Embarking on a Web Information Extraction Project

Daniela Xhemali
Computer Science
Loughborough University
Loughborough, LE11 3TU
UK
D.Xhemali@lboro.ac.uk

Chris J. Hinde
Computer Science
Loughborough University
Loughborough, LE11 3TU
UK
C.J.Hinde@lboro.ac.uk

Roger G. Stone
Computer Science
Loughborough University
Loughborough, LE11 3TU
UK
R.G.Stone@lboro.ac.uk

Abstract

Web Information Extraction (WIE) is a very popular topic, however we have yet to find a fully operational implementation of WIE, especially in the training courses domain. This paper explores the variety of technologies that can be used for this kind of project and introduces some of the issues that we have experienced. Our aim is to show a different view of WIE, as a reference model for future projects.

1 Introduction

Web Information Extraction (WIE) is much in demand. Its popularity comes not only from the need for continuous knowledge growth, but also from the needs of industry for quick, efficient solutions to information gathering.

Many ideas have emerged over the years, thus there are various WIE technologies that can be used to a degree. However, some people who know of such technologies, do not know what problems to solve with them. Equally, there are people who know what problems they are trying to solve, as well as a range of possible solutions, but do not know how to choose the right methodology. Some researchers decide to go for the more sophisticated techniques without even considering the easier solutions first.

This paper is a guide for researchers embarking on similar projects; it summarizes not only possible solutions to WIE but also raises relevant issues. It also asks important questions to new researchers giving them a more complete understanding of what they hope to achieve. To our knowledge no other paper has confronted the subject in this way.

The paper is as follows: Section 2 introduces the organization involved in this project. Section 3 discusses the various existing approaches to WIE. Section 4 discusses the issues and a classifies possible solutions. Section 5 concludes.

2 Course Information Extractor (CIE)

Apricot Training Management (ATM) is an independent brokerage assisting organizations to find appropriate training. ATM currently uses its Customer Relationship Management software package as the front-end to all necessary details about their clients including the clients' needs and behaviours as well as the various courses available. The latter is where the problems exist. Currently, a full-time employee is responsible for ordering the latest prospectuses from different training providers, cataloguing, shelving and manually entering the information found into the database. This is a time consuming, labour-intensive process, which does not guarantee an up-to-date database, due to the limited life expectancy of some course information such as dates and prices and other limitations in the accessibility of up-to-date, accurate information. Automating the extraction of information from training websites would make the process less labour intensive, however the number of possible sites is unlimited, thus the CIE will need to filter out all relevant training websites before attempting to extract specific course details. Courses can be based anywhere in the UK and can cover any topic.

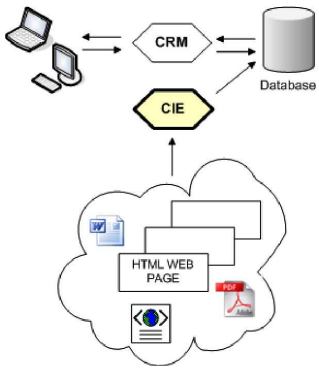

Figure 1: Course Information Extractor.

Our aim is to automate the process of finding, extracting and storing course information in the database without much or any user involvement, in order to relieve ATM's employees from needing to check every piece of information before using it. The CIE will have to work unnoticed in the background keeping the database up-to-date with the latest course information. Figure1 shows the role of CIE at ATM.

3 WIE approaches

WIE has been described as "An attempt to convert information from different text documents into database entries" [5], which is exactly what we are trying to achieve. This is very attractive to many individuals and organisations because, with the World Wide Web (WWW) currently being the biggest online public source of information and still growing at a fast pace, organisations will want to be able to manipulate and update this information and use the Web as a resource for data discovery.

Many researchers have dedicated their time to the investigation and improvement of WIE techniques, thus there exist various approaches to handling WIE, ranging from manual techniques to semi-automated to fully automated approaches (described below).

