the users, so it may be different for each particular case. In the area of information extraction, different levels of completeness of extrated information

have been covered: some capturing only selected information as "relevant" (Wang & Lochovsky, 2003), whereas others capture the complete information of the site (Zheng et al., 2007).

2.2.1.2. Collection, normalization and data preprocessing

The Web is a very noisy place to capture data. Information transmitted through the network tends to have inconsistencies or errors in practically all layers of the OSI model. Even considering only the application layer, we can find significant flaws in the extracted documents. This does not include consideration of the anomalies that can be contributed by the final information itself. It is therefore necessary to decide what information we aim to extract (eg complete HTTP packets, only the payload of the message, only the contents of the body tag, xml markup ignoring markup attributes, etc).

2.2.1.3. Capturing the required information

The information can be captured in different ways. Probably the easiest way is to manually capture it and provide it to the parser algorithm. There are other more automated methods, such as saving an actual navigation session in a browser, packet sniffing at the target site or in-depth investigation into the links of the site, by changing the call parameters using hidden web concepts as presented in (Raghavan & Garcia-Molina, 2001)

The level of automation of the extracted data often determines the quality of information. For example, a manual approach could also allow us to classify and label information that is considered important(becoming a supervised approach). On the other hand, a fully automatic approach can have more noise, but it could also be more abundant and cost effective in terms of both time and effort, making the unsupervised approach more attractive.

Information extraction algorithms, in general, assume that the information is already available and in many cases also labeled, as shown in XWRAP (Liu et al., 2002), RoadRunner (Crescenzi et al., 2001),(Chuang & Hsu, 2004),etc. In the case where information searching is a fundamental part of the process, we are talking about Information Retrieval(Baeza-Yates et al., 1999), as seen in (Noah et al., 2009) or (Capasso, Cesarano,

Picariello, & Sansone, 2006). Finally, there are algorithms that take into account the entire combined process to extract information such as in (Zheng et al., 2007).

2.2.1.4. Grouping and labeling information that represents the same concept

After or during the extraction of information, it is necessary to group together information that represents the same concept in order to obtain a more general model of the data. For example, in an html table, it is possible to capture information that appears in each row; however, if these rows represent different entries of the same information, it is necessary to group them together. Information captured in different instances of the same page but representing the same concept in each one of them should also be grouped. Each of these groups of information represents a possible useful concept to the user, and in addition to extracting the value, it is necessary to label it. Different strategies can be used for the labeling process, such as determining the type of content (a date, number, etc), finding common concepts in the grouped information, or simply reviewing "neighboring information" close to the concept in search of explicit labels.

There are papers citing different approaches, for example IEPAD(C.-H. Chang & Lui, 2001), DELA(Wang & Lochovsky, 2003) and TTAG (Chuang & Hsu, 2004) group information together and wraps discovered patterns using text alignment. (Su et al., 2009) or (Wimalasuriya & Dou, 2009) make use of ontological information to extract data.

Examples of papers with labeling of information include the automated hyponyms search (Da Silva, Barbosa, Cavalcanti, & Sevalho, 2007), custom rules using expert knowledge of html (Wang & Lochovsky, 2003), and others that use information already labeled to create new relationships (Wong & Lam, 2007), etc.

2.2.1.5. Obtaining higher-level concepts.

After we have labeled information groups, we can increase the quality of the objects by finding semantic relationships between groups. In a table, we understand that each of the components of a row can be encapsulated as something that in itself means something, but the set of rows or the table itself may also have an interesting semantic meaning for the user. For example, figure 2.2 shows 4 different attributes of an animal. Capturing these

4 values may be relevant to the user, but it would be even more useful if these 4 values were encapsulated and were related to the attribute table which mentions the name of the animal.

Hence, it is important to capture not only attributes or concepts, but also the relationships between them. This is the part in the process where developing new tools to improve the semantic level of applications is of vital importance. For example, using the Ontology Based Information Extractors (see (Wimalasuriya & Dou, 2010) for a survey), directly on the information or, better yet over preprocessed semi-structured information, we can include expert semantic knowledge in our system, improving the quality of the captured attributes by adding the relationships between them, or finding higher levels of generality.

2.2.1.6. Discovering the operations from the data

Once we have concluded the processing of attributes or concepts included in the service, we can also determine the features that the original site had. For HTML, the gateway to the functionality is usually within the same page as the links and forms. Since these functionalities are fully described in the page itself, it is possible to find and analyze them so as to include this in the generated service

2.2.1.7. Determining the actions to be exposed

One thing is the actions that the site offers; another is which of them are going to be exposed. As is the case with data, this will depend on each user. Here are some possible examples: get only links that redirect to pages within the same domain host, get only links with parameters that vary from page to page, get links with a variable component, etc.

2.2.1.8. Determining and labeling the mandatory and optional parameters of each action

For every operation found, we must determine the mandatory and optional parameters to successfully execute the operation. The information used to determine these parameters is strongly related to the data source we are using to do the analysis. For example, we can find much more valuable information with complete information of an HTTP session,

versus only the html pages, since we could completely rebuild the GET or POST messages. The labeling of the parameters should be much easier than the labeling of attributes,as they tend to be further away from the user and in a format which is easier for the software to interpret.

2.2.1.9. Labeling the operations and determining their return objects

The next problem is to determine the name we are going to give to each of the operations found. Finding this name is not as simple as finding a parameter's names but it is still easier than tagging attributes. Since the machine will require an unambiguous connection to the application, we have better quality information to find these labels.

2.2.1.10. Building the final service

After getting the attributes and functionalities for each page, considered as an object, we use this information to build the service. Because the most popular tools in modern platforms like J2EE and .Net are based on metadata and annotations like JAX-WS[2] and .Net (Ballinger, 2002), building the web service itself using the information collected through the previous stages is not difficult to implement.

Note that we did not have to make an early decision about the architectural style of service, whether it be REST(Fielding, 2000) or SOAP(Gudgin et al., 2002). We can make this decision at this final stage. For example, under a REST approach, we see each page as a resource, exposing the attributes in each state and using the operations or links as connections to move from one state to another. Choosing a SOAP approach instead, we group all functions as an interface, and in each case returning an object representing the page with attributes collected. For a more complete discussion of different approaches, see (Pautasso et al., 2008).

It is worth mentioning that all stages are not strictly necessary, and they also do not need to be in this strict order. In fact, several current extraction algorithms that implement some of these stages in a different order could still be used for our purposes.

[2]See http://jax-ws.java.net/

The focus of this work is on the extraction of the data model which we considered to be the key issue, leaving the last part of the process for future research.

2.2.2. BIOIE Theoretical Framework

In the domain of Information Extraction, there are several well-known algorithms that could be used for the task. Some of the most general can be used to capture attributes and labels. Most of these algorithms, however, end when the information has been collected.This makes it very hard to extend the system so it can use this information to generate a service.

In this paper, we present an algorithm that takes into account that the final objective is not just to collect and analyze onformation from web pages but rather the automatic generation of a web service. It collects and analyzes information from web pages to get all the necessary information to identify and label the possible attributes of the model, leaving enough information for an analysis of the next stages of the investigation.

The proposed algorithm is a Wrapper Induction algorithm with unsupervised learning, which requires the input of several instances of the same website as training information. It is inspired by the TTAG extraction system (Chuang & Hsu, 2004), but with special modifications making it unsupervised and also more efficient through the use of our knowledge of HTML documents. We also borrowed a few general ideas from the area of bioinformatics. We first present the rationale behind the decisions and later we discuss the final implementation.

2.2.2.1. Structure of a hypertext document: two different recognizable zones

A typical hypertext document can be seen as text content that represents a resource. This content has references (links) to other resources within its own content, so you connect resources between them.

