TOWARDS THE SEMANTIC WEB: THE AUTOMATION OF KNOWLEDGE ACQUISITION FROM THE MEDICAL WEB

"Mohammad Ali" Hashem Salti Eljinini

A Thesis Submitted for the Degree of Philosophy Doctor (PhD)

City University, London, UK
Centre for Health Informatics
School of Informatics

February 2007

ABSTRACT

The current web contains a wealth of information in the form of natural text. In the medical domain, the number of documents related to healthcare is already large and continues to grow at exponential rate. Today's desktops can retrieve millions of web documents but can understand none. HTML documents are made to be read and understood by humans and not by machines. In recent years, researchers have been working on the development of new languages for the semantic web. Annotating web documents with semantic metadata will enable contents-guided searching and reasoning which will lead the web to its full potential. Despite all the advances in this area the web at large is still un-semantic. It is impractical to go back and annotate the current web with semantic tags manually. Such a process is labour intensive, prone to errors, and requires expertise with the new complex technologies.

The objective of this work is the development of a novel methodology for extracting useful information from the medical web to be structured and ready for the semantic web. To accomplish this task, three sets of chronic disease-related websites have been downloaded, analysed and studied in depth. The study has revealed a common set of concepts along with their attributes which were used in the construction of the ontology. An information extraction system has been developed that utilises the ontology for extracting common structures from unseen chronic disease-related websites.

AKNOWLEDGMENTS

I am grateful to the following people and organizations:

- My first advisor Professor Ewart R. Carson for his constant support, guidance, patience, and many helpful comments throughout this research which would not have been possible without him.
- My second advisor Dr. Nazar A. Sarhan for his invaluable support, guidance, and many fruitful discussions.
- Dr. Abdul V. Roudsari (Head of the Centre), staff members and research students at the Centre of Health Informatics at City University for their support and encouragements.
- Dr. Ayman Alnsour (Dean of the Faculty) and my colleagues at the Faculty of Science and IT at Alisra Private University for their encouragement and useful comments.
- My parents for their love, support, and bringing me up to be what I am today.
- My wife Dina for standing by me all these years who has given me unlimited support. She has been always there when I needed her. Also I should mention my children, Omaima, Yumna, and Hashem for their unconditional love.
- The US National Library of Medicine for allowing me to use MMTx API with this research. Also to Jim Mork from the MMTx team at NLM for helpful discussions regarding MMTx.
- The Protégé-2000 team for making their Ontology Editing Tool available.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

API	Application Programming Interface
CSS	Cascade Style Sheet
DAML	DARPA Agent Markup Language
DARPA	Defence Advanced Research Projects Agency
DCMI	Dublin Core Metadata Initiative
DL	Description Logic
DOM	Document Object Model
DTD	Data Type Definition
FOL	First Order Logic
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTML	Hypertext Markup Language
IDDM	Insulin Dependent Diabetes Mellitus
IE	Information Extraction
ILP	Inductive Logic Programming
IR	Information Retrieval
KA	Knowledge Aquisition
KB	Knowledge Base
KR	Knowledge Representation
MUC	Message Understanding Conference
NIDDM	Non Insulin Dependent Diabetes Mellitus
NLP	Natural Language Processing
NP	Noun Phrase
OIL	Ontology Inference Layer
OWL	Web Ontology Language
PAL	Protégé Axiom Language
PDF	Portable Document Format
POS	Part-Of-Speech
PP	Preposition Phrase
RDF	Resource Description Framework

SAX	Simple API for XML
SQL	Structure Query Language
SVG	Scalable Vector Graphics
UML	Unified Modeling Language
UMLS	Unified Medical Language System
URI	Universal Resource Identifier
URL	Universal Resource Locator
VP	Verb Phrase
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

Chapter One

Introduction

"for fifty years, people have dreamt of the concept of a universal information database - data that would not only be accessible to people around the world, but information that would link easily to other pieces of information so that only the most important data would be quickly found by a user."

-Kevin Hughes, 1993

1.1 Background and Motivation

The current web contains a wealth of information in the form of natural text. In the medical domain, the number of documents related to healthcare is already large and continues to grow at an exponential rate. Today's PCs can retrieve millions of web documents but can understand none. HTML (HyperText Markup Language) documents are made to be read and understood by humans and not by machines. On close examination of HTML Tags (Table 1.1) it can be observed that these tags instruct the browser on how to render HTML documents on the screen, for example, where a new paragraph begins, what type of font and what size to use, which piece of text is a section heading, etc. Besides the ability to format text, HTML allows web authors to create relationships between pieces of text by the use of hypertext links. Linking information together across the globe is what made HTML powerful and very popular. Today the web is considered to be one of the major sources of information in the world.

HTML Tags	Description
<html>...</html>	Beginning and end of HTML document
<head>...</head>	Encloses title and other information like meta and style tags
<body>...</body>	May include attributes to set background and text colour, etc.
...	Headline – Largest
...	Headline – Smallest
...	Bold text
...	Italic text
<u>...</u>	Underlined text
...	Enclose attributes to set font size and colour
...	Paragraph
 	Line break
<dl>...</dl>	Creates a definition list
<dt>...</dt>	Encloses each definition term
...	Creates a numbered list
...	Precedes each list item, and adds a number ,letter, bullet image etc
...	Creates a bulleted list
<img src="name">	Adds an image – may have attributes for alignments, and border settings
<hr>	Inserts a horizontal rule
<table>...</table>	Creates a table
<tr>...</tr>	Sets off each row in a table
<td>...</td>	Sets off each cell in a row
<th>...</th>	Sets off the table header (a normal cell with bold, centred text)

Table 1.1 *A list of the most common tags used to format HTML documents*

Web users looking for information most likely will experience difficulties, and some will get lost in hyperspace. Figure 1.1 shows a search cycle that a user might have to go through while searching the web for information. The user starts with an initial query consisting of one or more terms. The search engine's database is looked up and a list of results that normally consists of hundreds of thousands of links is generated and presented to the user (with a default of ten results per webpage). The user is then confronted with choices; either to explore the list in order to locate websites with the needed information, or to modify the initial query and start all over again.

Today's search engines constitute state-of-the-art technology in Information Retrieval (IR), but they are plagued by several problems such as the limited coverage, low precision, redundancy, and duplicates. Despite all the advances in

search technologies, these problems still exist. This is due to the way search engines process web documents. Search engines process web documents as bags of words (Brin & Page, 1998) and so they employ statistical techniques to find word relevancy for indexing.

In the mid 1990s researchers turned to new ways to represent knowledge on the web (Decker et al., 2000) such as the creation of languages such as XML (Extensible Markup Language) and RDF (Resource Description Framework). The main goal is to add semantic structures to websites. XML and RDF allow web authors to represent knowledge on their websites in ways closer to representing knowledge inside databases. This is done by creating user-defined attributes, attribute values, and relations among these attributes. This would open the door for a new generation of smart search engines.

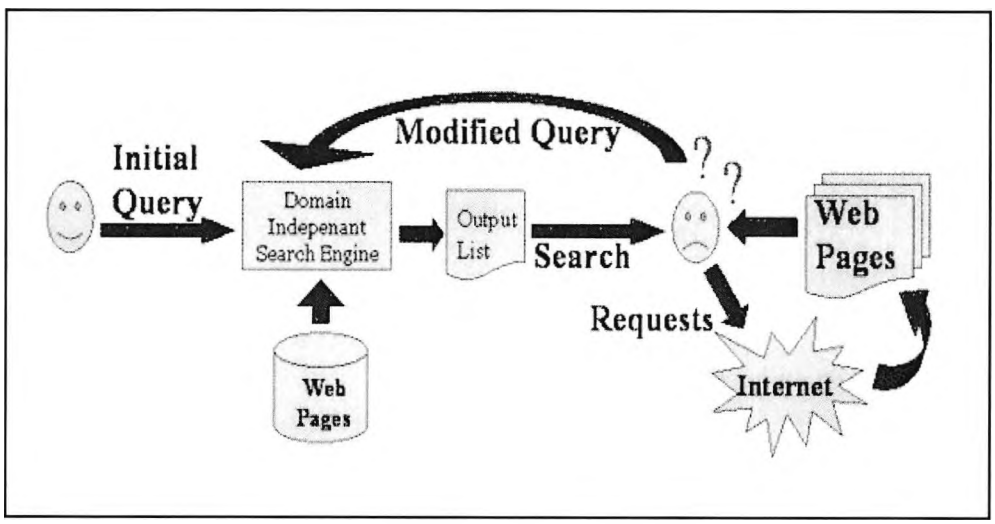


Figure 1.1 *The search cycle for users seeking information on the Web*

Annotating web documents with semantic metadata will enable contents-guided searching and reasoning which will lead the web to its full potential. While there are many definitions for the term "metadata" existing today such as "data about data", the term "semantic metadata" mentioned here goes beyond providing a list of keywords into adding meanings and structures through conceptualisation of the domain of discourse.

Despite all the advances in this area the web at large is still un-semantic. This is due to two factors:

1. It is impracticable to go back and re-design or even annotate hundreds of millions of websites with semantic tags manually. Such a process is labour-intensive and prone to errors.
2. Authors of new websites will need to learn the new complex technologies to be able to use in their websites. This requires expertise with the new complex technologies.

Thus before a significant change is seen, the web will remain un-semantic for a long time. This leads to the main thesis of this work; how to automatically annotate a website with semantic metadata.

1.2 Aim and Objectives

As shown in the previous section the need to automate the process of annotating web documents with semantic metadata is a very important step towards the semantic web. Research in this direction is still in its infancy and many problems still exist to this day. In this study the major problems will be examined and an attempt will be made to contribute towards their solution.

The aim is to develop a new framework that contributes towards the automatic construction of a semantic layer to a given website. The semantic layer is made of "nuggets of knowledge" that give a fair description about the website and its content. The main part of this work is to explore methods for extracting specific knowledge[1] from web documents. To accomplish this task, there is first the need to reduce the problem to a specific domain which was chosen to be diabetes management (in particular and other chronic diseases in general). This will allow one to work with a well defined sublanguage and vocabulary. Hence, the hypothesis

[1] Sometimes the terms "knowledge" and "information" are used interchangeably. In this case the meaning is "knowledge acquisition from information stored on the web". Knowledge is a structured form of information that can be understood by machines.

is that annotation of web documents with semantic tags can be automated with the guide of a domain-specific ontology. Another hypothesis which will be examined is that similar websites possibly exhibit similar information structures. This would aid in the process of acquiring knowledge from such websites. In other words, this would allow one to cover more general domains (i.e. chronic disease-related websites) and possibly broader domains. Finding common structures between these websites would allow one to also maximise the extraction of useful information.

This leads to the following specific objectives:

1. Studying and analysing health-related web documents in order to identify the difficulties with processing these documents and contributing towards an effective solution. The types of knowledge to be extracted are also being identified. Also, to compare the contents of diabetes, asthma, and hypertension related websites in order to find common structures.
2. Developing ontology for the domain of diabetes and other chronic disease-related websites to guide the system as to which features should be extracted from a given website. Also, in the process, exploring the roles and types of ontologies; and examining the possibility of reusing and integrating existing ontologies to accomplish the task at hand.
3. Reviewing state-of-the-art approaches in information extraction, identifying problems with these approaches when applied to web documents and making a contribution towards solving these problems.
4. Designing and developing a framework which consists of the components needed for the extraction system and applying the solutions which are discussed in detail in subsequent chapters.
5. Studying and exploring ways to represent the output structures.
6. Applying the system to the domain of diabetes in particular and to other health-related areas in general.
7. Providing a comprehensive evaluation of the system under study.

1.3 Contents of Subsequent Chapters

Given the aim and objectives of this research work, the material presented in this thesis falls into three main parts. The first part (Chapters 2 – 4) is introductory. The second part (Chapters 5 – 6) includes a description of the application domain, diabetes management, and the system development and implementation. The final part (Chapters 7 – 10) presents experimental testing, analysis of results, assessment of the system, discussion, and conclusions. In the following paragraphs, the sentences in italic contain information to guide the reader on how to read this thesis.

In chapter two a brief discussion about the present and future of the web is provided. The second-generation languages of the web are outlined, together with how they relate to each other, the limitations of each technology, and how the web is evolving towards the semantic web. This is followed by a discussion of the problems and issues of automating knowledge acquisition from the web. There are a number of issues and problems with annotating websites with semantic tags, starting from processing HTML documents and ending with the final representation. Emphasis is given to the types of knowledge to be extracted from websites. *Sections 2.2 and 2.3 contain important background for readers who are not familiar with web technologies and section 2.4 is a must read to understand the general problem of the thesis.*

In chapter three the role of ontology in the semantic web is discussed. This begins with a discussion of the definitions of ontology, types of ontologies and methods used for designing ontologies. There is a presentation of some of the languages used to represent ontologies. This is followed by consideration of some development environments and in particular Protégé-2000 which is being used as part of this work. A discussion of how ontology may be utilised for information extraction is also provided. In the last part, two special ontologies are compared (UMLS and WordNet), with a focus on ontology reuse. *Readers who are not interested in review information about ontologies such as history, types, etc. may skip sections 3.2 , 3.3, 3.4, and 3.5 safely. It is recommended to read section 3.6 since it contains*

information about the utilisation of ontologies for information extraction which is important for this research work. Section 3.7 contains original work and should be read too.