The traditional approach of dealing with WIE is through specialized programs called Wrappers. Wrappers convert Web data into more structured data to aid the extraction process. Early approaches to creating wrappers were based on manual techniques [8, 9] where individuals examined web pages, manually finding the areas of interest and then creating patterns for extracting them. Wrappers generated this way are labour intensive, difficult to create and maintain. Thus, the objective has been to automate the process of generating wrappers, facilitating the extraction of information from the web. WIE techniques can be grouped in the following five categories:

1. HTML Wrapper Generation [1, 3, 19, 24, 27].
2. XML Wrapper Generation [11, 13, 22].
3. Machine Learning [26, 25].
4. Extraction Ontologies [6, 7, 23].
5. Natural Language Processing [12, 15, 20, 28].

3.1 HTML Wrapper Generation.

WIE techniques based on structure analysis of HTML (Hyper-Text Markup Language) pages, rely on finding patterns in the structure of the source code of each page such as: HTML tags, font differences, layout etc.

The normal technique used in these approaches is the analysis of one web page at a time. A system using this technique is Webfoot [24], which divides the page into sentence-length segments based on page layout.

Yang and Zhang [27] also base their approach on analysing the content of one HTML page at a time, however, unlike Webfoot, they do not believe HTML tags are stable enough to be considered in the pattern finding process. Instead, they separate areas of interest based on apparent visual boundaries, such as headings, subheadings etc. Each web page is represented as a tree-structure of all patterns found on the pages. These patterns are then compared to each other to generate more generalized wrappers. Both approaches rely heavily on the HTML source code being consistent throughout, however, the Web is far from regular. There are many inconsistencies and irregularities, which need to be considered.

A different technique is the analysis of two HTML web pages. The source code of both pages is compared and the similarities and differences found used to generate a wrapper that would apply to as many pages as possible. RoadRunner [3] is a good example of this, which automates the Wrapper Generation process with no prior knowledge of the target pages, whilst managing to deal with nested HTML structures . RoadRunner is limited to union-free websites and it is also heavily dependent on HTML tags and structures being correctly applied.

Reis et al.[19] is similar to the above, although unlike RoadRunner, it is based on tree-edit distance i.e. the minimal cost required to transform one data tree to another. This research only concentrates on extracting large portions of news data, whereas our project will go into more detail and extract specific information related to courses, thus it may have to be more 'aggressive' in order to pinpoint the rules.

3.2 XML Wrapper Generation.

XML (eXtensible Markup Language) Technology has become an important part of WIE projects, due to the irregularities of the Web, in particular the inconsistency of HTML.

The main point of using XML in WIE is to convert the HTML code of each web page to a more structured format, then apply the information extraction techniques to the XML document. By converting the HTML document to XML, the data is separated from its layout. This is because, unlike HTML which is designed to focus on how the data looks on a page, XML is more concerned with describing what the data is. This is achieved by using a Document Type Definition (DTD) or an XML Schema that describes the data. XML tags are also not predefined, programmers must define their own tags, which gives more control over the description process. An XML document however, would be of no use by itself, as its tags do not mean anything to computers. XML documents have to be further manipulated by external packages, such as WIE systems.

Liu [11] develops mechanisms that provide a clean separation of the semantic meaning of information on each web page from the process of wrapper generation.

Myllymaki [13] also converts each HTML web page to XML as the first step of the extraction process, however, in this case the HTML code is first translated to XHTML (eXtensible HTML), which 'repairs' the code from any abnormalities such as missing tags, etc. There are various tools that achieve this such as the Tidy package [17]. The XHTML code is then converted to XML. The XSLT language is used to achieve this by using XPath to find parts of the XHTML document that match a template, then transforming the matching part into XML.