HTML and other standard formats for representing hypertext focus on describing and structuring the document solely from the point of view of structure and some document

presentation, but give no clues oriented to understanding the semantic content of the document; this responsibility is left entirely up to human interpretation. There are ongoing efforts to increase the semantic capacity of web (see a short introduction in (Shadbolt, Hall, & Berners-Lee, 2006)), but its adoption is not broadreaching yet and much remains to be explored. The implication of this reality is that most of the crucial information to shape and structure the semantics of hypertext documents today comes not from the overhead associated with the specific hypertext format used, but from the content itself.

We will use the term ”Context Information” to refer to the information associated with reserved words or special instructions of the hypertext format outside the domain of information that is intended to provide. Alternatively, we will use ”Domain Information” to refer to information within the content intended to be provided in the document.

Figure 2.2 is an example of a document showing context and domain information. Note that if we consider only context information, we have no clue as to what the document is about. On the other hand, if we consider only the information domain, we have some idea about the document, but it is not completely clear. In this case, domain information not only contains the information that interests the user, but it also contains the meta-information and presentation information that only has real meaning in the context given by the structural “Context Information” and after human interpretation. We also note that the context information is not completely useless because it give us some basic semantics clues. For example, the title, indicating that the information represents a table, etc. In the next chapter, we will use this distinction in the sections of the document to take advantage of certain properties that HTML pages often have.

2.2.2.2. Wrapper Induction Approach

In a web site or a web application, not all of the source text represents relevant information to be consulted through a service. In general, a service will provide the information in a manner that is as summarized, simple and structured as possible, avoiding any content oriented towards the aesthetic aspects of the document. Also, since the context is given

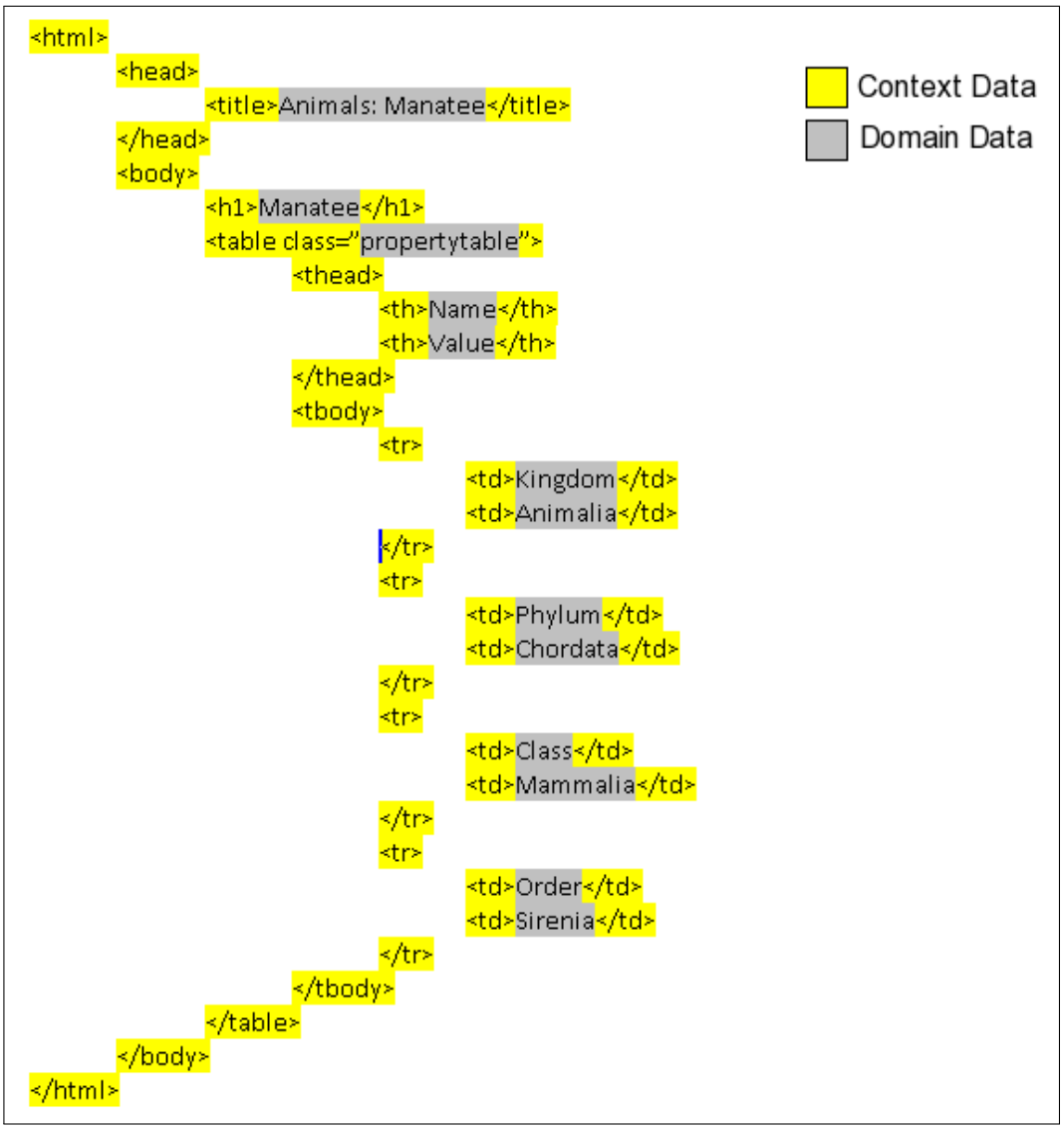

FIGURE 2.2. Context Information (marked yellow) in this case XHTML code vs. Domain Information (marked gray), which in this case refers to an animal.

by the parameters of the request itself, the information given by a service is often not contextualized. Hence, there are some zones of the document that are necessary to detect and extract so as to build a service and many others that should be simply discarded since they would not provide any useful value to an application that would use it.

In general the relevant information is the dynamic information inside the document, specifically information that changes from instance to instance that usually comes directly

from a database. But even when this information is what we want to deliver to the final user, it takes more than that to build the service. The grouped relevant information can help to obtain the needed semantics but cannot give us any structural information about the document and how the different parts of it relate to each other.

On the other hand, the static information of the document is usually there to give context to the dynamic part, to present the content or simply to connect and organize it. This static content is usually our only clue about the original layout of the application, and it is important to determine most of the metadata and the structure of the relevant information.

Thus we need an information extraction (IE) method that does not immediately crop areas of the document, because unlike other extraction scenarios (see (C. Chang et al., 2006) for a survey) we not only care about the information itself, but also about the structural and semantic information required to create a useful service. Furthermore, because it is very difficult to obtain labeled data in this kind of process and there is a huge wealth of unlabeled information available, a unsupervised approach seems the most appropiate one.

Therefore, we used an unsupervised wrapper induction approach for data mining. We assumed that the web pages were based on a template and the training information needed to build the template would not be labeled, so it would not show us the location of the information we are looking for.

Not all websites are built from a template, but the ones we were initially interested in, with an information query profile, are generally built to access the contents of a database and use layouts to present data to end users.

2.2.2.3. Bioinformatics meets the Web: A Web page as a sequence of DNA.

If we look at the most concrete nature of a web application, we could define a page of a web application as a text generated by software that uses database and a template provided by the programmer to answer a particular query using standard protocols of communication for direct consumption of users. The text describes a particular instance of the information that the application can answer. For example in Figure 2.2we could

say that this website is the result of a query on manatees, but the application could also be prepared to answer questions about many other animals or other items. The templates used by a web site could generalize the content and functionality of a web page, since they somehow represent the generalized visual perception of the real model made by the developer.