In chapter four, information extraction is reviewed. The different state-of-the-art approaches used in information extraction are discussed. The problems associated with each method are explained. *Readers with a solid background on information extraction may skip this chapter. However, it is recommended to read sections 4.4 and 4.5 which discuss problems with current information extraction technologies and the author's critique of these systems.*

Chapter five is devoted to diabetes management, the chosen application domain. Diabetes management has been chosen because it is one of the most common chronic diseases in the world. The population of diabetes patients is very large and continues to grow at an alarming rate. Some 100 diabetes-related websites dealing with different aspects of the disease have been extracted for off-line experiments. In this chapter there is also a study of the characteristics of these websites. This study has been extended to cover two more types of chronic disease-related websites, namely asthma and hypertension. *Readers familiar with diabetes management can skip this chapter except for section 5.8 where the main contribution starts.*

During the course of this work, a system has been developed in two stages; in the first stage an initial java system was developed and used to study and analyse the sets of websites that were collected, while in the second stage the system was farther developed to become a general information extraction system that was used as a test bed. Chapter six is devoted to the development and implementation of this system. *Readers should read this chapter and the rest of the chapters where the system under study is presented, tested, experimented with, discussed, and conclusions are drawn..*

In chapter seven the experimental testing procedures are discussed. The methodologies used in the testing process are presented. Some case scenarios also

have been presented. Last, the procedures used for evaluating the components under study and the overall system have also been discussed.

Chapter eight provides an analysis of the results and an assessment of the system. During the first part of this research work three sets of chronic diseases websites have been identified and analysed. A full analysis of the results that were produced are presented in this chapter. Also, an analysis of the results obtained from experiments carried out with information extraction system are presented here.

Chapter nine provides an overall detailed discussion. This includes some issues and assumptions about the data that have been collected (i.e. the set of websites). Other issues that have been discussed are related to the methodologies that have been used, such as weaknesses and strengths. Finally, the value and usability of this research work have been discussed.

Conclusions are presented in chapter 10 where there is consideration of the extent to which the objectives have been met. Also, the contributions to knowledge are indicated, and finally recommendations are made for future work.

Chapter Two

The General Problem and Methodological Approaches

"If an engine of the future combines a reasoning engine with a search engine, it may be able to get the best of both worlds, and actually be able to construct proofs in a certain number of cases of very real impact. It will be able to reach out to indexes which contain very complete lists of all occurrences of a given term, and then use logic to weed out all but those which can be of use in solving the given problem."

-Tim Berners-Lee, 1998

2.1 Introduction

In recent years, researchers have been working on the development of a new generation of web technologies. The web of the future is named "The Semantic Web" which was originally proposed by Tim Berners-Lee (Berners-Lee, 2001), the inventor of the World Wide Web. The semantic web allows information to be maintained in a structured form and processed by machines easily. The semantic web can be thought of as a set of globally-linked databases.

In the first part of this chapter the second generation of web languages (beyond HTML) is discussed for representing and processing data on the web. The intention is to show first how the web evolves towards the semantic web, to show how these languages fit together, and then outline their limitations and weaknesses. It is important here to note that the main idea was to separate data content from visual representation. Note that in the unsemantic web the data are embedded within HTML code (as shown in Figure 2.1). Separating the data from HTML code means the data can be formatted and presented in many different ways without affecting the real content.

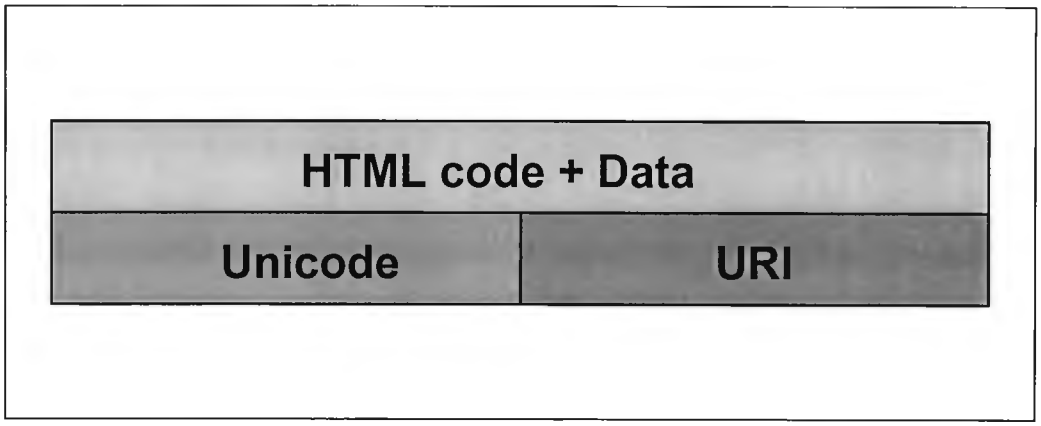


Figure 2.1 *The layers of the unsemantic web*

It is considered that HTML will never disappear, or at least not in the near future. This is because there will always be a need for rendering information on the screen, and HTML is good at doing that. What is needed is to add other layer(s) for machine understanding in the form of "knowledge structures". In this work the concern is with gathering the data from HTML documents and transforming data into nuggets of knowledge ready for the semantic web. This leads to the main thesis of this work: how to extract meaningful data that can be transformed into knowledge structures? Extracting information from an HTML document is not an easy task and there are a variety of problems needing to be addressed and discussed. The second part of this chapter is devoted to these problems along with methodological approaches that are discussed in general.

2.2 Data Structures on the Web: Present and Future

Before discussing the problems of extracting information from a given website and transforming it into structured format, first let us see what is meant by structured data and how these can be understood by a machine. Figure 2.2 shows a table of structured data that is familiar to people working with databases. The first row (Header) shows a list of predefined attributes. These attributes provide a description of the information to go into each column. Each row resembles a record, or an instance. The ability to query the table becomes very efficient using the Structured

Query Language (SQL). Also, the ability to mine the data in order to discover new knowledge becomes possible.

On the web some data are presented in tables (called semi-structured data) which can be transformed into a structured format with relative ease using techniques such as wrapper generation (Kushmerick et al., 1997). Examples of tabular data are:

1. List of automobiles for sale (i.e. make, year, mileage, etc)
2. List of flights (i.e. departure location and time, arrival location and time, etc)
3. List of apartments for rent (i.e. number of beds, address, cost, etc)
4. List of books for sale (i.e. title, ISBN, number of pages, price, etc)
5. List of job posting (i.e. job title, salary, etc)

These techniques are discussed in more detail in chapter four, when reviewing the Information Extraction (IE) field. However, most data on the web are in the form of free or unstructured natural text. This creates the need for other methodologies to deal with natural text.

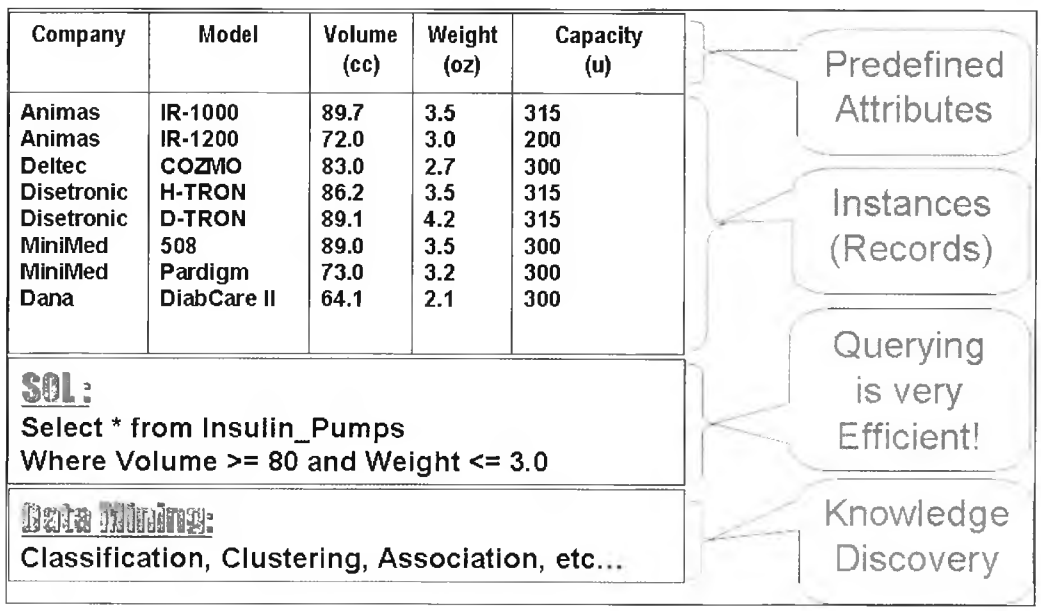


Figure 2.2 *Structured data that can be understood by machines*

Figure 2.3 shows an example of an attribute-value set extracted from unstructured text. The extracted values are filled inside the slots of a predefined template. Another way of representing the same data is by the insertion of XML (Extensible Markup Language) tags as shown in Figure 2.4. While XML is becoming a global standard for information exchange over the web, obviously, it is becoming a candidate technology for managing information over the Internet. XML documents are text files that contain user-defined tags. This allows authors to create markup languages for any field.

The elements inside an XML document are presented in a hierarchical manner, in this example; element "Disease" is the root. There is only one root in an XML document. Elements may have zero or more other elements nested inside them. An XML document can be presented as a tree structure, a powerful data structure used in Computer Science.

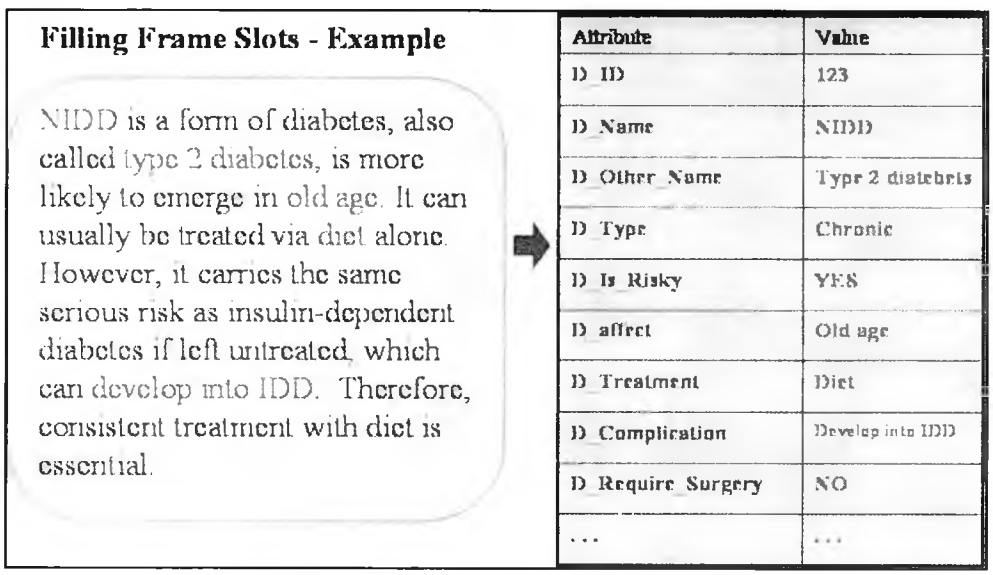


Figure 2.3 *Information extraction from free text into a predefined template*

Now, if everybody agrees on using a certain set of user-defined tags in their XML documents on the Internet, computer programs can process the file effectively and efficiently.

<Disease>
<D_ID> 123 </D_ID>
<D_Name> NIDD </D_Name> is a form of diabetes, also called
<D_Other_Name> type 2 diabetes </D_Other_Name> , is more
likely to emerge in <D_affect> old age </D_affect> . It can usually
be treated via <D_Treatment> diet </D_Treatment> alone.
However, it carries the same serious risk as insulin-dependent
diabetes if left untreated, which can <D_Complication> develop
into IDD </D_Complication> . Therefore, consistent treatment with
diet is essential.
</Disease>

Figure 2.4 *XML tags are used to identify terms (as values) with attributes*

An XML document may use a Data Type Definitions (DTD) file, to define and confirm the document's structures. For example, which elements and attributes are permitted inside the XML document? Later the World Wide Web Consortium (W3C) has introduced an alternative to DTDs, which is called XML Schema. The following example shows how attributes are defined in XML Schema:

```
<xsd:schema>
  <xsd:element name="company"
        type="xsd:string"/>
  <xsd:element name= "model"
        type= "xsd:string"/>
<xsd:element name="volume"
        type="xsd:float"/>
...
</xsd:schema>
```

Schemas follow XML syntax, offering many improvements over DTDs, and have rich grammatical structure for XML documents. Many people and organisations have created DTDs and schemas for a broad range of categories, such as patient records, medical prescriptions, and finance. A powerful feature of XML is the separation between data content and its representation. An XML document can be transformed into different formats such as HTML, Portable Document Format (PDF), and Scalable Vector Graphics (SVG). It can be viewed on different devices such as WAP phones, digital TVs, and medical devices.