Despite the success of this research in relying less on HTML and more on the content itself, it is still too rigid in the way it finds items of interest on the target web pages. E.g. they will look into the cell of a table and extract the information that is in bold. Needless to say, this technique will not work if the item in bold is changed to e.g. italic. Nevertheless, this is an improvement on techniques such as the one used by W4F [22], which uses absolute paths to the location of interest; the path to the third column of the second row of the first table would be: HTML.BODY.TABLE[1].TR[2].TD[3]. W4F cannot deal with layout changes.

3.3 Machine Learning (ML).

ML techniques are popular because of their ability to make intelligent guesses when extracting information from the Web and eliminate much of the user involvement.

Besides Information Extraction and Data Mining, ML has an extensive spectrum of applications including: Medical diagnosis, Credit card fraud detection, Speech recognition etc. Techniques used can be categorised based on how much available feedback is given to the learning process [21]. These categories include:

Supervised Learning - The system learns relationships between input(s) and output(s).

Reinforcement Learning - The system learns not to make the same mistake twice.

Unsupervised Learning - The system learns relationships between inputs alone.

The above three categories of learning techniques learn rules from user-defined training data and then use this newly acquired knowledge to extract new information. Tools that adopt this approach include: [2, 4, 26].

Crystal [26] uses a ML algorithm, however it requires a semantic hierarchy of the data, as well as the training data to be manually annotated by an expert. Crystal learns and creates extraction rules by generating multi-slot concept frames. This allows for related information to be extracted together.

The multi-slot concept is essential to our research as well, because a web page may list many courses with associated titles, prices, locations etc. however, unless all this information is extracted as a set, the result would not be useful.

Omini [2] is a system that uses five different heuristics to fully automate the object extraction process from the Web. Omini works with static and dynamic websites achieving very high precision and recall values, 100% and 92-98% respectively, as it not only uses each heuristic individually, but also combines them. Furthermore, each heuristic is assigned some confidence (probability) estimated from the training set to increase efficiency. Omini's limitation however, is that it is only successful at recognising and retrieving single groups of records that are of interest on the page; it does not handle multiple areas of interest well. Another weakness is that Omini assumes that the records to be extracted are always under the largest HTML tree. Therefore the system fails for some websites.

As appealing as ML techniques may be, example annotations for ML are expensive and time consuming processes, hence ML techniques may not be the answer to every WIE project.

3.4 Extraction Ontologies

HTML-Based and ML approaches try to find answers to questions such as: "How to discover patterns", "How to create rules", "How to train a system so it learns where to look" etc. However, there are other questions that can be asked about a WIE system, which the above do not consider. One such question is: "How to restructure the content of a web page, so the new structure can be easily extracted". One answer to this question involves using Ontologies to restructure the web pages into standard models that are independent of the original information sources. The idea behind this is that if we can identify how the data is organised on a web page then it is much simpler to extract. [23].

One main benefit of using Ontologies is that they can be used to reason about relations. However, ontologies are complex and there are various aspects to be taken into account such as:

3.4.1 Reusability

Reusability is difficult to achieve yet very important because the more reusable an Ontology is the less dependent it will be from a specific domain, therefore it can be generalised to apply to a much larger range of data. These types of Ontologies are also known as "Upper Ontologies" as they try to describe very general concepts that are global to most domains. An alternative would be to try and merge various ontologies together, however, this is an error-prone process, time-consuming and expensive.

3.4.2 Ontology Maintenance

Change is inevitable, as requirements grow, knowledge of a domain evolves, errors may emerge, thus Ontologies have to be maintained and updated regularly. This is particularly challenging for large-scale Ontologies, where the information to be analysed is more complex.

The focus of the research from Snoussi [23] is websites which change the content frequently but not the overall structure e.g. stock exchange quotes. The system works by first converting the page content into XML, then using Ontologies to model the data, assigning it semantics and finally carrying out the extraction of the data.

The difficulty with this approach is that the description showing how the data is found on a page is created manually, nevertheless, this shows that if we identify how the data is organised on a web page then it is simpler to extract.