So we are interested in finding the general template behind a website. This template represents the ancestor of all instances of web pages for that particular template, and it give us valuable information to understand the target document. But finding the ancestor of a species from the code that represents it is a subject that has been studied for decades in Biology, where advanced and mature techniques are already available. The field of Applied Bioinformatics uses knowledge in computer science and statistics to obtain relevant biological information from a large database of molecular information. Sequence analysis of proteins or DNA/RNA are performed to recognize evolutionary, functional or structural patterns, discover ancestors, or find patterns that lead to an understanding of the function of particular genes. If we view this from a broader perspective, there is no major difference between trying to find the template of a document from observed instances, and looking for the ancestor of a species by observing genetic information.

The use of text alignment to extract information from document has been previously explored in ((Chuang & Hsu, 2004; Wang & Lochovsky, 2003; C. Chang & Kuo, 2005), etc.), but it has focused on testing specific algorithms, with little o no concern about how to get the most out of the alignment to obtain a higher quality result and significant semantic meaning.

The alignment of sequences is a way to rearrange 2 or more sequences of characters, adding or removing spaces, in order to maximize a score function that gives a higher value to better matches (common subsequences) among the sequences.

There are 2 approaches to align sequences: global alignment and local alignment. Global alignment consists in obtaining the best global fit over the two complete sequences.

Local alignment looks for the best match for a specific area of the sequence, regardless of the rest.

It is important to understand the differences and limitations of the global and local approach to obtain a meaningful result. For example, if there is a major difference in the sequences and you are only searching for a small common componentsuch as finding a genetic disease, a local approach may be more appropriate. If there is a great similarity between sequences and we are looking for general structure, a global approach makes more sense.

In sequence aligning, it is also important to consider the cost matrix or substitution matrix, which puts a numeric value on how good or bad the match is between 2 characters in the final alignment. Researchers in Bioinformatics have built empirical cost matrices that achieve relevant results for biological research such as BLOSUM (Henikoff & Henikoff, 1992) or PAM (Dayhoff & Schwartz, 1978). Going back to the web senario, although we cannot directly use the tools designed for nucleotides and proteins, we have an importat factor playing in our favor: the sequences of web pages are built by humans

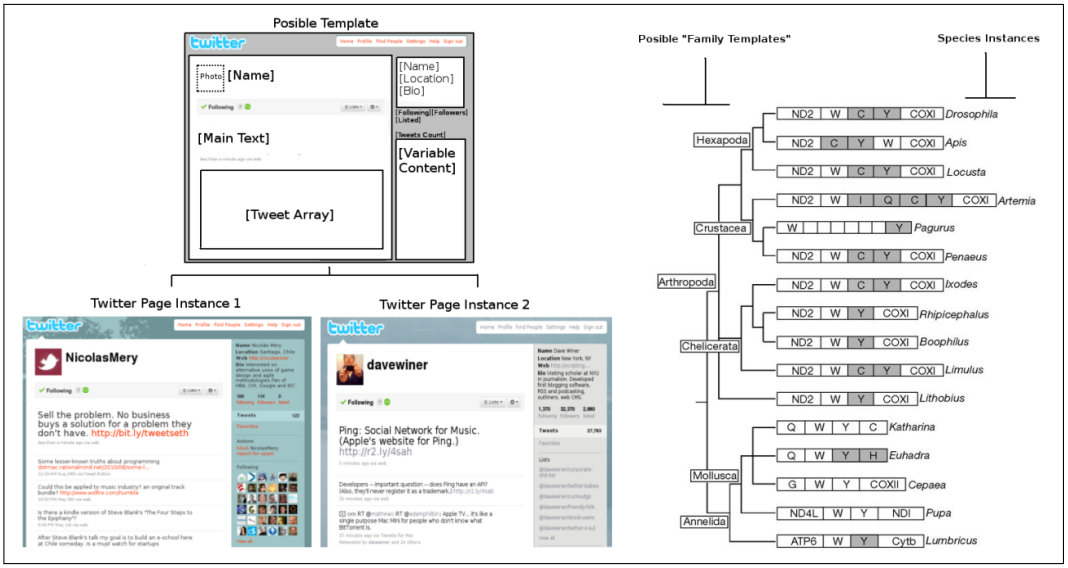

FIGURE 2.3. The image is divided into two parts: On the left side are two instances of twitter web and a template that could be inferred from them. On the right, we see the classification of several species of arthropods from their tRNA.

so we can interpret the meaning of each of them. Taking advantage of this fact, we can design matrices of substitution that increase the semantic level of alignment, by for instance encapsulating the context information keywords, the domain information words or any other semantic relation that we find in a document.

2.3. Prototype Implementation

As a proof of concept, we decided to build a prototype that would implement all the ideas explained in the previous section. We describe the design and implementation of this prototype here.

2.3.1. Data preprocessing

As mentioned before, the content of web pages presents a lot of noise at various stages, including the HTML layer. Many times the source code of the pages is HTML that does not meet standards, by marking incompleteness, inclusion of proprietary content, invalid characters, etc. The processing of the next steps requires the strong assumption that all pages are well formed XML.

To normalize all the training data coming from the web (html classic, non-validated xhtml, etc.), we use HtmlCleaner[3] which transforms each document into its closest valid HTML form. We also remove all unnecessary whitespace text, and set the order of xml alphabetically for greater stability. Because the information is not exactly the same as that originally obtained from the real servers, the testing stages must also pass through this filter.

2.3.2. Data Classification

Based on the ideas presented in section 2.2.2.1, we first classified the content of the text into context and domain zones. The domain was considered to be all text that represents values of xml attributes and plain text nodes present as children of xml tags. The rest of the markup obtained was considered to be a context zone.

[3]Available in http://htmlcleaner.sourceforge.net/

The main reason why the document is separated into two areas is because, as a sequence to be aligned, they exibit different properties. Contextual information has a high probability of a match, as this information is usually provided by the template, and the existence of this template is given by assumptions about the nature of the sites. On the other hand, the domain information, which is usually self-generated or obtained from a database, is highly unlikely to match in different instances of documents. The only elements that are likely to match in this area are metadata that is presented as part of the template to contextualize the information, but this information is a small percentage of what we expect to see in this section of the document. This analysis suggests performing a global alignment in the case of context information (in search of a common structure) and a local alignment in the case of domain information (searching for only small similarities).

Another reason for the separation is performance. Because the most commonly used alignment algorithms are usually of quadratic complexity (Smith & Waterman, 1981), they have problems when the number of sequences is too large (in the case of HTML documents that may have thousands to tens of thousands of symbols). Since it is possible to separate the document into subsequences and then bring them together in linear time, a larger number of separations will provide a considerable performance gain.

Finally, the quality of the alignment improves if we locally narrow the hypothesis space of the solution. If you have an idea of the parts of the document that are equivalent and the parts that are not, we can use this information to prevent the alignment algorithm from finding patterns that only occur by chance.

After separating the different parts of the domain from the context, we proceed to align and to generate the basic pattern of the area of context, as explained in the following stages.

2.3.3. Context Alignment

2.3.3.1. Context Preparation for Alignment

Before aligning the sequences, we compressed them, mapping words to another dictionary of smaller words, allowing the aligning process to work with sequences of reduced size. Besides the obvious performance gain obtained at the time of alignment, a smart selection of dictionaries can help to improve the alignment quality through the use of specific substitution matrices for each dictionary. When aligning nucleotides, the substitution cost matrices are of primary importance in obtaining relevant scientific results. For example, the BLOSUM62 substitution matrix is constructed empirically to obtain good alignments of amino acid sequences. But unlike what happens in biology, the sequences we have to analyze are built by humans and we have absolute certainty of their semantic meaning, so there is no need to build an matrix empirically, but instead directly use our knowledge of the substution matrix to obtain the expected alignment.