This is where Style sheets come in. Style sheets contain instructions that define how the elements inside an XML document should be transformed to meet clients' viewing needs. There are two types of style sheets. The first one is known as Cascading Style Sheets (CSS) and is mostly used with HTML; however, they also can be used with XML. CSS does not have the capability to perform any computations; they are mainly used to define global templates, such as what colour, size, and font to be used with headings, paragraphs, etc. It provides consistency with every page on the website. The second and newer type of style sheet is the Extensible Style sheet Language (XSL). It consists of two parts: 1) The Extensible Style sheet Transformation Language (XSLT). 2) The XSL formatting objects. The basic processing of XSLT is pattern matching, inherited from languages such as PERL. XSLT consists of a set of template rules, such as, "If a condition is met in the input, then generate the following output". Once an XML document is transformed into another form, an XSL-formatting object can be used to describe margins, fonts, colours, and other formatting objects to be used with the new resultant document.

To be able to interpret XML documents beyond the realm of XSL, there are two Application Programming Interfaces (API) that can be used for this purpose, the Document Object Model (DOM) and the Simple API for XML (SAX). These APIs are implemented in many programming languages such as Java and C++. They can be downloaded from the Internet and are ready to be used in any application. DOM transforms an XML document into a tree structure and stores it completely in memory; hence an XML document does resemble a tree with its hierarchical structure. DOM offers a powerful set of functions for manipulating the tree structure. SAX, on the other hand, does not store the XML document in memory. Data are passed to the application from the XML document as it is encountered. SAX methods are event-driven, which means that they are invoked when a markup such as start tag or end tag is reached. To summarise, DOM consumes more memory space, is fast, has richer functions, and allows editing of documents before it is rewritten to file. SAX takes up much less memory, is slower, and does not

allow modification of the XML document. Which one to use? This depends on the specifications of the application to hand.

Much of the technology discussed above is about how to structure and represent data on the web. As stated above, the major problem with XML is the need for global vocabulary (or schema) that everybody can agree with and therefore data can be shared between applications. It is not possible just to merge XML files with different vocabularies to obtain a global one. This is because XML documents are represented as hierarchical trees, which means that the relation between parent and children is one-to-many, thus making it impossible to merge two XML files and keep the data consistent.

In recent years researchers realised the limitations of XML to meet the expectations of the semantic web and began to work on other richer languages such as RDF (Resource Description Framework) and RDF-Schema, OIL (Ontology Inference Layer), DAML+OIL (DARPA Agent Markup Language + OIL), and finally OWL (Web Ontology Language). These languages provide features other than structuring data and will be discussed briefly in the following section.

2.3 Languages for the Semantic Web

Here, the interest is mainly in the features each language provides so their differences will be examined, together with how they relate to each other. The reason this study was conducted is first to see how the semantic web is evolving and second to see which language is best to adapt to our work. The architecture of the semantic web is based on many layers as proposed by Tim Berners-Lee, see Figure 2.5. In his proposal it should be noted that all layers are built on XML and they have XML syntax. The next layer is the RDF and RDF-Schema which are two W3C standards and are considered to be the basis of the Semantic Web. See Berners-Lee (1999) for full details on the layers of the semantic web.

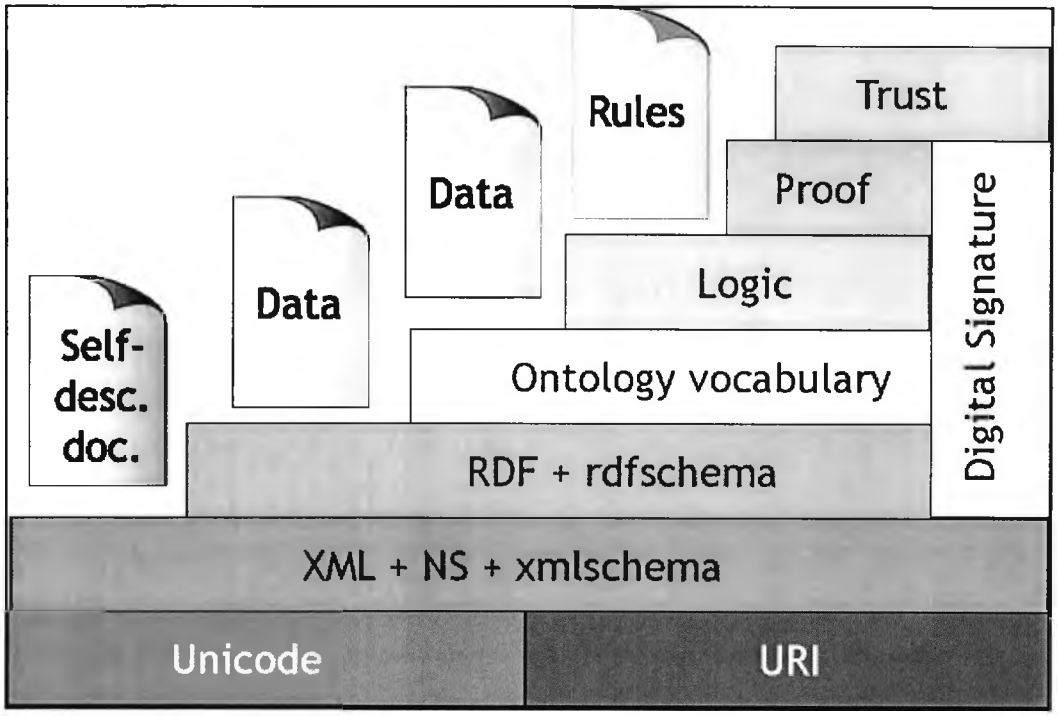


Figure 2.5 *The layers of the semantic web as presented in (Berners-Lee, 1999)*

2.3.1 RDF (Resource Description Framework) and RDF-Schema

RDF (Manola & Miller, 2003) is a language that builds on XML to describe web resources and provide metadata (data about the data). Unlike the relation between parent and children represented by XML, in RDF relations can be many-to-one and one-to-many. Since RDF is made of statements where each statement is equivalent to one cell in a relational database table. RDF brings the expressive power of relational database to the web (i.e. joins and views).

RDF provides a simple data model in the form of statements where each statement consists of a triple <Subject, Property, Object> which represents a well-defined fact. Figure 2.6 shows an example of an RDF statement. This statement says that "DiabetesUK" is the publisher of the website defined by the address "http://www.diabetes.org.uk".

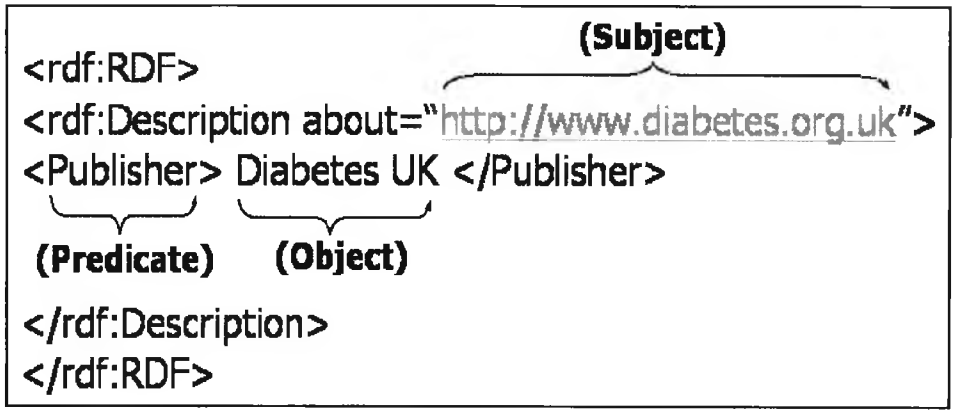


Figure 2.6 *Example of RDF triple*

A subject is a resource identified by a URI and can be anything on the web such as a web page, part of a web page, a set of web pages, or even non-accessible entities such as a name or a date. The predicate (property) represents a specific attribute or relation that describes a resource. The object is the value of this property. Statements can be chained together to form a graph. This can be made possible by making an object of one statement the subject of another.

In the semantic web hierarchical structure, the RDF and RDF-Schema are shown on the same level but actually the RDF-Schema is the next level in the hierarchy (i.e. on top of RDF). The RDF-Schema extends on RDF by providing externally specified semantics to specific resources (Broekstra et al., 2001). Figure 2.7 presents a good example of RDF and RDF Schema in graph format.

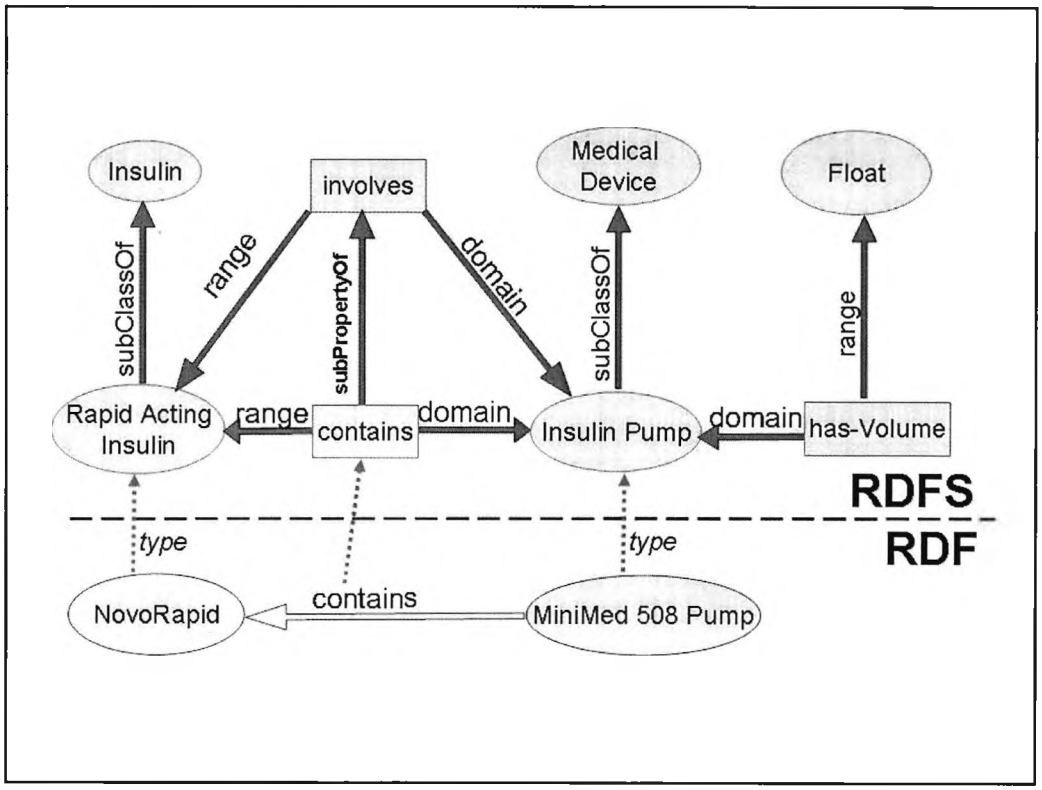


Figure 2.7 *Example of RDF/RDF Schema in graph format*

In other words, the RDF-Schema provides ways to define domain-specific vocabularies and structures for conveying metadata about web resources. A web resource can be just about anything like a webpage, somebody's name, an email, a product, a concept, etc. In the example above RDF Schema describes insulin pumps as a subclass of medical devices and may contain rapid acting insulin which is a subclass of insulin. Now, when constructing RDF statements (i.e. the one shown below the dashed line in Figure 2.7), such statements must conform to this schema as presented in the example.

The following statements describe some parts of the graph presented in Figure 2.7 in RDF (textual):

<rdf:Description rdf:about="NovoRapid">
<rdf:type rdf:resource="Rapid Acting Insulin"/>
</rdf:Description>

<rdf:Description rdf:about= "MiniMed 508 Pump">
<contains rdf:resource="NovoRapid"/>
</rdf:Description>
<rdfs:Class rdf:ID="Rapid Acting Insulin"/>
<rdfs:subClassOf rdf:resource="Insulin"/>
</rdfs:Class>

This example is shown here as a demonstration on how RDF and RDF Schema can be used in real world examples. The interested reader can see Manola & Miller (2003) for a detailed coverage on RDF and RDF Schema.

2.3.2 OIL and DAML+OIL

RDF and RDF-Schema fail to meet the expectation of the semantic web because they are not expressive enough for a fully-fledged knowledge representation model on the web. Looking again at Figure 2.5, it can be seen that there are other layers on top of the RDF and RDF-Schema that are needed to fulfil the mission of the semantic web. These layers extend on the RDF-Schema to provide formal semantics and reasoning facilities in the form of ontologies. An ontology is a formal, explicit specification of a shared conceptualisation (Gruber, 1993), that has been developed by the AI community to facilitate knowledge sharing and reuse. Ontologies, which are discussed in more detail in the next chapter, will play a crucial role in knowledge management on the semantic web.