Embley [6] concentrates on the car advertisements domain, however, his main goal is to show that Extraction Ontologies can be used to aid semantic understanding and the Semantic Web. Unlike [23], this research uses ML rules over the chosen heuristics, to determine whether a web page is applicable for a given Ontology.

The research achieves over 90% for both recall and precision ratios and it is also successful in retargeting the 'car ads' application to other domains such as: mobile phones, restaurants, games etc. However, the Ontology needed for each different domain requires a few dozen person-hours to be updated to fit the description of the new domain. This can quickly turn into a very expensive process.

3.5 Natural Language Processing (NLP)

Information on web pages is displayed in many different formats ranging from structured tables to completely unstructured text. Creating rules to extract information from structured sources is easier than free text because one can be successful at creating extraction rules without making the system too inflexible to potential changes, however with free text, the layout of the data is no longer helpful, and understanding the meanings of the items of interest is more important.

Computers are not capable of 'understanding' the data they work with; they need to be 'told' how to get to it. NLP techniques try to help computers recognise language structures as an individual would. This is not as straight forward as it sounds, because there are many exceptions to the rules that define a language as well as other irregularities summarised below:

1. Pronouns are used to replace nouns. E.g. "The course was good. It helped everyone". How would the computer know that 'it' represents the 'course'?
2. Some words are spelt the same but mean different things, e.g. river bank vs. financial bank.
3. Synonyms are also common. WIE systems needs to realize that all synonyms describe the same situation.
4. English has exceptions. Most verbs produce their past tense by adding 'ed'. But eat → ate, etc.
5. We use many vague terms, which are imprecisely defined. WIE systems need to reason about them.

The above give only a taste of the problems facing NLP techniques. Our research will need to deal with extraction from free text especially when dealing with course descriptions.

Many research works have concentrated in this field. These works usually incorporate techniques such as: filtering, lexical analysis of words and phrases to separate the free text into tokens of text, part-of-speech tagging or otherwise known as grammatical tagging which uses algorithms to tag the words in free text as parts of speech e.g. nouns, adjectives, verbs etc.

Riloff and Jones [20] researched the idea of automating the construction of a domain-specific dictionary, using as input only a set of un-annotated training pieces of text and some 'seed' words from the interest domain. The heart of their approach is a technique called 'mutual bootstrapping' which learns extraction patterns from the 'seed' words, then uses these patterns to extract more words from which to continue learning. This approach is successful in generating a dictionary of extraction patterns in parallel with a semantic lexicon of the interest domain. However this may end up being too general, as many domains use similar terms e.g. this research found that the 'vehicle' dictionary created for texts related to terrorism was very similar to the 'weapons' dictionary.

More recent uses of NLP techniques include: classifying people's opinions over subjects from various web pages [15], summarising the content of specific websites automatically [28], analysing music lyrics [12] by concentrating mainly on structure detection and text categorisations etc. These however, treat text as just a grammatical part of language and do not worry about really 'understanding' its meaning.

The above gave a brief description of the various methods that can be used to deal with WIE. Each approach has its pros and cons, however the question still remaining is: "How does one choose the right approach for a project?"

The following section discusses our attempt to test some of the simpler solutions to WIE and the issues encountered on the way.

4 Issues and Irregularities

With the rapid pace of progress in science and technology, it is inevitable that WIE methods will also continue to develop and result in increasingly better outcomes. However, does this mean that we should go for the most sophisticated WIE methods straight away or do we consider the simpler techniques first? How far can we get without using any intelligence at all?

Our research is based on the principle that there is no need to use high-tech technologies unless the simpler techniques prove insufficient in achieving the project's goals. Thus, in an attempt to answer the above questions, we decided to make a start on our project using the most familiar WIE technique i.e. Wrapper generation (see Sections 3.1 and 3.2).

Several training course websites were selected by ATM. So far, ten HTML-based websites have been analyzed, each including multiple web pages. Wrappers have been created for five of them, using Regular Expressions and PHP working with the assumption that all web pages of the same website share very similar if not the same layout and format of the data.