The context information was first compressed at the word level. Because transforming each word into a new sequence of size 1 involves losing the similarity information between words, we correct this by inserting into the substitution matrix a cost proportional to the difference in size of the words. We call this substitution heuristic Cost Matrix Size.It does not generate alignments which are as perfect as those achieved without compression, but it produces good results since it avoids the separation of indivisible words due to common substrings. In our experiments, we have seen a reduction of the size of the sequences from 40 to 60 percent of the original size.

Having compressed the context zone, the alignment to obtain the structural template for the document began. Unlike domain information, we know that the context information, in the case of XML, is represented by a tree. For reasons of performance and to narrow the size of space hypotheses, we separate the document as much as we can at the time of alignment. Taking advantage of the tree structure, we first align the root node, and then go recursively aligning the children nodes. Another advantage of this fine-grained

separation is that the alignment of small sequences tends to become very common, making it possible to cache alignment results which produces a huge performance gain.

This idea of recursively aligning documents had been explored before in TTAG to help build more general templates. However, TTAG does not separate the different parts of the document, and considers only the names of elements, without including information about attributes and without full use of the acquired knowledge of html documents. The result is an alignment that is less reusable and of lower quality.

In the next section, we will see how the alignment itself is carried out, and how the Base Pattern is constructed.

2.3.3.2. MultiTreeAligner

For the alignment of each node in the area of context, we used a custom multi-aligner based on Clustal(Thompson et al., 1994) that we call the MultiTreeAligner (MTA) and its operation will be explained below:

(i) Using a progressive approach, we aligned each sequence pairwise, using the global alignment algorithm of Crochemore(Crochemore, Landau, & Ziv-Ukelson, 2002), more specifically, a modified version of the NeoBio toolkit[4].

(ii) Then, we constructed a distance matrix for each sequence alignment using the Levenshtein(Levenshtein, 1966) using the edit distance as the parameter of measurement.

(iii) With the distance matrix, we constructed a phylogenetic tree using neightbor-joining(Saitou & Nei, 1987) to determine the order in which the sequences are progressively aligned (particularly the NINJAS(Wheeler, 2009) library was used).

(iv) Finally, multiple alignment was done using NeedlemanWunsch(Needleman & Wunsch, 1970), in particular a modified version of the NeedlemanWunsch implementation of the NeoBio toolkit

[4]Sources available http://neobio.sourceforge.net/

As we did the alignment using the XML tree hierarchy, we needed to keep this structure in the information when browsing through the nodes, so the compression won't alter this. However, at the moment of the alignment of the nodes themselves, it is no longer necessary to maintain this structure and stronger compression can be applied, further reducing the size at the expense of not being valid XML.

Applying a new layer of compression gives us the opportunity to build a substitution matrix tuned to what we know about this part of information, making the aligner include different substitution costs depending on the position and context of the symbol. For example, a character representing an xml attribute, can be never found before the element name, and this name cannot be divided or moved. More sophisticated information can be incorporated as XML attributes that can only go within a certain markup, no matter how similar it is seen as a sequence. The substitution matrix that incorporates all this knowledge is called the "Cost Semantic Web Matrix" and its mission is to prevent matchs that make no sense in a XML/HTML document.

Once the child nodes of the node currently under review are aligned, we proceeded to decompress the strongest compression layer to return a xml representation of the node and then rerun the algorithm on the children of each of the nodes found. For each aligned item, it was necessary to build the Base Pattern which is then attached to the rest of the constructed sequence until all nodes have been aligned and the Base Pattern represents the initial template for the complete document.

2.3.3.3. Base Pattern Construction

Having received an aligned text, it was possible to construct a pattern that represents this information in a single sequence. The idea (mentioned in TTAG) is quite simple:

Over the aligned sequence of M sequences of size N, a vertical sweep is carried out to build a sequence of size N, such that for each position i<N in the pattern, G is the character that represents a gap $x_{i,j}$ is the character at position i of sequence j, we place a character C_i such that:

$$C_{\mathrm{i}} \begin{cases} x & \text{If } \mathrm{x_{i,j}} = \mathrm{x}\ \forall 0 < j < M \\ x? & \text{If } \mathrm{x_{i,j}}\ \in x, G \forall 0 < j < M \\ * & \text{otherwise} \end{cases}$$

where x? is an optional x character and * a character that represents a wildcard. We call this construction of the Base Pattern. We will later perform further operations on these sequences.

2.3.4. Aligning and Grouping Domain Zones

Once the context zone is aligned, we started to group and align the various domain areas and then re-joined to the Base Pattern representation of the document so far. Each of the domain areas was grouped according to where they should go in the context Base Pattern, using as a reference the prefix of the context information in which the domain zone was found and the position of that context prefix in the Base Pattern.

After the domain fragments are grouped, we proceeded to align the clusters. Unlike the context, we knew nothing about these sequences, as they tend to be plain text. The only thing we knew is that they are sentences composed of words, so we just used a word-level compression and an Size Cost sustitution matrix heuristic.

To align, we used a local approach to help us find words that are only part of the template, probably to give context or clarify semantics for the remainder of the information domain, leaving the rest (the real encapsulated information) as wildcards.

After each of the domain clusters were aligned, we constructed a Base Pattern for each of them to be incorporated into the BasePattern of context in the positions previously found. The patterns should be incorporated in a descending order of positions to prevent changes in the position at the time of insertion. When all patterns had been incorporated into the initial BasePattern, we got our first version of the template.

2.3.5. Increasing the generality of the Base Pattern

The Pattern obtained so far represents the template of the document at a word or character level, but does not yet reflect the data model behind the document. So we need to inspect the Base pattern to find the variables and relationships needed. One of the first things we did was to unite all the wildcards that appear together in sequence within a single wildcard, to represent these sequences full of information. These wildcards will later be transformed into the possible variables of the model. Furthermore, this reduction in the size of the pattern helps to improve the performance of subsequent stages.

Then we looked for repetition of patterns within the sequence, for example a row of a table is repeated many times and this is represented by the cardinality of the variables. To discover these patterns, we used the same algorithm presented in TTAG, which uses 2 heuristics to find optionality and cardinality of the information.

Unlike the relatively small documents used in TTAG, however, our documents are much larger, and the solution given by TTAG tends to collapse for documents of more that 400 characters. Since Web documents are usually two order of magnitude larger, the solution is not acceptable. To adjust for this, we modified the procedure in order to run the current algorithm at node-level, starting from the leaves and then climbing up a in the tree hierarchy reusing the sub-results obtained (in a Dynamic Programming fashion). Thus we encapsulate the cost explosion of the algorithm at the node level. Because itś unusual to see nodes with so many children in the documents, the performance is now much better.

2.3.6. Sibling Merger

Until now the system was able to detect patterns and variations that can be identified when compared with other states of the same document, but not when the variability was produced in part by simultaneous variations within document itself and between documents.

The most common case is what happens with the rows of a table of variable size. Suppose you have several documents that have the same table but with a different number

of rows. The algorithm would detect that there is a minimum number of common rows and another optional group. Only if this information is structurally the same will it recognize a variable number of clusters that represent the same row. In contrast, if there are some differences in the rows and they are not structurally the same, it will recognize a document with N different variables, but similar in content within the same domain of information. As this scenario is very likely to appear, it needs to be handled; it must be recognized when it occurs to produce an alignment between the sibling nodes involved in the situation. We call this strategy Sibling Merger.

We first choose the candidate nodes that could be in this situation during the stage of the alignment of context nodes. A node is a candidate to be processed by Sibling Merger if it has sibling nodes with similar structures repeated in at least X different quantities in different training documents. Then. for each group of candidates. it determines if they are sufficiently similar to each other using the cost of alignment and the distance matrix as metric.