The IST key Action On-To-Knowledge, a European project, has developed OIL (Ontology Inference Layer) to employ the full power of the ontological approach to facilitate knowledge management on the web (Fensel et al., 2000). OIL, as a formal knowledge representation language that extends on RDF and RDF-Schema, has become a candidate for the next (logical) layer on the semantic web.

At the same time the DARPA Agent Markup Language (DAML) was developed by a U.S. government-sponsored group in cooperation with the W3C as a knowledge representation language for expressing more sophisticated RDF class definitions

than permitted by RDF-Schema. Later the DAML group has joint efforts with the OIL group and the result of these efforts became DAML+OIL, a language for expressing far more sophisticated classifications and properties of resources than RDF-Schema.

2.3.3 Web Ontology Language (OWL)

The DAML+OIL language has gone through several revisions by the W3C group. In the year 2001 it became known as the Web Ontology Language (OWL) and in the year 2004 OWL became a standard. The following example shows a small part of an OWL file describing publications:

```
<owl:Class rdf:ID="Author">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="author">
  <rdfs:domain rdf:resource="#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="date">
  <rdfs:domain rdf:resource="#Event"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="price">
  <rdfs:domain rdf:resource="#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
```

Table 2.1 presents the complete list of OWL constructs that are used along with RDF Schema constructs to build ontologies on the web.

owl:AllDifferent	owl:equivalentProperty	owl:priorVersion
owl:allValuesFrom	owl:FunctionalProperty	owl:Restriction
owl:AnnotationProperty	owl:hasValue	owl:sameAs
owl:backwardCompatibleWith	owl:imports	owl:sameIndividualAs
owl:cardinality	owl:incompatibleWith	owl:someValuesFrom
owl:Class	owl:intersectionOf	owl:SymmetricProperty
owl:complementOf	owl:InverseFunctionalProperty	owl:Thing
owl:DatatypeProperty	owl:inverseOf	owl:TransitiveProperty
owl:DeprecatedClass	owl:maxCardinality	owl:unionOf
owl:DeprecatedProperty	owl:minCardinality	owl:versionInfo
owl:DataRange	owl:Nothing	
owl:differentFrom	owl:ObjectProperty	
owl:disjointWith	owl:oneOf	
owl:distinctMembers	owl:onProperty	
owl:equivalentClass	owl:Ontology	

Table 2.1 *The list of OWL constructs used for constructing ontologies on the web*

OWL has three increasingly expressive sublanguages:

- OWL-Light: Supports the need for a classification hierarchy and simple constraints.
- OWL-DL: DL is for "Description Logic", the formal foundation of OWL. This sublanguage supports the need for maximum expressiveness and at the same time hold on to computational completeness and decidability.
- OWL-Full: Supports the need for syntactic freedom of RDF, but provides no computational guarantees.

The semantic web layers can be summarised as follows: XML provides the syntax for defining a customised tagging scheme and the XML-Schema adds some constraints. RDF provides a flexible approach for representing data with simple semantics and RDF-Schema provides the vocabulary for describing these data (resources) as classes and properties. OWL adds more vocabulary for describing

properties and classes, and also adds facilities for reasoning, which qualifies for an expressive ontology language on the web.

This layered set of languages provides an infrastructure for information on the web to be processed and shared by computer applications more efficiently and effectively. However, the web becomes much more complex than it should be. Many web authors are artists; their main concern is to design beautiful catchy views of their web documents for human consumption. The other face of the web (the machine-understandable face) needs new hard-to-learn skills especially for non computer scientists. This supports the need for the automation of the construction of this face.

2.4 Knowledge Acquisition from the Web: Problems and Issues

The last section has demonstrated the importance of constructing knowledge structures automatically for a given web (HTML) document. In this section the issues that were confronted within this process are discussed. First, consider what types of knowledge need to be extracted from web pages. Second, processing HTML document by computer programs is not an easy task. This is mainly due to the fact that these documents are written by people with different qualifications and backgrounds. The problems and challenges to be dealt with are presented in the next subsection.

The work here stems from the field of information extraction; in the next subsection there is a discussion of the problems of information extraction from web documents and a general outline of the approach being proposed is presented.

2.4.1 Types of Knowledge on the Web

Most researchers working on the topic of information extraction from web pages are concerned with the subject that is incorporated within these pages. For example, a system extracting information from Universities' websites is concerned with information about students, courses, faculty staff, research projects, etc. (Craven et al., 2000). Universities' environments are very dynamic and subject to continuous change. For example, four years from now there will be different groups of students, courses, research projects and maybe some new faculty staff too. The extracted information becomes outdated quickly. Also, this problem is most experienced today by people who search for information on the Internet. On close examination of the results returned by search engines, it is found that many websites contain outdated material. Hence, it is important to save the date when a website was last updated.

When extracting knowledge from a website, it is important to know that there are two main types of knowledge needing to be extracted:

1. Website-content knowledge that is embedded within the pages, and essentially that is the subject(s) covered by the website.
2. Website-about knowledge, for example, the title of the website, information about the website's author, date of last update, purpose of the website, the group it belongs to, links structure and organisation of topics, the language used, etc.

From consideration of the University example the importance of knowing when the website was last updated has been demonstrated. Other information about the website is important as well. It should be mentioned here that there have been some initiatives for standardising a set of metadata tags to be used to describe websites (i.e. the second type of knowledge mentioned above) such as the Dublin Core Metadata Initiative (DCMI). However, this type of knowledge will not be investigated any further for this work. It has been presented here for the sake of

completeness and may be considered for further investigation in future work. The second type (website-about knowledge) will be studied in depth and presented in a subsequent chapter.

2.4.2 HTML Documents on the Web

Processing HTML document with computer programs is not an easy task. HTML documents on the web have been written by different authors from all over the world. Many documents are cluttered with images, commercial ads, scripts, and many other irrelevant elements. Such elements are considered to be noise that must be dealt with when parsing HTML documents. A good way to work around this is to transform HTML document into XML format. XML documents can be processed much more efficiently and there are many open source tools that can aid in this process. Once the text in web documents has been correctly extracted and examined, it has been noticed that many paragraphs lack correct grammatical structures, for example the lack of using full stops as a punctuation mark which normally is placed at the end of a declarative sentence to indicate a full stop to that sentence. Also, one of the main problems is the lack of having complete sentences. Such problems deprive Natural Language Processing (NLP) tools from parsing sentences correctly. These problems and others are explained in more detail in chapter 6 along with solutions that have been used in this work.

2.4.3 Adaptability of Information Extraction (IE) Systems

Most research about online information extraction is done on the basis of extracting information from very narrow, domain-specific sources such as job postings, apartment rentals, automobile sales, and other tabular data. The success of their techniques was due to the fact that the internal organisation of these documents is very similar (i.e. semi-structured). The main ingredient of today's IE systems is the pre-defined template which consists of a set of attributes such as title, author, price, etc. and the set of extraction rules that is used to extract values for these attributes. It can be argued that for this reason IE systems are hard to adapt to new domains.

The pre-defined template needs to be changed to fit the new task. The manual construction of pre-defined templates (*and actually the extraction rules*) is labour-extensive, time-consuming, and often requires domain expertise.

In recent years researchers began investigating the problem of adaptability in IE systems. With the amount of text available in free form on the web which contains invaluable information for every domain from education to medicine, adaptive IE has become an extremely active area in research. A desirable scenario would be the ability to purchase a generic IE system off the shelf and actually use it for any task with minimum work done by the user. Why is this an important problem in this project? Because, one wishes to be able to extract useful information from any websites or at least from a relatively broad domain such as the domain of chronic disease-related websites.

Current state-of-the-art IE methodologies require users to provide the system with examples to learn from. The examples are in the form of documents that are tagged with the correct answers. Once these documents are processed, the system produces a set of rules that can be used to extract similar patterns from new unseen documents. Today research activities on adaptive IE paradigm are centred around two main tasks:

1. Experimenting with different machine learning algorithms with the aim of minimising the required supervision; for example, by providing less tagged examples to the system.
2. Working on the development of more user-friendly interfaces; for example, by providing means for users to tag their own set of examples easily, and set-up the required tasks with minimum interaction with the system.

During the initial investigation and the analysis of the data collected from the web for this research work, it has been observed that many concepts are common in many chronic disease-related websites and possibly in other domains. The collection of common concepts, once analysed and developed, can greatly contribute

towards the above mentioned problems; possibly in the form of a library of templates (and extraction rules) that can be reused in many IE tasks. It should be mentioned here that templates in this work are presented and implemented as objects. This would allow for the mapping of real-world objects more efficiently. Also, templates implemented as objects are extensible and more importantly are reusable. This would increase the efficiency of IE systems and become more adaptable.

The general methodology that was developed for this project works as follows:

1. Find out the concepts that are common to all chronic disease-related websites, which are to be fixed in predefined sets of templates (objects).
2. When processing a new website; determine the category of each page on the website and see if there is an existing template for that category, or else to construct a new one if possible. This process becomes possible with the aid of ontology.

Several ontologies have been investigated that could be valuable in this research, such as WordNet (a general ontology) and UMLS (a domain-specific ontology). Protégé-2000 has been used, an ontology modelling tool, to work with both ontologies. The work is reported in the next chapter.

It has been hypothesised that websites with similar contents have common structures. For example, many of the diabetes-related websites offer information about diabetes, types of diabetes, treatments, symptoms, insulin management, food and diet, products, drugs, complications, etc. Many websites organise these topics in certain ways that resemble very much an ontology of taxonomic hierarchy, which shows how topics are related.

Three sets of chronic disease-related websites have been extracted from the web and analysed. An ontology has been constructed that contains common structures found

on these websites. A prototype for extracting useful information from similar websites has been constructed. The ontology is used to guide the information extraction system to relevant blocks of text. The work is discussed in detail in chapter 6.

2.5 Conclusion

In this chapter second generation web languages have been discussed in general terms and the functionality and role of each language outlined, together with how they relate to each other. A brief account has been given on how the semantic web is evolving, with an explanation of the main problems facing communities who want to learn and apply these technologies to their websites. Also, it has been noted that it is impossible to go back and annotate millions of websites manually. It has been argued that the ideal solution to this problem is to gather the data from HTML documents and transform them into nuggets of knowledge ready for the semantic web. In the second part of this chapter the problems and methodologies of extracting information from websites in general have been considered. One particular solution to aid the process of knowledge acquisition from the web is with the guide of ontologies. In the next chapter an overview of ontologies and their role are presented.

Chapter Three

Review: The Role of Ontologies

> *"An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e., its ontological commitment to a particular conceptualisation for the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualisation) by approximating these intended models."*
>
> - Guarino, 1988

3.1 Introduction

Ontology has its roots in philosophy as the science of being and existence. Philosophers such as Heraclites and Aristotle, Kant, and many others tried to answer questions such as: what is being? what categories are common to all beings? Questions like these led the philosophers to work on the categorisation of entities that exist in our world (Sowa, 2000). The philosophers organised the categories in hierarchies, but they could not agree on the exact content of these hierarchies. Aristotle presented ten basic categories for classifying anything. He placed the term "Being" at the top of his ontology which is then divided into Substances and Accidents, while Heraclites placed the term "Thing" at the root and divided things into tangible and intangible.

Today, ontology has become a popular topic in many fields that ranges from manufacturing and enterprises to medicine. Many branches of computer science are utilising ontologies to solve problems in more effective ways. This is done by

representing ontologies formally where they can be understood, shared, and manipulated by computer programs. In Natural Language Processing (NLP) ontologies are being used to characterise words meanings (Soderland et al., 1995; Bodenreider et al., 2002).

More recently, research in information retrieval has evolved towards semantic information retrieval, where ontologies are being used to improve the search task with intelligence. This is done by utilising the meaning of the terms to be searched for (Gondy & Hsinchun, 2001; Srinivasan et al., 2002). One of the largest sectors that utilises ontologies is the healthcare sector and that is because of the large variety of medical terms used by practitioners. There are many medical ontologies existing today such as the UMLS (Bodenreider, 2004), GALEN (Rector et al., 1994; 1995; 1997), and SNOMED-CT (Elkin et al., 2006). Ontologies help in the organisation, management, and retrieval of medical knowledge stored in heterogeneous sources.

Ontology has also become a hot research topic in Artificial Intelligence (AI). The most referenced definition of the term ontology in the literature is "an ontology is an explicit specification of a conceptualisation" (Gruber, 1993) where conceptualisation consists of entities called objects and the relationships between these objects. In Guarino (1998a) this definition has been extended to state that only a partial account of a conceptualisation is specified. The most challenging problems for researchers in AI are ontology integration, sharing and reuse. Probably these challenges are mostly felt by the semantic web community; if the future web is visualised as a huge, global network of web ontologies, then web agents of the future must be able to share and reuse these ontologies.