Our approach is based on a web page being chosen at random from a predefined website. The pattern recognition process is performed on this page. The resulting wrapper is then applied to all other pages, enhancing the wrapper if necessary so it applies to all pages of that website.

The wrappers created for the five websites share many common functions, particularly in finding anchors within each page that serve as starting points for the extraction process, however there are also differences due to the many ways the same concept is expressed in different websites. This shows that the CIE will need to construct a 'generalized' wrapper using rules from each of the existing individual wrappers. Thus a knowledge base will be needed to determine which part of the wrapper to use, and fuzzy logic will be necessary when exact matching is not applicable and the closest match will need to be found instead.

The CIE will also need to find all web pages containing course information from each website visited. We are currently working towards creating a spider program, which will find all possible web pages within a given website, taking into account the possibility of broken links, the variety of file name extensions available such as: .cfm, .asp, .pl etc., the different formats of image and relative links, whilst rejecting frame links and web pages that are not direct children of the root page. The relevance of each page will then be checked by applying a series of steps including: applying 'human discovered' indicators to look for course keywords on each page; applying 'program discovered' indicators where an Ontology will be used to aid the categorization of the domain etc. Once the page is determined as relevant, the generalized wrapper will be applied to extract the information.

The following lists the issues encountered so far, which make it difficult for Wrappers to perform well in isolation. Also, we try to show possible solutions to these problems and what has already been done that may be of help.

4.1 Data Formats

Web data exists in many different formats ranging from structured to completely unstructured data. A lot has been achieved with the extraction of structured data because they obey the same format/layout, they keep to the same order within each area they appear and they do not contain any missing information, thus extraction rules for this kind of data do not present many issues. Some work has also been done on semi-structured data, but there is still a lot of room for improvement, however the most challenging area at present is related to unstructured data, because they can be of any nature; they follow no format/layout rules and they can be very unpredictable, thus generalising rules for this type of data can be very error-prone. Some of the methods currently used to deal with unstructured data are Ontologies, as they help in introducing structure by establishing relations amongst different concepts in the document, and NLP which attempts to understand the text based on the rules that apply to the natural language used by humans.

4.2 Areas of Interest

Our research centres on extracting course information from web pages. However, some training providers display the content of their courses in .pdf, .doc or .xls formats. Some work has been done towards extracting data from .pdf documents where the format of the document is analysed and converted to XML before extraction; however, no results have been reported from .doc or .xls documents.

4.3 Tabular Information

Research has been carried out in extracting information from HTML tables and some good results have been achieved. However, tables are not always as structured as they should be; often rows or columns are merged into single cells. Some tables use the first row to show the headers of the data, some use the first column and some do not have headers at all.

During the analysis of the ten course websites, it was discovered that some tables used images as data headers. This would require Image Recognition tools to identify the objects in the images and extract features that make up the images such as lines, regions and possibly areas with specific textures.

It has also been observed that some cells contain ambiguous information e.g. it is not uncommon for a website to display N/A in a cell or even leave the cell empty.

Some tables run over many pages and users need to click somewhere to get to the following set of results. This data is not displayed in the main page's source code, thus it presents a challenge for the extraction process.

Furthermore, tables are used not only to display results but also to give structure to other information on a page such as links, ads etc. These should clearly not be extracted. Some research works deal with this issue by only considering for extraction tables with more than 1 row and 1 column. This assumption is not restrictive enough, thus many tables would get extracted unnecessarily giving inaccurate results.

4.4 Logins/Registrations

Some websites require users to log in before allowing access to their data. Logging in assumes that a user has already registered on that website. It is reasonable for the WIE system to request the users' input, however, the system should 'remember' the details entered and use them again in the future, if need be. If the system encounters a problem or it is uncertain about the credibility of a particular source, the users' input may again be required to confirm or refute the record. The system should then 'learn' from this input, so it can reason for itself and eliminate users' involvement in the future. ML techniques would be suitable in such cases.