2.4. Results

2.4.1. Experiment Setup

The BIOIE algorithm was tested for different web sites to determine its validity, problems and strengths. We first explain how the testing was done and then we present and analyze the results.

BIOIE was built to fulfill the objective of generation of a data model for a particular web site, so in the tests we wanted to measure its ability to obtain clusters of information and the feasibility of doing it under the size and accuracy conditions that area actually demanded by the Web. In particular, we will measure the amount of discovered clusters, comparing them with those that should be found, and also showing how many of these were relevant to the user.

Clusters were classified into the following categories:

Correct: We considered a cluster as to be correct when it succeeds in encapsulating a common concept from different documents.

Locally Correct: The cluster was successfully encapsulated at a local level, but it failed to link with global clusters that represent the same information.

Incorrent: The cluster failed to encapsulate a concept, it was only data without clear correlation between them.

To show the behavior of the generated clusters when the information increases, we studied the change in the quality of clusters when the richness of the sample increased (increase the number of documents). The experiment shows the behavior of the site "www.fmylife.com" when the number of documents goes from 1 to 19. This site was chosen because it is a complex site and it could give us an idea of the behavior at different stages of evolution of the generated template.

Since performance is an important issue in the feasibility the system, we measured the algorithm running time, the average size of documents and the average percentage reduction in size when compressing the document with the most basic compression (Size Cost Compression). Finally, we also showed how the processing time was affected when we increased the number of documents to be processed. For this last test, we used the site "www.svpropiedades.cl" and a variable number of documents.

2.4.2. Data Sources Used

Public web sites of three different categories were used as sources for the tests:

Goverment and Universities: These are perhaps the sites we were most interested in since they often have mass consumption of public information but usually do not expose any web services. To represent university sites, we used pages taken from the site of Universidad Catolica de Chile; and as an example of a government site, we chose www.infoescuela.com, which is used by the Chilean gobernment to provide information about schools.

Results	Time (sec)	Documents Quantity	Documents Average size	Size after Basic Compression	Clusters Found	Clusters Expected	Correct Clusters	Locally Correct Clusters	Incorrect Clusters
www.fmlife.com	106.5	19	38094	39%	84	27	4	74	6
www.infoescuela.cl	10.9	9	8011	36%	28	24	21	7	0
cursos.puc.cl	580.0	22	36714	46%	26	26	26	0	0
www.9gag.com	33.3	9	27159	39%	170	26	5	165	0
www.svpropiedades.cl details	553.6	100	6573	42%	24	24	24	0	0
www.svpropiedades.cl search	24.7	8	15699	44%	15	14	14	1	0

TABLE 2.1. Results obtained on selected sites

Popular Sites: These are popular websites with public content. The sites 9Gag.com and fmylife.com were taken as examples of web 2.0 sites with widespread use.

Real Estate Agencies: As examples of sites that are usually crawled for commercial purposes, we used real estate web sites. Using this algorithm in such sites would facilitate the construction of web applications for search and sale of apartments and houses.

2.4.3. Results

Here we present the results we obtained from the tests. The implemented prototype was run on an Intel M520 i5 with 4GB of RAM under the platform 1.6.20 OpenJDK 64-Bit Server for Linux.

2.4.3.1. Cluster Quality

Table 2.1 shows the results we obtained by running the algorithm on the selected sites. For the sites “www.infoescuela.cl”, “cursos.puc.cl” and “www.svpropiedades.cl” we obtained nearly perfect results, and the templates were correctly captured. The work with sites ”www.fmlife.com” and “www.9gag.com” was not as succesful since the algorithm could not generalize some clusters that represented the same information. All clusters discovered, however, made sense from a local perspective and were bounded correctly. The only case where there were incorrect clusters was with “www.fmylife.com”, since this

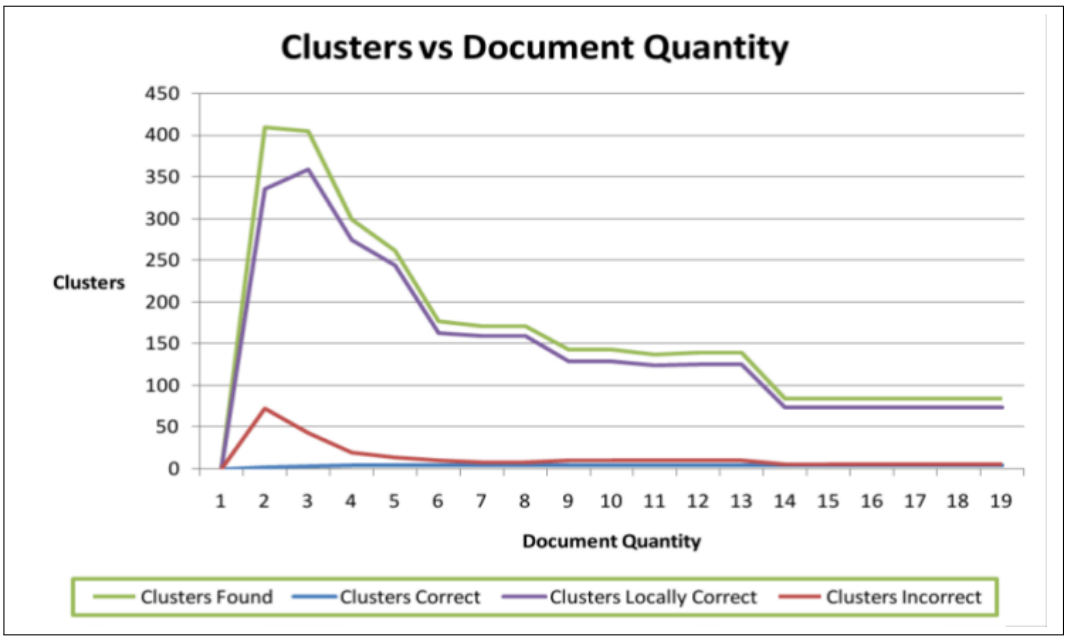

FIGURE 2.4. Graph showing the evolution of www.fmylife.com template quality as the documents were increased

site includes very subtle optional information within each story, making it almost impossible for the algorithm to not confuse the boundaries of each cluster, without introducing higher-level semantics.

2.4.3.2. Cluster Stabilization behavour

Figure 2.4 shows the evolution in the quality of the generated template for the site www.fmylife.com“ as the variability of the training information increases. Obviously, for the case of a single document, the algorithm had nothing to compare and did not find any patterns.

With two documents, we started to see some results but they were still quite poor. The number of clusters grew very fast and many incorrect patterns were found due to the lack of information. As we improved the quality of the sample, the right relationships began to appear, reducing the amount of erroneous relations, increasing the number of correct patterns and correctly linking with other local similar patterns; fewer locally-correct patterns were obtained but they were of better quality. Finally, the algorithm stabilized, achieving