As stated in the objectives, the development of ontology for the domain of diabetes websites is needed to guide the system as to which features should be extracted from a given website. This chapter provides an overview of ontologies. First, the types of ontologies and the principles for designing ontologies are explored. Many languages have been developed to formally represent ontologies and many development environments have been developed to help knowledge engineers to

construct their ontologies. A number of formal languages used to represent ontologies and some of the environments used are presented, in particular Protégé-2000 which is used as part of this work. In the last part of this chapter the focus is on two ontologies, UMLS and WordNet, and a comparison between the two by conducting some experimentation to determine which one is more suitable for this work.

3.2 Types of Ontologies

Many ontologies have been developed during the 1990s. Ontologies can be categorised based on several characteristics such as contents, usage, and internal structure. There are two types of ontologies based on content; these are general ontologies and domain-specific ontologies. General ontologies capture general knowledge about the world, and domain-specific ontologies capture the knowledge in a specific domain. In the following sections each content-base type is discussed and a number of well-known ontologies, their purposes, usages, and internal structures are presented.

3.2.1 General Ontologies

General ontologies contain terms related to general things such as events, time, space, causality, etc. In this section three well known general ontologies are presented. These are Cyc Common Sense Ontology, SENSUS, and WordNet. All of them have been developed since the 1980s and early 1990s and have been used in many research projects.

3.2.1.1 Cyc Common Sense Ontology

The Cyc project began in 1984 with a goal to formalise human common sense knowledge in the form of terms, rules, and relations, to be used in Knowledge Base (KB) applications (Lenat, 1995). The main components of Cyc are a KB, an inference engine, and a knowledge representation language. The KB is made up of

several layers organised by generality (Figure 3.1). The top layer facilitates an Upper Ontology which consists of the most general concepts (Figure 3.2). The second and third layers contain a large set of short, simple assertions organised in subsets called Microtheories.

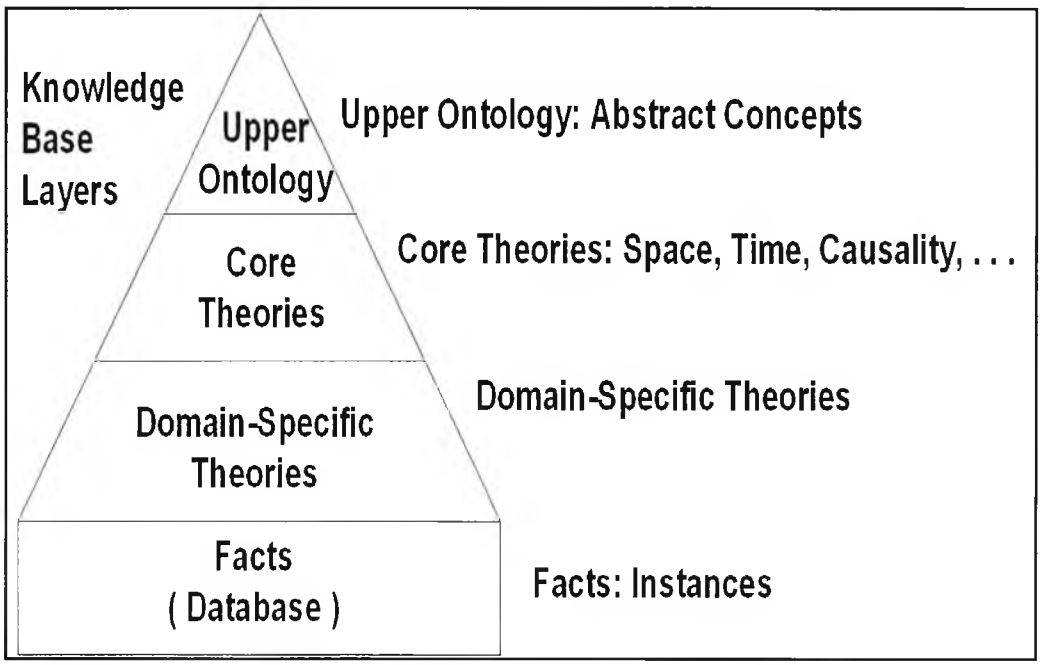


Figure 3.1 *Cyc knowledge base layers (adapted from Cyc website)*

Microtheories are divided into two parts; 1) Core which represents general facts such as space, time and causality. 2) Domain-specific which presents a large variety of topics organised in hierarchies. Microtheories help in indexing and coping with inconsistency in the KB. The bottom layer is a database full of world facts.

Cyc has also a Natural Language Processing (NLP) subsystem made up of lexicon, syntactic parser, and semantic interpreter. The aim of the NLP subsystem is to solve ambiguity in natural language text.

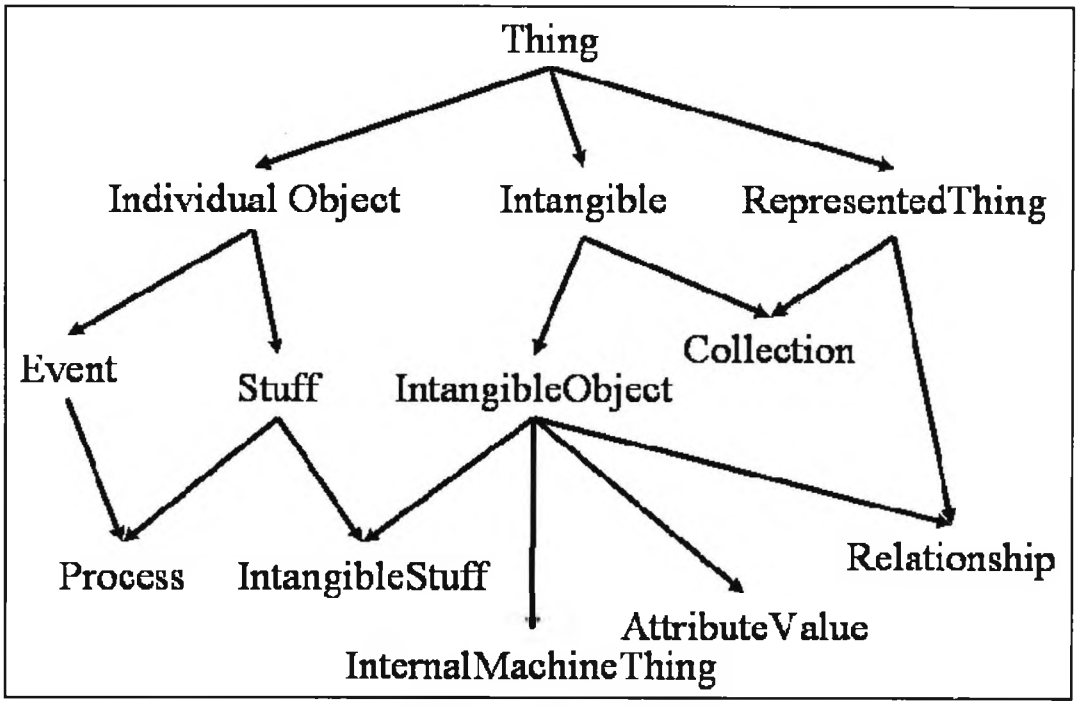


Figure 3.2 *Cyc upper level ontology (Lenat, 1995)*

Recently, parts of WordNet, MeSH, FCA World-Facts, and other ontologies have been integrated within the Cyc system. Cyc KB is formally expressed in the CycL language which has been derived from first order predicate calculus. In the late 1990s Cyc has turned parts of their work to open source under the name OpenCyc. Their goal is to allow researchers to use and contribute to the project.

3.2.1.2 SENSUS

The Information Sciences Institute (ISI) produced this broad-coverage, generic ontology by the integration and adoption of several resources like the PENMAN Upper Model, the ONTOS ontology, WordNet, the LDOCE semantic categories for nouns, and the Harper-Collins Spanish-English bilingual dictionary (Knight & Luk, 1994). The SENSUS ontology has over 70,000 concepts covering a wide range of different fields. The purpose of this ontology is to be used in NLP systems mainly for machine translation. SENSUS can also serve as a top-level ontology when building domain-specific ontologies. However, its top level ontology is called Ontology Base (OB) and is made up of around 400 terms. These terms are manually

generated from merging the PENMAN upper model and ONTOS ontology. Figure 3.3 shows the top level layer of PENMAN upper model which was used in the SENSUS ontology.

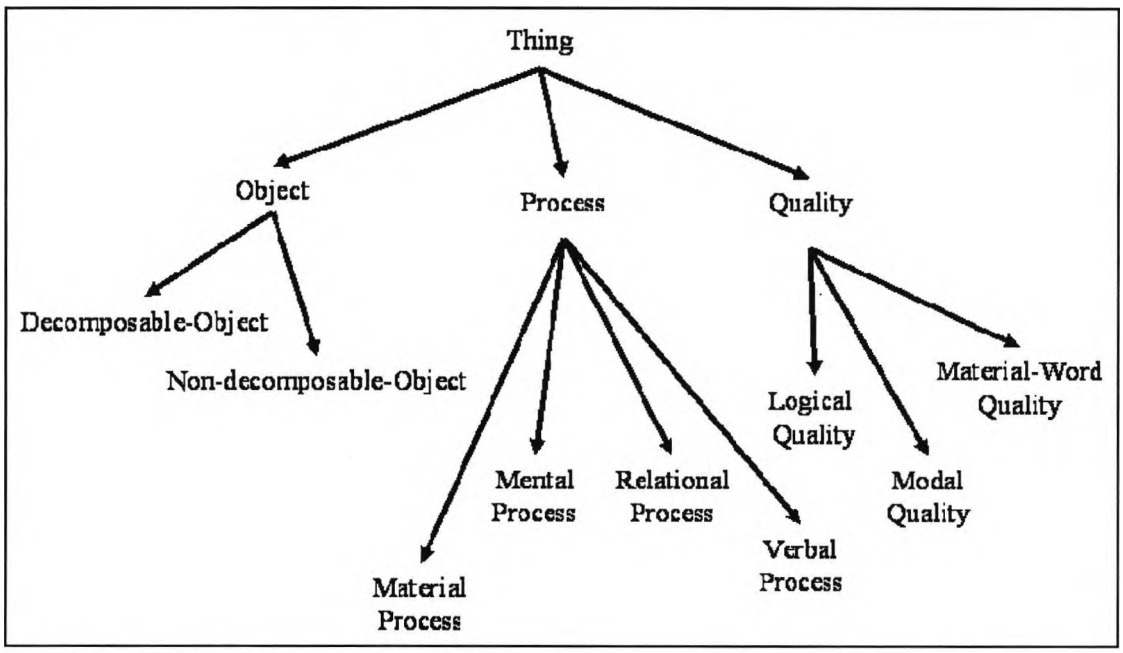


Figure 3.3 *The top level layer of PENMAN upper model*

The middle layer of the ontology is mostly WorldNet concepts merged with LDOCE and has been manually verified to ensure consistency. In the lower layer of the ontology Spanish terms from the Harper-Collins dictionary are linked to their WorldNet English counterparts. The ontology is represented formally in Loom, FrameKit, and Prolog.

3.2.1.3 WordNet

WordNet is a long-term research project started in 1985 by a group of psychologists and linguists at Princeton University (Miller, 1995). WordNet is defined as a machine-readable lexical database which organises words into five syntactic categories; these are nouns, verbs, adjectives, adverbs, and other function words. It contains over 118,000 word forms and 900,000 word senses, and semantic relationships such as synonyms (two forms have one meaning), hyponyms (is-a),

polysems (two meaning have one form), and meronyms (has-part). Synonyms in WordNet play a central role in the ontology and are organised in a hierarchy. Figure 3.4 shows the top level hierarchy of noun concepts.

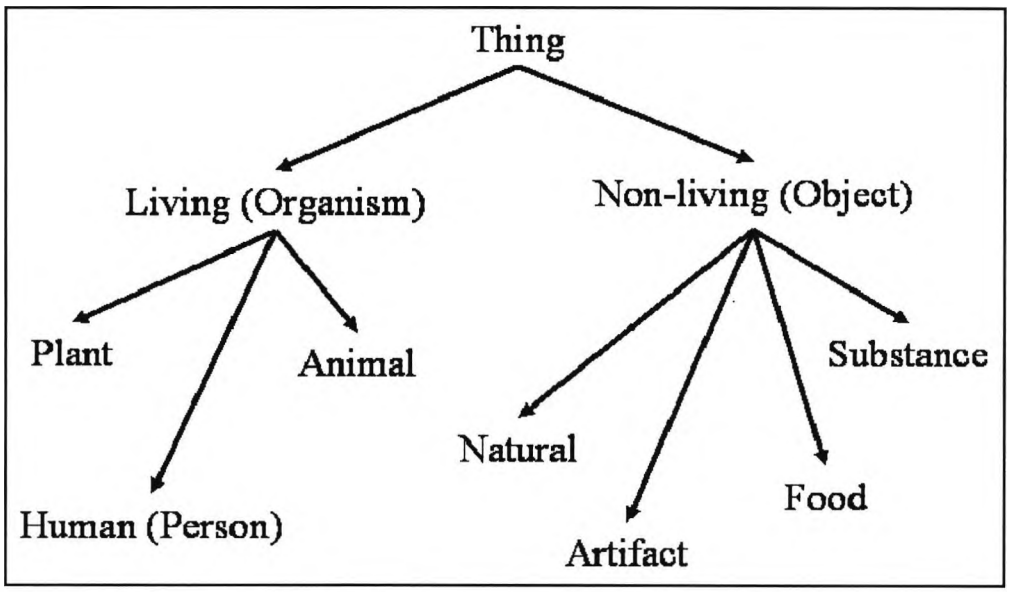


Figure 3.4 *WordNet top level hierarchy of noun concepts*

WordNet has been used in many research projects worldwide, being integrated into other ontologies such as Cyc and SENSUS. WordNet has been used for example in IR for semantic-based searching, where users' queries can be expanded to include other synonyms to gain broader coverage. In this work WordNet was investigated and the results are reported in a later section of this chapter. WordNet alone is not sufficient for the medical web since it lacks many of the medical terms. A recent study showed that only 2% of the medical concepts (from the UMLS health disorder class) are present in WordNet (Burgun & Bodenreider, 2001).