4.5 The "Deep Web"

Also known as the Hidden Web, WIE from the Deep Web is a popular topic amongst researchers [14, 18]. One of the reasons behind this popularity is that incredible results have been achieved at extracting information from public sites, however, there is a large number of websites still 'hidden' from the crawler-based Search Engines, which cannot create their indices unless they can 'see' the information on the pages they are crawling. The weakness of Search Engines with the Deep Web data means that over 85% of users who regularly take advantage of Search Engines to locate information [10] will

never come in contact with this type of data.

Unlike basic HTML sites, where information is statically placed on the page, 'hidden' websites store their information in databases and provide HTML-based search forms to facilitate users' interaction with the database content. These websites are referred to as Search Sites. The general procedure is for users to submit their request as keywords, then the website queries the database to find any possible matches and returns them back to the user. Many course websites have embraced this structure, due to the vast number of courses they advertise, hence they require some prior information from the user on what they are searching for e.g. course title, location etc. One of the problems with such websites is that the URL of the results page is dynamic, thus the WIE system is not able to crawl to this page the conventional way. Our observation however, has shown that if one opens a query result in a new window (by right clicking on the link and choosing 'Open in New Window') the address bar shows the complete URL to the results page. This has given us the chance to study the URL and find patterns in its structure.

4.6 Vague Notation

Today's Web has been created for human browsing. Unlike the Semantic Web where information is given well-defined meaning, machines cannot understand the meaning of the data in the current web. The knowledge presented in some websites may also be ill-structured, uncertain or vague, thus we need methods that go beyond the two-value-based logical methods to be successful in extracting this kind of data.

Some of the issues with our language were described in Section 3.5. One of these issues was Synonyms. Our observation thus far has shown that there are many similarities amongst websites within the training course domain, despite them being expressed in different ways. This is difficult to capture using Wrappers, particularly for large domains. Some of the methods currently employed to deal with these cases involve using Transitive Similarities and Inheritance Distance to recognise various phrases, however they do not achieve recognition of all the similarities existent, due to the complexity of our language. A better way of dealing with this would be creating Ontologies, as they are controlled vocabularies that formally describe objects and relations. Later developments have taken this even further and introduced Fuzzy Ontologies to allow the representation of different viewpoints within a single framework [16].

The above discussed some of the issues encountered as well as some of the existing solutions. Basic wrappers would struggle to achieve good results in these situations due to their limitations in generalizing rules for all possible data formats and their tendency to fail when web pages change their layout or content. Thus, more intelligent solutions would be required.

5 Conclusions

In this paper we have discussed our recent effort in building an automatic Web Information Extraction system for the training courses domain. A variety of existing technologies have been discussed and a number of issues encountered have been exposed together with a range of possible solutions. One of the questions raised in this paper was "How does one choose the right approach for their project".

Potential solutions can be classified using criteria such as price, functionality, time scale, amount of programming involved etc. however, the most important factor is the type of project itself e.g. if information is to be extracted from a small number of websites, basic wrappers would be the cheapest, quickest, hence most sensible choice to go for, however, if the number of websites is unlimited, then there is need for a system robust enough to work with a large range of websites and deal with potential changes in their content and layout. Thus, more intelligent approaches need to be considered.

One can also decide to choose 'off-the-shelf' vs. 'write-your-own' code, as an easier and cheaper approach to programming, however, this limits the control over the system, particularly if errors occur or a certain functionality needs to be changed or enhanced. The time then required in finding and fixing problems may be as long as writing the entire code from scratch.

For the successful completion of our project, we have chosen the 'write-your-own' code method and we believe that we may need a combination of approaches, including Ontologies, Machine Learning and potentially Image Recognition as well as Fuzzy Browsers.