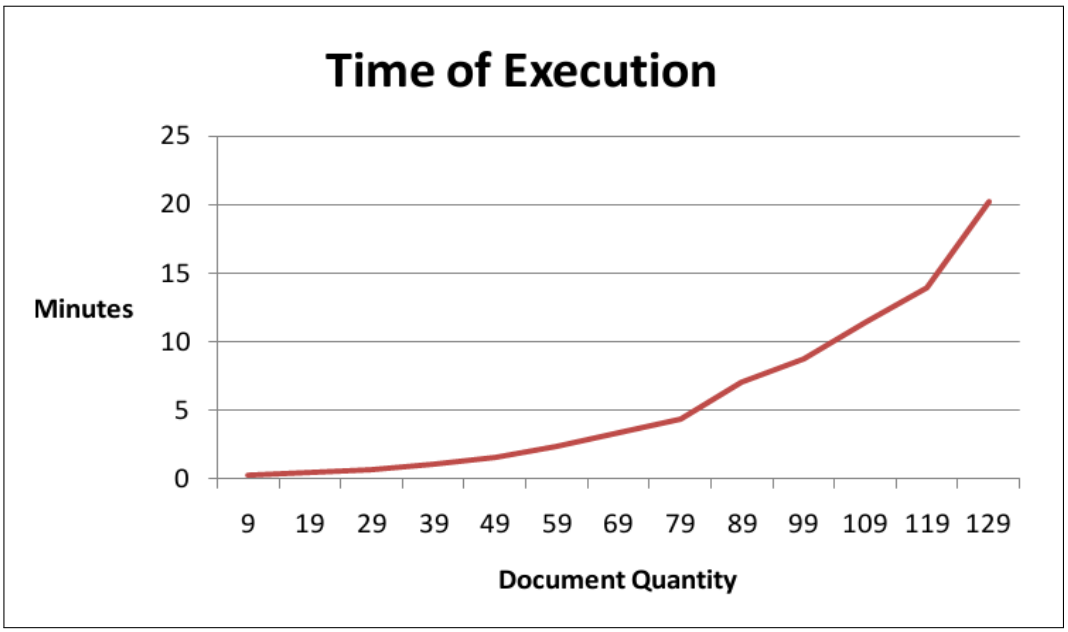

FIGURE 2.5. time required by BIOIE to analyze " www.svpropiedades.cl" details for different amounts of documents

its best solution. From this point, increasing the number of documents did not produce any improvement.

It is interesting to note that very few samples are needed to get very good results. In general, no more than 10 documents are required to obtain the optimal result. This is consistent with results reported in (Arasu & Garcia-Molina, 2003). In our experiments, we were able to obtain optimum results even with no more than 3 or 4 very well-chosen documents. The small numer of samples required by the algorithm, combined with its unsupervised nature, makes it an attractive option for mass use or even commercial use.

2.4.3.3. Time Performance Analisis

Figure 2.5 shows the variation of the time required by the algorithm as we increase the number of documents. The curve is quadratic in the range of values measured, which is consistent with the fact that the multi-alignment algorithm heuristics used are of a quadratic nature.

We see that the algorithm begins to take a lot of time for a very large number of documents, however the growth is quite slow for the ranges needed to achieve the optimum, according to that mentioned in section 2.4.3.2 which validates the algorithm as a viable alternative for practical use.

2.5. Conclusions

This paper presents an unsupervised information extraction system focused on search and encapsulation of variable concepts from a group of web documents. It works by taking advantage of html document properties and pattern detection techniques taken from bioinformatics.

The algorithm presented was tested with different types of web sites, achieving satisfactory results in terms of quality and performance comparable to existing systems of extraccion. It exibited good performance and the execution times are more than acceptable for widespread use in real cases, without the need for expensive hardware or large computing capacity.

Unlike other studies that use text alignment, our algorithm takes into account both the structure of the page markup, as well as the text contents encapsulated inside, to look for patterns and find a template of the document. This allows detection of more complex relationships without the need to deliver labeled information.

Although the algorithm uses ”domain“ information , it doesn't make semantic use of this, delegating the task to information extraction systems based on ontologies. So, beyond the clusters found and the information found within them, we believe that our work, when seen as an information modeling system, is an interesting contribution; its output is compatible with other technologies of extraction and it can be used for subsequent analysis whether semantic analisis, tagging or filtering procedures are performed. This was thought to provide a foundation for the construction of the next stages of the system to achieve a true structured representation of information on the page, in order to be able to build the desired web service and obtain a more comprehensive web.

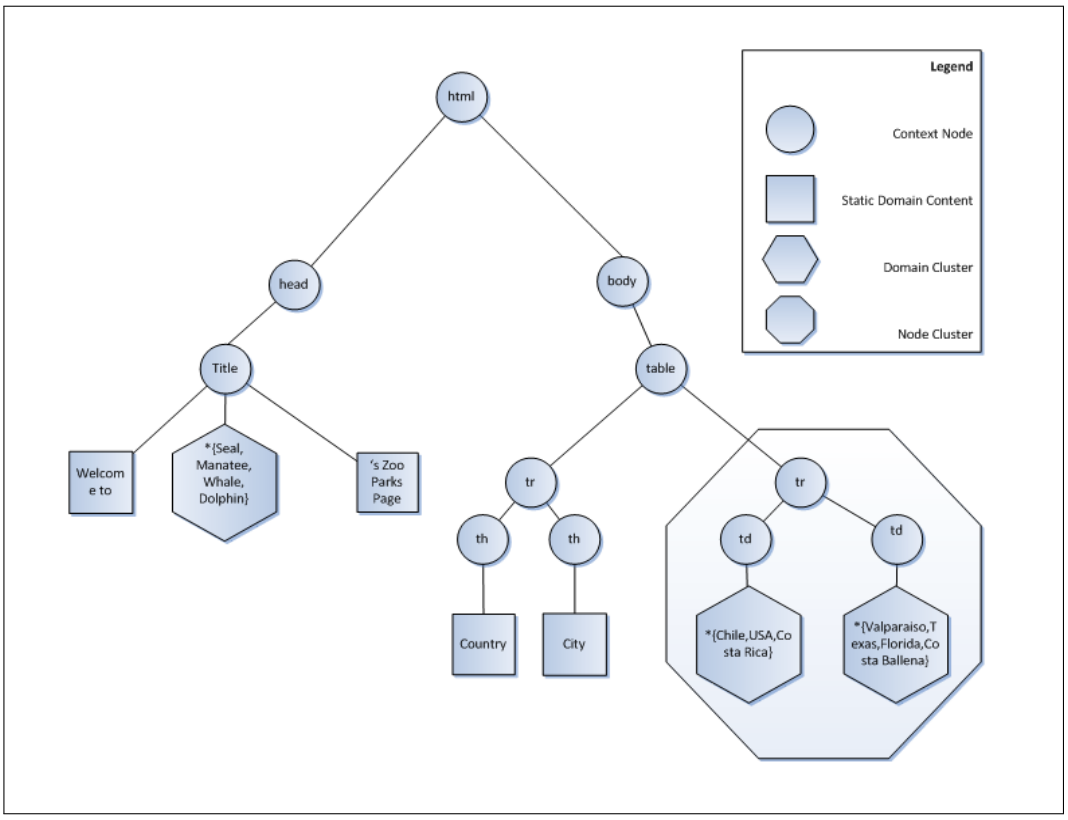

FIGURE 2.6. Figure shown a representation of the model presented by BIOIE. Circles represent static html modes, squares represent static text presented in the document, hexagons are detected clusters of variable text and octagons are clusters of variable node structures

Figure 2.6 shows the model obtained by BIOIE. We see that besides the domain of clusters obtained (which is necessary to understand the algorithm for extracting information), it also gets:

Near Static Data: Helps us understand the context of the cluster, gives us the chance to perform labeling, and tracks analysis to detect semantically similar clusters.

Node Clusters: Helps us understand relationships between different clusters and build higher-level semantic objects.

Position of Clusters in the Document: Helps us to find useful indications: for example, clusters that are related to each other or which are more likely to be important. It also tells us in which node or attribute node in the document the

information appears, giving us additional assistance to discriminate the validity of the assumptions and the importance of the cluster.

BIOIE thus creates a necessary legacy so as to continue on with our work. In the future, we plan to analyze the information given at this stage, to add the necessary semantic relationships and to get a more rich and interesting model to build interfaces. On the other hand, with the pageś data model completed, we 'll begin to apply a similar analysis for the possible operations on the model and build relationships between pages by using information encapsulated links within each page.