3.2.2 Domain –Specific Ontologies

General ontologies deal with general knowledge. Such ontologies lack the deep taxonomy found in many domains. Hence, there is a need for domain-specific ontologies to deal with this issue. In this section three medical-based ontologies are presented.

3.2.2.1 GALEN

GALEN, a UK-European joint project, has been developed in the late 1980s in the UK. The purpose of GALEN is to serve as a reusable terminology resource for clinical applications (Rector et al., 1994; 1995; 1997). The main model is called the Common Reference Model and is an ontology formalised in GRAIL (the GALEN Representation And Integration Language) which is based on description logic.

3.2.2.2 SNOMED-CT

SNOMED-CT (Systematised Nomenclature of Medicine - Clinical Terms), a collaborative project developed by the College of American Pathologists and the UK National Health Service (NHS), started in late 1990s and was first released in late 2001. SNOMED-CT has been formed by the convergence of SNOMED-RT and the UK's Clinical Terms Version 3 (formerly known as the Read Codes). The aim is to help clinicians to record and communicate patient records in standard form. SNOMED-CT is made up of three tables:

1. Clinical Concepts with around 340,000 concepts; each concept has a unique identifier and a fully specified name, also a cross reference with Read Code.
2. Description of concepts which has around 910,000 descriptions; each description has a unique identifier and a term that represents the clinical concept.
3. Relationships between concepts; currently it contains around 1.4 million semantic definitions.

SNOMED-CT is being used all over the world, and in the year 2004 it has been integrated within the UMLS.

3.2.2.3 UMLS

The United States National Library of Medicine's Unified Medical Language System project (UMLS) has developed a metathesaurus as an extensive cross referencing of a large number of existing coding systems. In the year 2003, the metathesaurus contained 875,255 concepts and 2.14 million concept names from 100 sources. Terms in the metathesaurus are organised by concepts. Hence, terms that hold the same meaning are linked together by a unique concept identifier (CUI). Every term is made up of several strings where each variation (i.e. upper case, lower case, singular, and plural) is one string and has a unique string identifier (SUI). However, all variations are listed under the same term and also have a unique term identifier (LUI). Relationships between concepts are also included in the metathesaurus; there are eleven types of relationships as shown in Table 3.1. CUIs are used to link relationships between different concepts.

No.	Relation Name	Description
1	RB	broader relationship
2	RN	narrower relationship
3	RO	other relationship than barrower, broader, or synonymous.
4	RL	like relationship
5	RQ	unspecified source asserted relationship
6	SY	source asserted synonymy
7	PAR	has parent relationship
8	CHD	has child relationship
9	SIB	has sibling relationship
10	AQ	allowed qualifier
11	QB	qualified by a concept

Table 3.1 *Relationships between concepts in the UMLS*

The UMLS also has a semantic network which classifies each concept into one of the categories in its upper-level ontology (Bodenreider & McCray, 2003). In the

2003 edition of the UMLS semantic network there are 135 semantic types (represented as nodes) and 54 relationships (represented as links between nodes). The primary relationship between the concepts is the "is-a" relationship; the other relationships are divided into five categories, and these are: "physically related to", "spatially related to", "temporally related to", "functionally related to", and "conceptually related to". The "is-a" relationship presents the hierarchy part in the UMLS semantic network. All concepts in the semantic net are divided under two headings; these are Entity and Event. Figure 3.5 shows the top layer of the Entity hierarchy.

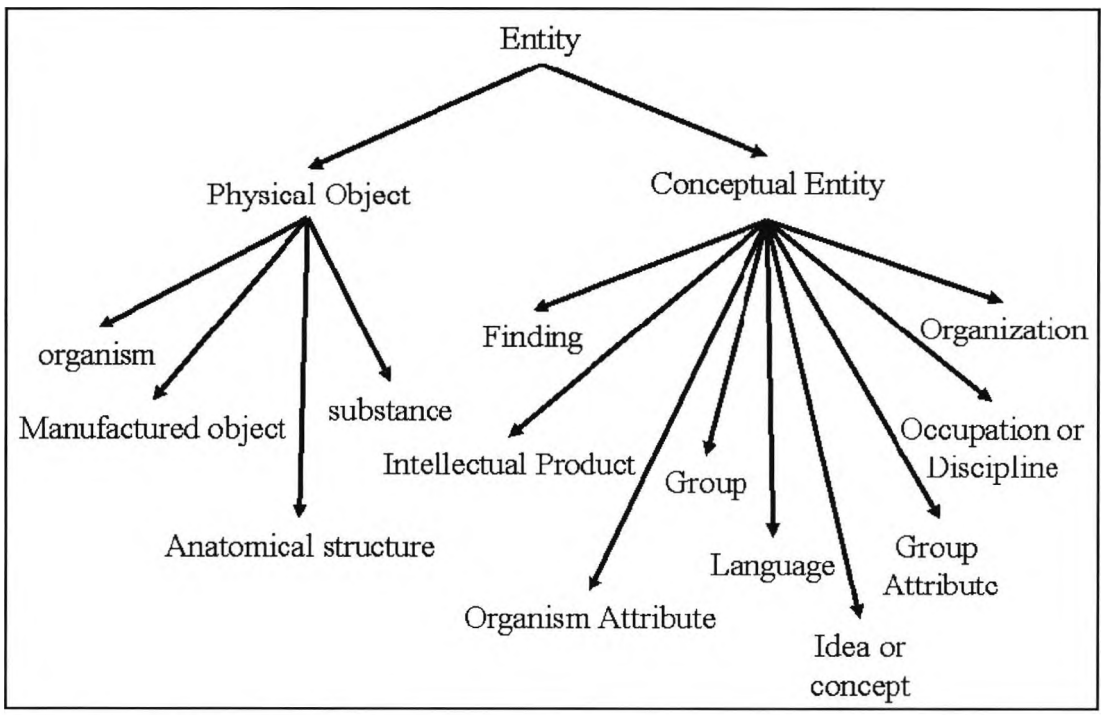


Figure 3.5 *UMLS upper level hierarchy of concepts*

The third component of the UMLS is the SPECIALIST lexicon and lexical programs which are aimed at processing English biomedical text. The SPECIALIST lexicon is an English language lexicon containing many biomedical terms along with their syntactic, morphological, and orthographic information (McCray et al., 1994). Besides the SPECIALIST lexicon, there are several lexical programs written in Java, which are used mainly for word normalisation, word index generation, and lexical variant generation.

The SPECIALIST lexicon and lexical program have many uses as specified in the UMLS literature such as to interpret and refine user queries, to map the user's terms to appropriate controlled vocabularies and classification schemes, to interpret Natural Language, and to assist in structured data creation.

The UMLS knowledge sources can be accessed through an online server using a graphical interface, a command line interface, or an Application Programming Interface (API).

3.3 Ontology Design

From the time of Greek Philosophers until recently, ontologies were merely a set of categories of entities that exist in our world. There were no systematic methodologies existing for the creation of ontologies, and it was only based on the contemplation of the philosophers. The utilisation of ontologies in many disciplines made researchers realise the need for well-defined principles and methodologies (Gruber, 1993; Guarino & Giaretta, 1995; Guarino, 1998b; Uschold & Gruninger, 1996; Fridman-Noy & Hafner, 1997). The basic methodology of designing ontologies can be defined as follows:

1. Determine the purpose and the scope of the ontology; what it is intended to be used for.
2. Capture the representative terms (classes) that exist in the domain of discourse to be used in the ontology.
3. Provide definitions for these classes. Other information can be added such as part-of-speech (depending on the purpose of the ontology).
4. Organise these classes in hierarchies (similar to object-oriented paradigms) by defining the relationships between these classes, for example generalisation-specialisation relations.
5. Transform the ontology into a formal model using one of the knowledge representation languages.

6. Provide evaluation and documentation.

Once the purpose and scope of the ontology has been set, the major step in the design is to determine what terms are representative classes and how to go about collecting these terms. Steps 2 to 4, the knowledge acquisition phase, can be repeated iteratively as much as needed for refining the ontology.
There are three different approaches in collecting and organising terms (Fridman-Noy & Hafner, 1997); these are as follows:

1. *Top-down approach*: start from the root-term and organise the terms down the hierarchy from the most general to more specific.
2. *Bottom-up approach*: start from the lower part and go up the hierarchy by organising the terms from specialised terms into more general terms.
3. *Middle-out approach*: start from somewhere in the middle and move out to eventually cover both up and down sides.

These methods and approaches are revisited in section 3.7 when discussing the ontology for the domain of diabetes management.

There are also several issues to consider when building ontologies, these are:

1. *Competency questions*: These questions can help to assure that the purpose and scope of the ontology have been met.
2. *Ontology reuse*: It is well worth it after setting the purpose and scope of the ontology to see if such ontology exists and if it can be reused which can save considerable time.
3. *Ontology integration*: It has been observed that many existing ontologies are the product of ontology integration; one particular example is the SENSUS ontology; it has been built by integrating other ontologies such as the PENMAN upper model, ONTOS ontology, WorldNet, and other ontologies. This can be most useful when integrating the newly designed domain-specific ontology with one of the existing general ontologies.

4. *Automatic knowledge acquisition*: This approach can save time in the process of acquiring the terms to be used in the ontology. WordNet used a dictionary as the basis for acquiring the terms to be entered in the knowledge base. Others used text corpus to extract terms that are relevant to the domain of discourse. Probably the major challenge in this approach is to separate relevant terms from irrelevant terms especially if dealing with a large corpus.
5. *Sub-ontology development*: The process of organising the ontology into sub-ontologies where each sub-ontology capture related concepts. This approach has been used in many projects; Cyc divided the ontology into groups called "Microtheories"; each one is related to a sub-domain, and WordNet divided the terms into five major groups such as nouns, verbs, etc.

Gruber defines a set of principles to follow in the design of ontologies; these are:

1. *Clarity*: Provide meaning for the defined terms with objectivity and lack of ambiguity. Also provide documentation using natural language and clear examples.
2. *Coherence*: The ontology should be consistent internally and on the natural documentation and examples level.
3. *Extendibility*: The ontology should be extendable for special uses. Providing new terms should not require making changes to the existing vocabulary.
4. *Minimal encoding bias*: The conceptualisation should be made at the knowledge level without depending on any particular symbol-level encoding. This would enable knowledge sharing among agents that are implemented with different knowledge representation systems and style of representation.
5. *Minimal ontological commitment*: The ontology should make as few claims as possible about the world being modelled, allowing the parties committed to the ontology to specialise and instantiate the ontology as required.

3.4 Ontology Languages

All ontologies that have been discussed in the previous section are expressed in some formal knowledge representation (KR) languages. For example, Cyc KB is represented in CycL language which has been derived from first order predicate calculus. SENSUS uses Loom, FrameKit, and Prolog. The GALEN ontology is formalised in the GRAIL language which is based on description logic. UMLS uses semantic networks to represent the concepts and their relationships. In general, ontology formal languages can be categorised into two types; these are logic-based and graph-based.

Logic was developed in the fifth century B.C. by Greek philosophers to represent and reason about knowledge (Sowa, 2000). Many types and forms of logic have been developed since then; some of these are predicate logic, propositional logic, modal logic, temporal logic, fuzzy logic, and finally description logic. Graph-based KR languages have been heavily researched during the seventies and eighties by the AI community.

The most well known graph-based systems are semantic networks, conceptual graphs, and the Unified Model Language (UML) which is used to model object-oriented systems.

It is important at this stage to see how these languages relate to knowledge representation on the semantic web, in particular the Web Ontology Language (OWL). First of all, OWL uses XML/RDF-based syntax for compatibility and portability. This means that systems designed to work with XML/RDF syntax should be able to parse OWL statements. Second, OWL is based on description logic (DL) which provides formal semantics and reasoning support. DL languages are descendents of semantic networks, frame-based systems, and the KL-ONE family. Constructs in DL have been mapped to OWL which allows the construction of complex concepts and roles from atomic ones. Concepts are equivalent to classes which are used to classify the entities in the domain of discourse. They are arranged

in an inheritance hierarchy. Roles are equivalent to properties. Objects are called individuals in DL which are instantiations of the concepts. It should be noted here that there are two types of concepts (classes); these are abstract concepts and concrete concepts. Abstract concepts cannot be instantiated; for example the term diabetes cannot have direct instances. On the other hand insulin is an instance of the concept hormone. The distinction between classes and instances is not easily perceptible. Many computer programming languages have been implemented based on DL such as Loom, Classic, OIL, DAML+OIL, and OWL. The main purpose of these languages is to describe the domain of discourse in a formal way which can be processed by machines.