6 Acknowledgements

We would like to thank the whole team at ATM for the support and help they have offered to us since the start of the project.

References

[1] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. In *Workshop on Management of Semi-structured Data*. 1997.

[2] D. Buttler, L. Liu, and C. Pu. A fully automated object extraction system for the world wide web. In *International Conference on Distributed Computing Systems*, volume 21, pages 361–370, 2001.

[3] V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the 27th VLDB Conference*, Universita di Roma Tre., 2001.

[4] F. Denis, R. Gilleron, and F. Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348:70–83, 2005.

[5] L. Eikvil. Information extraction from world wide web - a survey. Technical Report 945, Norwegian Computing Centre. Norway, 1999.

[6] D. Embley. Toward semantic understanding: An approach based on information extraction ontologies. In *Proceedings of the 15th Australasian database conference*, pages 3–12. Australian Computer Society Inc., 2004.

[7] D. Embley. Toward tomorrow's semantic web: An approach based on information extraction ontologies. Utah, USA, 2005. Position Paper for Dagstuhl Seminar.

[8] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semi-structured information from the web. In *Proceedings of the Workshop on Management of Semi-structured Data*, 1997.

[9] G. Huck, F. P., K. Abere, and E. Neuhold. Jedi: Extracting and synthesizing information from the web. In *CoopIS*, 1998.

[10] S. Lawrence and C. Giles. Accessibility of information on the web. *Nature*, 400:107–109, 1999.

[11] L. Liu, W. Han, D. Buttler, C. Pu, and W. Tang. An xml-based wrapper generator for web information extraction. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 540–543, 1999.

[12] J. Mahedero, A. Martinez, and P. Cano. Natural language processing of lyrics. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 475–478, Singapore, 2005.

[13] J. Myllymaki. Effective web data extraction with standard xml technologies. *Computer Networks*, 39(5):635–644, 2002.

[14] M. Nakatoh, T.and Sakai, Y. Koga, and S. Hirokawa. Generation of query url for search sites. In *Proceedings of SSGRR*, pages 616–627, 2002.

[15] T. Nasukawa and J. Yi. Sentiment analysis: Capturing favourability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, pages 70–77, 2003.

[16] D. Parry. *Fuzzy Ontologies for Information Retrieval on the WWW*. Elie Sanchez, France, 2006.

[17] D. Raggett. Html tidy library project. http://tidy.sourceforge.net/, 2007. Last visited May 2007.

[18] S. Raghavan and H. Garcia-Molina. Crawling the hidden web. In *Proceedings of International Conference on Very Large Databases*, volume 27, pages 129–138. EDIT, 2001.

[19] D. Reis, P. Golgher, A. Silva, and A. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International Conference on World Wide Web*, pages 502–511, 2004.

[20] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence*, pages 474–479, 1999.

[21] S. Russell and P. Norvig. *Artificial Intelligence: A modern approach*. Prentice Hall, London, UK, 2 edition, 2003.

[22] A. Sahuguet and F. Azavant. Looking at the web through xml glasses. In *Proceedings of the Fourth IECIS International Conference on Cooperative Information Systems*, page 148, 1999.

[23] H. Snoussi, L. Magnin, and J. Nie. Heterogeneous web data extraction using ontology. In *The AI-2002 Workshop on Business Agents and the Semantic Web*, Calgary, 2002.

[24] S. Soderland. Learning to extract text-based information from the world wide web. In *Proceedings of Third International Conference on Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence (AAAI), 1997.

[25] S. Soderland. Learning information extraction rules for semi-structured and free text. *The Journal of Machine Learning*, 1998.

[26] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. Crystal: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–1321, 1995.

[27] Y. Yang and H. Zhang. Html page analysis based on visual cues. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, page 859. IEEE Computer Society, 2001.

[28] Y. Zhang, N. Zincir-Heywood, and E. Milios. World wide web site summarization. *Web Intelligence and Agent Systems*, 2(1):39–54, 2004.