Finally, among the improvements we would like to do to BIOIE, we plan on considering more advanced philogenia tree constructions for more precise alignments and also a more relaxed level of separation of Domain Zones-Context Zones for less rigid results, to give better support to sites with relevant information in html format.

Chapter 3. CONCLUSIONS AND FUTURE RESEARCH

3.1. Review of the Results and General Remarks

We have presented an unsupervised information extraction system that can retrieve and encapsulate the variable concepts found in the web pages of a site, an important step towards a more ambitious goal that is automatic synthesis of a web services API for a web site. The system works by looking at the hypertext web documents as composed of two recognizable zones and then take advantage of each of them trough several aligning methods and algorithms, These methods are heavily inspired in well known structural and functional pattern detection techniques used in the field of Bioinformathics.

Unlike other systems that use text alignment, our algorithm takes into account both the structure of the page markup, as the text contents encapsulated inside to look for patterns and find a template of the document. This allows to find more complex relationships without the need to deliver labeled information.

An important challenge to materialize the initial theoretical ideas was that if we wanted those techniques imported from Bioinformatics to work for this completely different scenario, we had to improve the performance in a very significative way. In fact, at the beginning we used a more direct approach but very soon we realize that the running time was going to be un reasonable even with the help of very powerful hardware. So we had to use a combination of good heuristics, efficient data structures, data compression and data encoding to boost performance on the implemented prototype. The effects were quite impressive, since the computational time could be reduced, in some cases, from several days to a few seconds. Furthermore, it was possible now to run it in real scenarios and sites.

A very interesting feature of our proposal which makes it different from most other extraction systems is its unsupervised nature. Since it does not require the information to be labeled it works very well with the huge unlabeled unclassified information contents

available on the web. It is also worth mentioning that very few samples (less than 10) taken from the web site are needed to arrive to very good or sometime optimal results.

The system was tested both from running time and quality of the solution found for a representative sample of web sites. From the point of view of performance the tests demonstrated the technical feasibility of the approach since we were able to process real web sites in very short periods of time. From the point of view of the quality of the results, fortunately, the best results were achieved on the sites we were more interested in from the beginning (universities, government) where we got almost perfect results exceeding thus our own expectations.

The prototype seems to behave not so good with sites that include an excess of markup within the relevant domain information and in sites where there are no templates. The reason for this may reside in the rigid separation of the document between two zones and should be carefully examine if we need a truly general information extractor. For example we could have used parametric thresholds to separate domain from context information. Nevertheless, since this kind of web site was out of the scope of our research and initial motivation, it represents an interesting opportunity for further research and improvement but it does not dims in any way the value of what we achieved.

Finally, this work allowed us to realize how hard is the automatic generation of software from the data when the information is noisy and not oriented rather to human consumption that to computer or software consumption. A truly general solution to this issue requires further research. In fact important efforts are invested into transforming the information of the web so it can be more easily processed by machines.

3.2. Future Work

Although we are quite happy with BIOIE it leaves enough space for improvement. Besides a more advanced information extraction mechanism itself it needs to incorporate the search for inter-page relationships in the process of building the methods of the service.

The idea is to examine the client-server interactions and the hyperlink structure present in the documents.

Also, since our system produces as output a data source that is well suited for further semantic analysis, we plan to combine our algorithms and extraction system with ontology-based information extraction (OBIE). This way, since we start with a source that has been already analyzed from the structural point of view, we could focus on what OBIE is really good: detect and categorize semantic relationships. The output produced by our system could in fact be used for other types of analysis including similarity, labeling, etc.

Because our work rests on robust sequence alignment and bioinformatics techniques we will benefit immediately from improvements or discovering in those areas. A good example of this are recent works on GPU-based alignment or precise phylogenetics-tree construction techniques.

References

Adelberg, B. (1998). Nodose a tool for semi-automatically extracting structured and semistructured data from text documents. In *Proceedings of the 1998 acm sigmod international conference on management of data* (pp. 283–294). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/276304.276330`

Akutsu, T., Arimura, H., & Shimozono, S. (2000). On approximation algorithms for local multiple alignment. In *Proceedings of the fourth annual international conference on computational molecular biology* (pp. 1–7).

Altschul, S., Gish, W., Miller, W., Myers, E., & Lipman, D. (1990). Basic local alignment search tool. *Journal of molecular biology*, *215*(3), 403–410.

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, *25*(17), 3389.

Arasu, A., & Garcia-Molina, H. (2003). Extracting structured data from web pages. In *Proceedings of the 2003 acm sigmod international conference on management of data* (pp. 337–348). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/872757.872799`

Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval* (Vol. 463). ACM press New York.

Bailey, T., Williams, N., Misleh, C., & Li, W. (2006). MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic acids research*, *34*(Web Server issue), W369.

Ballinger, K. (2002). *.net web services: Architecture and implementation with .net*. Pearson Education.

Baumgartner, R., Flesca, S., & Gottlob, G. (2001). Visual web information extraction with lixto. In *Proceedings of the international conference on very large data bases* (pp. 119–128).

Biotechnology Information, N. C. for. (2004, March). *Just the facts: A basic introduction to the science underlying ncbi resources.* Available online. Retrieved Dec 10, 2010, from `http://www.ncbi.nlm.nih.gov/About/primer/bioinformatics.html`

Califf, M., & Mooney, R. (1999). Relational learning of pattern-match rules for information extraction. In *Proceedings of the national conference on artificial intelligence* (pp. 328–334).

Capasso, P., Cesarano, C., Picariello, A., & Sansone, L. (2006). Content-Based News Retrieval on the Web. *IJCSNS*, *6*, 5B–88.

Carrillo, H., & Lipman, D. (1988). The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics*, *48*(5), 1073–1082.

Chang, C., Kayed, M., Girgis, M., & Shaalan, K. (2006). A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 1411–1428.

Chang, C., & Kuo, S. (2005). Olera: semisupervised Web-data extraction with visual support. *Intelligent Systems, IEEE*, *19*(6), 56–64.

Chang, C.-H., & Lui, S.-C. (2001). Iepad: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on world wide web* (pp. 681–688). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/371920.372182`

Chinnici, R., Moreau, J.-J., Ryman, A., & Weerawarana, S. (Eds.). (2007, June). *Web services description language (wsdl) version 2.0 part 1: Core language* (Tech. Rep.). Available from `http://www.w3.org/TR/2007/REC-wsdl20-20070626/`

Chuang, S.-L., & Hsu, J. Y.-j. (2004). Tree-structured template generation for web pages. In *Proceedings of the 2004 ieee/wic/acm international conference on web intelligence* (pp. 327–333). Washington, DC, USA: IEEE Computer Society. Available from http://dx.doi.org/10.1109/WI.2004.143

Cowie, J., & Lehnert, W. (1996, January). Information extraction. *Commun. ACM*, *39*, 80–91. Available from http://doi.acm.org/10.1145/234173.234209

Crescenzi, V., Mecca, G., Merialdo, P., et al. (2001). Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the international conference on very large data bases* (pp. 109–118).

Crochemore, M., Landau, G., & Ziv-Ukelson, M. (2002). A sub-quadratic sequence alignment algorithm for unrestricted cost matrices. In *Proceedings of the thirteenth annual acm-siam symposium on discrete algorithms* (pp. 679–688).

Da Silva, A. S., Barbosa, D., Cavalcanti, J. a. M. B., & Sevalho, M. A. S. (2007). Labeling data extracted from the web. In *Proceedings of the 2007 otm confederated international conference on on the move to meaningful internet systems: Coopis, doa, odbase, gada, and is - volume part i* (pp. 1099–1116). Berlin, Heidelberg: Springer-Verlag. Available from http://portal.acm.org/citation.cfm?id=1784607.1784701

Dayhoff, M., & Schwartz, R. (1978). A model of evolutionary change in proteins. In *In atlas of protein sequence and structure.*

Eddy, S. (1995). Multiple alignment using hidden Markov models. In *Proceedings of the third international conference on intelligent systems for molecular biology* (Vol. 3, pp. 114–120).