3.5 Ontology Development Environments

Knowledge engineers have realised the need for tools that help in the construction of sharable and reusable ontologies. Several programs have been developed to facilitate the collection and organisation of domain-specific knowledge into manageable pieces. Several tools have been investigated and compared during the course of this work. These tools have many features in common since they all share the same goal which is building manageable ontologies. Some tools provide extra features while others lack some important ones. In the following, some of these features are presented:

1. *Graphical User interface*: Most ontology tools present forms which allow users to enter information about classes, slots, facets, and other information.
2. *Application Programming Interface (API)*: Some of these tools provide users (programmers) with a set of APIs for integration with their own applications.
3. *Import and Export Format*: Allow users to import and export their ontologies from and to other formats. The most common formats are XML and RDF(s).
4. *Extensibility*: Some tools allow users to develop and add new features that can be used from within the GUI.
5. *Merging*: Very few tools provide facilities for merging ontologies.

6. *Model architecture*: Some of the most common models that are used when building these tools are OKBC, DL, Prolog, and F-Logic.
7. *Purpose*: Some tools are tailored to a specific domain, but most of them can be used with any domain and considered to be general.

Many of these tools have shared powerful facilities such as ontology visualising, consistency checking and reasoning. Three tools have been investigated and are presented in the following sections; these tools are OilEd, OntoEdit, and Protégé-2000.

3.5.1 OilEd

OilEd (Bechhofer et al., 2001) has been developed by the Information Management Group at the University of Manchester in the UK. It is a small and simple tool based on Description Logic (DL) that can be used for building small scale DAML+OIL ontologies. It misses some of the features that are found in full scale ontology engineering environments such as collaboration, merging and integration. However, it is bundled with the FaCT reasoner that can be used to check consistency and classify ontologies.

The main (classes) screen is shown in Figure 3.6. Classes are listed inside the left panel. Once a class is highlighted its details are shown in the right-side panels. There are other tabs for entering and editing Properties, Individuals, Axioms, Containers, Namespaces, and Imports.

OilEd can import/export from and to DAML+OIL, OWL RDF(S), and OIL. OilEd is considered a general purpose ontology editor and is not restricted to any specific domain. OilEd is an open-source tool implemented in Java.

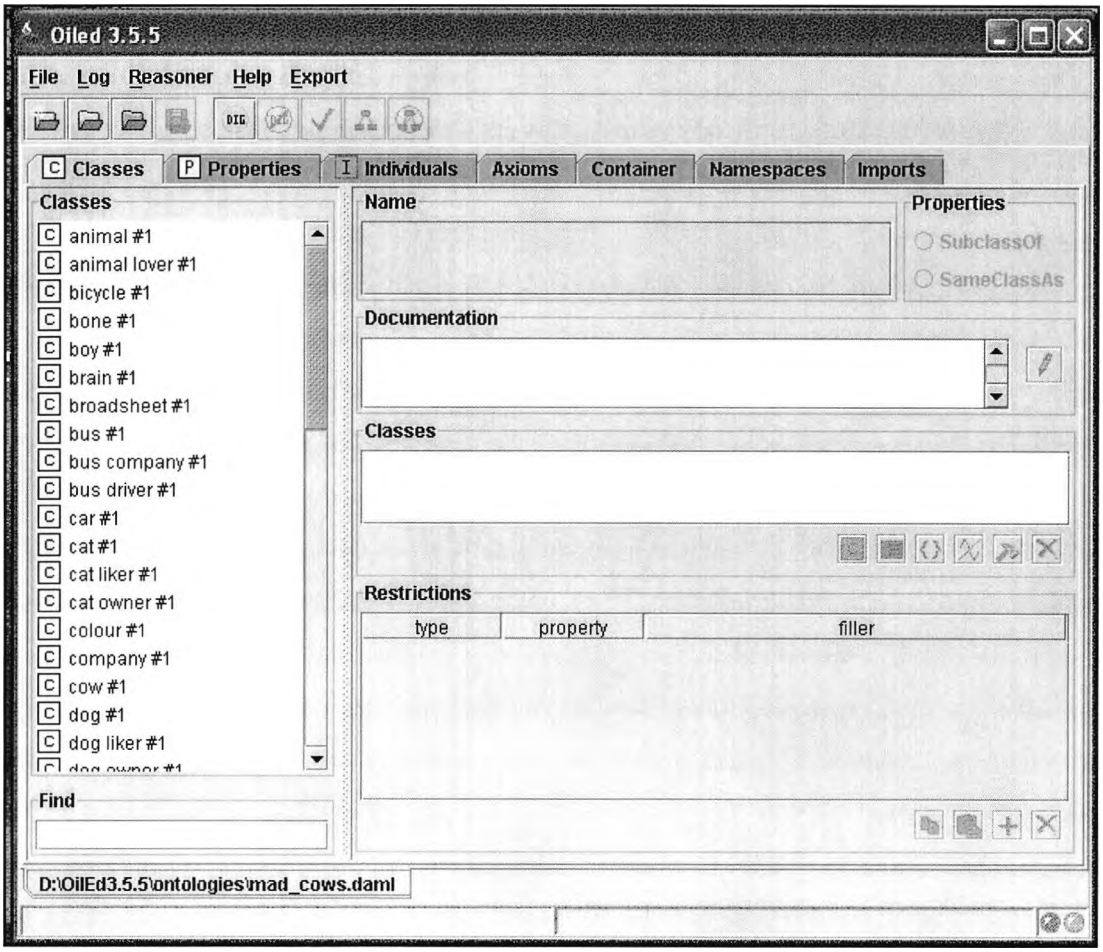


Figure 3.6 *Screenshot of the classes window of the OilEd tool*

3.5.2 OntoEdit

OntoEdit (Sure et al., 2002a; 2002b) is based on frames and First-Order-Logic (FOL). It is a multilingual ontology engineering environment developed by AIFB, University of Karlsruhe. There are three versions of this tool; the freeware version (called OntoEdit Free) which provides basic functionality that includes import/ export from and to XML, RDF(S), F-Logic, and DAML+OIL. Other features are multilingualism and multiple ontology management. It has several external plug-ins such as the visualisation plug-in. The free version limits the number of concepts, relations, and instances that can be constructed to 50 each.

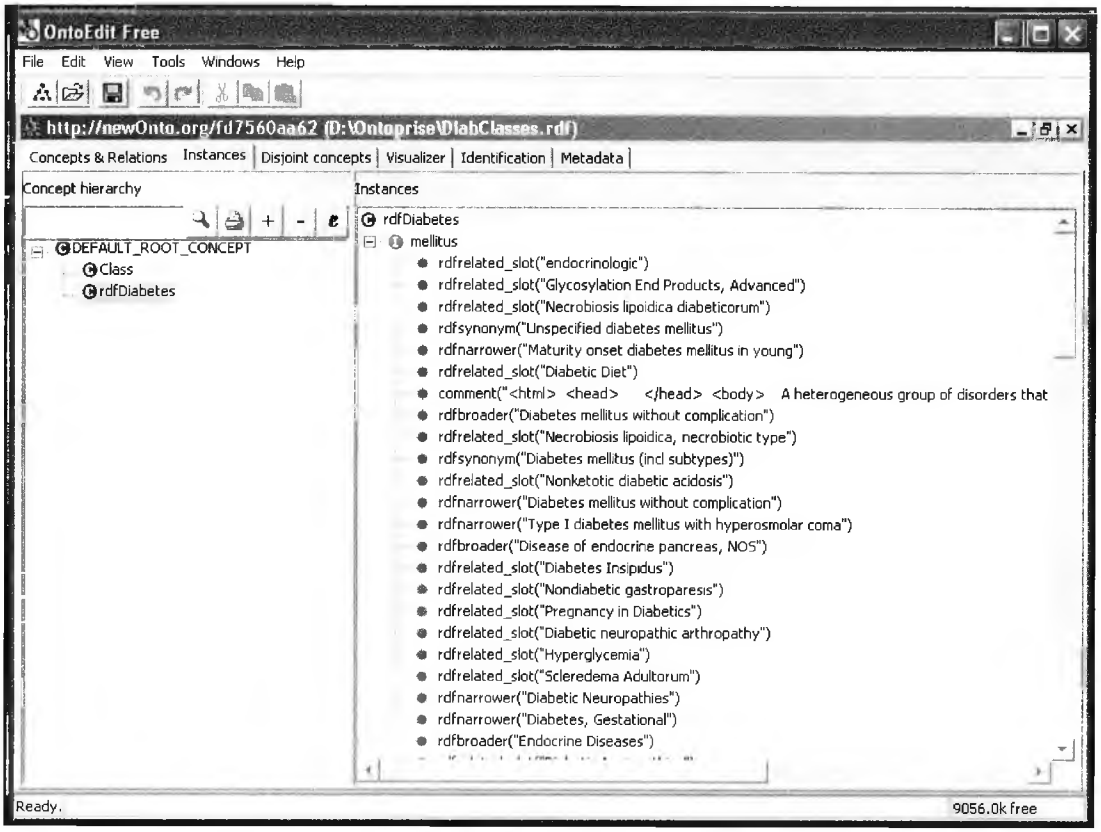


Figure 3.7 *Screenshot showing the OntoEdit tool*

The commercial versions are OntoEdit and OntoEdit professional; they add several features in the form of plug-ins such as textual and graphical rule editors, inference engine, and query tool. Also, they allow collaborative construction and sharing of ontologies. Figure 3.7 shows a screenshot of the free version. The left-side panel shows the class hierarchy and when a class is selected the right-side panel shows the instances under the highlighted class.

3.5.3 Protégé-2000

Protégé-2000 (Fridman-Noy et al., 2000; 2001; Knublauch et al., 2004; Fridman-Noy et al., 2006) is being developed by the medical informatics group at Stanford University and is one of the most popular ontology development environments with thousands of users worldwide. It is an open-source project written in Java.

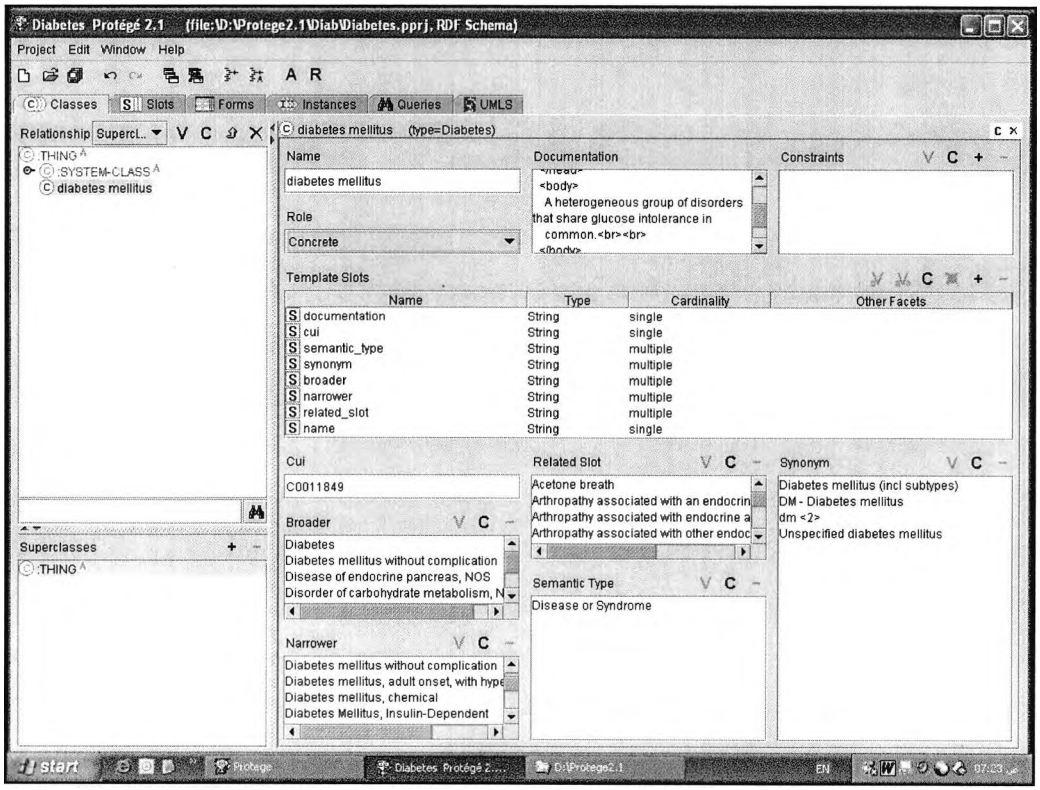


Figure 3.8 *Screenshot showing the Protégé-2000 tool*

Protégé-2000 is based on FOL and frames, can import from XML(S) and RDF(S), and export to XML(S), RDF(S), CLIPS, FLOGIC, Java, and HTML. Like OilEd and OntoEdit it has an interactive and simplified GUI for constructing and managing ontologies. Looking at Figure 3.8 the left panel presents the class hierarchy and on the right is a custom-tailored template for entering class information.