Edgar, R. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, *32*(5), 1792.

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. Unpublished doctoral dissertation, University of California, Irvine,

Irvine, California. Available from http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Fiumara, G. (2007). Automated Information Extraction from Web Sources: a Survey. *BOF*, 1.

Florescu, D., & Kossmann, D. (2002). Xl: An xml programming language for web service specification and composition. In (pp. 65–76).

Freitag, D. (2000). Machine learning for information extraction in informal domains. *Machine learning*, *39*(2), 169–202.

Goth, G. (2004). Critics say web services need a rest. *IEEE Distributed Systems Online*, *5*.

Group, O. M. (1999, Oct.). *The common object request broker: Architecture and specification (corba 2.3.1 specification)* (Tech. Rep.). Author. Available from http://cgi.omg.org/cgi-bin/doc?formal/99-10-07

Gudgin, M., et al. (Eds.). (2002, April). *Soap version 1.2 part 1: Messaging framework* (Tech. Rep.). Available from http://www.w3.org/TR/soap12-part1/

Henikoff, S., & Henikoff, J. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, *89*(22), 10915.

Huelsenbeck, J., & Ronquist, F. (2001). MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, *17*(8), 754–755.

Konagurthu, A., & Stuckey, P. (2006). Optimal sum-of-pairs multiple sequence alignment using incremental Carrillo and Lipman bounds. *Journal of Computational Biology*, *13*(3), 668–685.

Kushmerick, N. (2000). Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, *118*(1-2), 15–68.

Laender, A., Ribeiro-Neto, B., Silva, A. da, & Teixeira, J. (2002). A brief survey of web data extraction tools. *ACM Sigmod Record*, *31*(2), 84–93.

Laender, A. H. F., Ribeiro-Neto, B., & Silva, A. S. da. (2002). Debye - data extraction by example. *Data & Knowledge Engineering*, *40*(2), 121–154. Available from http://www.sciencedirect.com/science/article/B6TYX-44JJ8WT-1/2/34510b76d91ca36092e81e607181f3de

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, pp. 707–710).

Li, Y., & Bontcheva, K. (2007). Hierarchical, perceptron-like learning for ontology-based information extraction. In *Proceedings of the 16th international conference on world wide web* (pp. 777–786). New York, NY, USA: ACM. Available from http://doi.acm.org/10.1145/1242572.1242677

Liu, L., Pu, C., & Han, W. (2002). XWRAP: An XML-enabled wrapper construction system for web information sources. In *Data engineering, 2000. proceedings. 16th international conference on* (pp. 611–621).

Luscombe, N., Greenbaum, D., & Gerstein, M. (2001). What is Bioinformatics? A proposed definition and overview of the field. *Yearbook of Medical Informatics*, 83–100.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453. Available from http://www.sciencedirect.com/science/article/B6WK7-4DN8W3K-7X/2/0d99b8007b44cca2d08a031a445276e1

Noah, S. A., Zakaria, L., & Alhadi, A. C. (2009). Extracting and modeling the semantic information content of web documents to support semantic document retrieval. In *Proceedings of the sixth asia-pacific conference on conceptual modeling - volume 96* (pp. 79–86). Darlinghurst, Australia, Australia: Australian Computer Society, Inc. Available from http://portal.acm.org/citation.cfm?id=1862739.1862751

Notredame, C., & Higgins, D. (1996). SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Research*, *24*(8), 1515.

Notredame, C., Higgins, D., & Heringa, J. (2000). T-coffee: a novel method for fast and accurate multiple sequence alignment1. *Journal of molecular biology*, *302*(1), 205–217.

Parker, C., Fern, A., & Tadepalli, P. (2006). Gradient boosting for sequence alignment. In *Proceedings of the 21st national conference on artificial intelligence - volume 1* (pp. 452–457). AAAI Press. Available from `http://portal.acm.org/citation.cfm?id=1597538.1597611`

Pautasso, C., Wilde, E., & Marinos, A. (2010). First international workshop on restful design (ws-rest 2010). In *Proceedings of the first international workshop on restful design* (pp. 1–3). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/1798354.1798375`

Pautasso, C., Zimmermann, O., & Leymann, F. (2008). Restful web services vs. "big"' web services: making the right architectural decision. In *Proceeding of the 17th international conference on world wide web* (pp. 805–814). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/1367497.1367606`

Raghavan, S., & Garcia-Molina, H. (2001). Crawling the hidden web. In *Proceedings of the international conference on very large data bases* (pp. 129–138).

Sahuguet, A., & Azavant, F. (2001). Building intelligent web applications using lightweight wrappers. *Data & Knowledge Engineering*, *36*(3), 283–316.

Saitou, N., & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, *4*(4), 406.

Schlimmer, J. C. (Ed.). (2002, October). *W3c web services description requirements.* Available from `http://www.w3.org/TR/ws-desc-reqs/`

Sessions, R. (2004, December). Fuzzy boundaries: Objects, components, and web services. *Queue*, *2*, 40–47. Available from http://doi.acm.org/10.1145/1039511.1039533

Shadbolt, N., Hall, W., & Berners-Lee, T. (2006, January). The semantic web revisited. *Intelligent Systems, IEEE*, *21*(3), 96 -101.

Sleeper, B., & Robins, B. (2001, June). *Defining web services.* Available from http://www.perfectxml.com/Xanalysis/TSG/TSG_DefiningWebServices.pdf

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, *147*(1), 195–197. Available from http://www.sciencedirect.com/science/article/B6WK7-4DN3Y5S-24/2/b00036bf942b543981e4b5b7943b3f9a

Su, W., Wang, J., & Lochovsky, F. H. (2009, July). Ode: Ontology-assisted data extraction. *ACM Trans. Database Syst.*, *34*, 12:1–12:35. Available from http://doi.acm.org/10.1145/1538909.1538914

Tekaia, F. (2003, December). *Bioinformatics faq.* Available online. Retrieved Dec 10, 2010, from http://www.bioplanet.com/bioinformatics_faq.html

Thompson, J., Higgins, D., & Gibson, T. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, *22*(22), 4673.

Wang, J., & Lochovsky, F. H. (2003). Data extraction and label assignment for web databases. In *Proceedings of the 12th international conference on world wide web* (pp. 187–196). New York, NY, USA: ACM. Available from http://doi.acm.org/10.1145/775152.775179

Wheeler, T. (2009). Large-scale neighbor-joining with ninja. *Algorithms in Bioinformatics*, 375–389.

Wimalasuriya, D. C., & Dou, D. (2009). Using multiple ontologies in information extraction. In *Proceeding of the 18th acm conference on information and knowledge management* (pp. 235–244). New York, NY, USA: ACM. Available from http://doi.acm.org/10.1145/1645953.1645985

Wimalasuriya, D. C., & Dou, D. (2010, June). Ontology-based information extraction: An introduction and a survey of current approaches. *J. Inf. Sci.*, *36*, 306–323. Available from http://dx.doi.org/10.1177/0165551509360123

Wong, T.-L., & Lam, W. (2007, February). Adapting web information extraction knowledge via mining site-invariant and site-dependent features. *ACM Trans. Internet Technol.*, *7*. Available from http://doi.acm.org/10.1145/1189740.1189746

Zheng, S., Song, R., Wen, J.-R., & Wu, D. (2007). Joint optimization of wrapper generation and template detection. In *Proceedings of the 13th acm sigkdd international conference on knowledge discovery and data mining* (pp. 894–902). New York, NY, USA: ACM. Available from http://doi.acm.org/10.1145/1281192.1281287