Protégé-2000 has gained its popularity due to its flexibility and extensibility features. It allows users to create three types of plug-ins:

1. *Tab plug-ins*: These are KB applications that use the Protégé-2000 API components and add extra functionality such as visualisation, and inference. There are many tabs that have been developed by Protégé-2000 users and are available on the website[2] for downloading. Some of the most popular ones are PAL (Protégé Axiom Language) tab, OntoViZ tab, Jambalaya tab, UMLS tab, WordNet tab, and many other tabs. Figure 3.9 displays a

[2] http://protege.stanford.edu/

screenshot for the UMLS tab in-action. The UMLS tab allows the user to online-search the UMLS KB server for medical terms.

2. *Slot Widgets*: Allow users to build customised and domain-specific input screens for editing slot values.
3. *Back-ends*: Allow users to store their ontologies in customised format. This is done by mapping between Protégé-2000 in-memory KB representation and the file output in the required format.

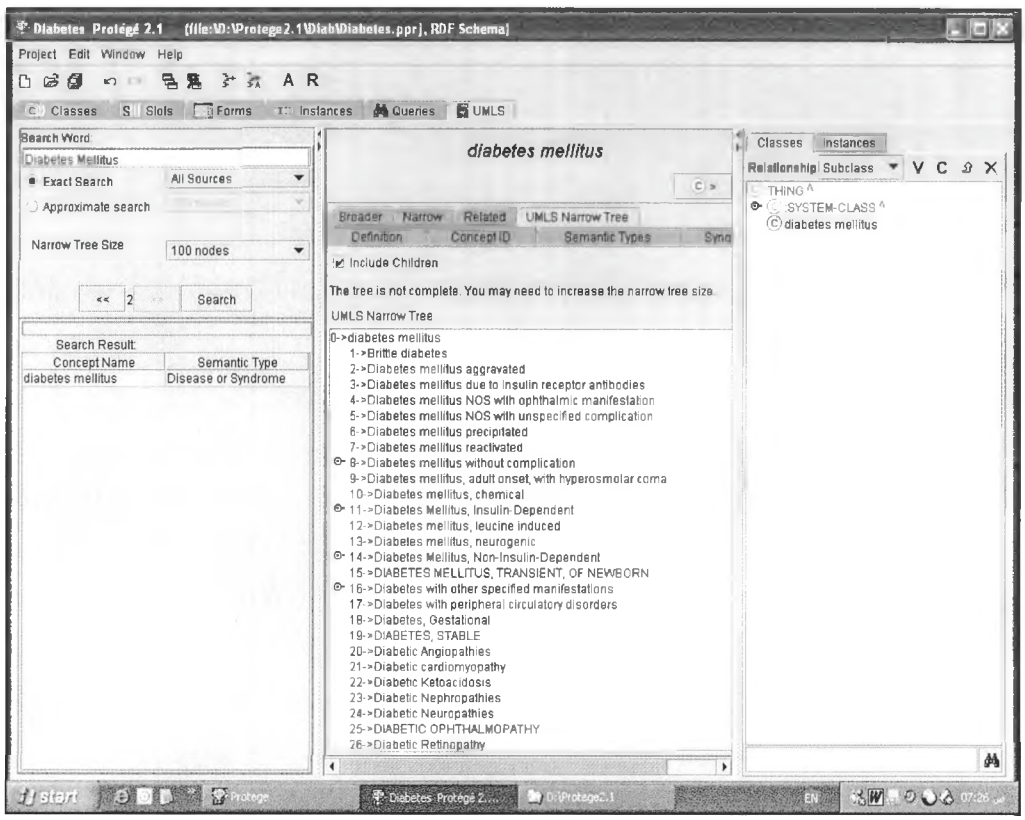


Figure 3.9 *Screenshot showing the UMLS tab*

3.6 Ontology for Information Extraction

As stated in chapter 1, the aim of this work is the design of a framework for automating the construction of a semantic layer for a given webpage with the guide of ontology. The semantic layer should give a fair description about the webpage and its content.

The exact purpose of the ontology for extracting information from a giving webpage has been determined as follows:

1. The ontology will show that different extracted terms have the same sense (Example: Type-1 and Juvenile are both refer to insulin-dependent diabetes). This issue is related to vocabulary control and the assignment of subject classification codes, which are used to control which term to use. In the example above, one code should refer to both terms.
2. An extracted term may have more than one sense (Example: blood could mean the red fluid that pumped by the heart or the descendants of one individual). The ontology will help in choosing the correct one. Researchers report that investigating surrounding words helps to determine the correct meaning.
3. The relation between two terms (Example: insulin is-a hormone)
4. Classification of terms is shown from the hierarchy of domain-specific terms. (Example: Class Diabetes is a generic concept, while Type-1 is a particular occurrence of the generic concept).
5. The categorisation of web pages based on a predefined set of categories saved in the ontology. More importantly is the process of finding relevant sentences and blocks of text in a webpage, possibly by filtering out irrelevant sentences .

Careful examination of the above points concludes the need for two ontologies, as follows:

1. Terminology-based ontology: This is a well-developed ontology which contains a metathesaurus that covers most and probably all the terms that may occur in the domain of discourse. Such ontology can be used to cover the first four points above.

2. Application ontology: Is more specific to the application at hand. In this work such ontology covers basic concepts found on chronic disease related websites (the common ones) along with their attributes.

Ontologies that cover the first point above have been researched for this project with two strong candidates that were selected. WordNet and UMLS have been chosen for representing the first one. Some experiments have been conducted in the process to determine which one is more suitable. These studies are discussed in the following section. The second type is discussed in detail in chapter 6.

3.7 UMLS vs. WordNet: Empirical Evaluation

This section presents a study that was conducted in order to determine which ontology is more suitable for this work. This takes into consideration the need for ontology that is general, yet provides domain-specific terminology (i.e. the domain of diabetes management).

In order to capture representative terms, a set of 80 terms has been extracted from a collection of diabetes-related websites. These terms are categorised into two types:

1. Directly-related terms (i.e. hyperglycaemia, A1C).
2. Indirectly-related terms (i.e. autoimmune, obesity).

The semantic types of these terms were retrieved from WordNet and UMLS and shown in Table 3.2. The UMLS semantic types were abbreviated; Table 3.3 displays the full name of each abbreviation. The study has shown that approximately 50% of the directly-related terms and 25% of the indirectly-related terms listed in Table 3.2 are not found in WordNet[3], but all 80 terms are found in UMLS.

[3] Approximately 70% of these terms that are not found in WordNet are pharmacological substances.

Directly-related			Indirectly-related [4]		
Term	UMLS	WordNet	Term	UMLS	WordNet
A1C	LP	NF [5]	acanthosis nigricans	DS	disorder
acarbose	C / PS	NF	ACE inhibitor	PS	drug
acetohexamide	OC / PS	NF	albuminuria	LR / DS	symptom
amylin	APP / H	NF	amyotrophy	DS	symptom
aspart insulin	APP / PS	NF	angiopathy	DS	pathology
aspartame	APP	sweetener	autoimmune	PF	body function
biguanide	OC	NF	bunion	AA	symptom
cheiroarthropathy	DS	NF	callus	AA / DS	symptom
c-peptide	APP / BAS	NF	charcot's foot	DS	NF
dehydration	DS	NF	creatinine	OC / BAS	NF
dextrose	C / BAS	glucose	dermopathy	DS	NF
DiaBeta	OC / PS	drug	dieting	TP	treatment
dymelor	OC / PS	NF	erectile dysfunction	DS	disorder
excessive urination	SS	symptom	gangrene	DS	pathology
tolazamide	OC / PS	drug	gland	BOO	organ
fructosamine test	TP	NF	hemodialysis	TP	chem. reaction
fructose	C / PS	sugar	hyperlipidemia	DS	symptom
glimepiride	OC / PS	NF	hypertension	DS / SS	disorder
glucagon	OC	hormone	hypotension	PF	disorder
glucose	C / BAS	sugar	immunosuppressant	PS	drug
glyburide	OC / PS	drug	impotence	DS	disorder
sulfonylurea	OC / PS	drug	Injection	HCA	treatment
glycosuria	DS / SS	symptom	ketone	OC	organic compound
Hyperglycaemia	DS	symptom	lancet	MD	medical device
hyperosmolar	DS	symptom	lipid	CD	organic compound
insulin pump	MD	NF	lipoatrophy	DS	NF
insulinoma	NP	NF	macular edema	DS	NF
ketoacidosis	DS	pathology	metabolism	OF	organic process
ketonuria	SS	symptom	mg/dL	QC	NF
ketosis	F / DS	symptom	mmol/L	QC	NF
meglitinide	OC / PS	NF	monofilament	MD	NF
metformin	OC / PS	NF	mononeuropathy	DS	pathology
micronase	OC / PS	drug	obesity	SS	condition
miglitol	OC / PS	NF	protein	OC / PS	organic compound
nateglinide	OC / PS	NF	proteinuria	LR / DS	symptom
pioglitazone	OC / PS	NF	retina macula	BOO	area
polydipsia	SS	symptom	retinopathy	DS	NF
polyphagia	BD / SS	symptom	risk factor	QC	NF
polyuria	SS	symptom	syringe	MD	medical device
tolbutamide	OC / PS	drug	Urine	BS	body Waste

Table 3.2 *List of directly-related and indirectly-related terms and their semantic types*

[4] Laymen may not tell from reading the definitions whether these terms are diabetes-related or not.
[5] NF: Not Found.

The list of 23 UMLS concepts that the 80 diabetes-related terms belong to are noted along with their concept group (Table 3.3). The UMLS semantic network reduces the complexity of the Metathesaurus by grouping concepts into semantic groups. The 15 semantic groups provide a partition of the UMLS Metathesaurus for 99.5% of the concepts (Bodenreider & McCray, 2003).

No	Abbreviation	UMLS Concept	Concept Group
1	AA	Acquired Abnormality	Disorders
2	APP	Amino Acid, Peptide, or Protein	Chemicals & Drugs
3	BAS	Biologically Active Substance	Chemicals & Drugs
4	BD	Mental or Behavioural Dysfunction	Disorders
5	BOO	Body part, Organ, Organ Compound	Anatomy
6	BS	Body Substance	Anatomy
7	C	Carbohydrate	Chemicals & Drugs
8	DP	Diagnostic Procedure	Procedures
9	DS	Disease or Syndrome	Disorders
10	F	Finding	Disorders
11	H	Hormone	Chemicals & Drugs
12	HCA	Health Care Activity	Procedures
13	LP	Laboratory Procedures	Procedures
14	LR	Laboratory or Test Results	Phenomena
15	MD	Medical Device	Devices
16	NP	Neoplastic Process	Disorders
17	OC	Organic Chemical	Chemicals & Drugs
18	OF	Organism Function	Physiology
19	PF	Pathologic Function	Disorders
20	PS	Pharmacologic Substance	Chemicals & Drugs
21	QC	Quantitive Concept	Concepts & Ideas
22	SS	Sign or Symptom	Disorders
23	TP	Therapeutic or Preventive Procedure	Procedures

Table 3.3 *List of UMLS concepts used in Table 3.4*

In the final step, each WordNet concept is mapped into the equivalent UMLS concept. This is to insure that the selected WordNet concept does have equivalent concepts in UMLS. Table 3.4 shows this mapping along with their concept group.

No	WordNet Concept	UMLS Concept	Concept Group
1	Area	Body part, Organ, Organ Compound	Anatomy
2	Body function	Pathologic Function	Disorders
3	Body Waste	Body Substance	Anatomy
4	Chemical Reaction	Therapeutic or Preventive Procedure	Procedures
5	Condition	Sign or Syndrome	Disorders
6	Disorder	Disease or Syndrome	Disorders
7	Drug	Organic Chemical / Pharmacological Substance	Chemicals & Drugs
8	Glucose	Carbohydrate	Chemicals & Drugs
9	Hormone	Organic Chemical	Chemicals & Drugs
10	Medical device	Medical Device	Devices
11	Organ	Body part, Organ, Organ Compound	Anatomy
12	Organic compound	Organic Chemical	Chemicals & Drugs
13	Organic process	Organism Function	Physiology
14	Pathology	Disease or Syndrome	Disorders
15	Sugar	Carbohydrate	Chemicals & Drugs
16	Sweetener	Amino Acid, Peptide, or Protein	Chemicals & Drugs
17	Symptom	Disease or Syndrome[6]	Disorders
18	Treatment	Therapeutic or Preventive Procedure	Procedures

Table 3.4 *Mapping the extracted WordNet concepts into UMLS concepts*

3.8 Summary of Research Issues

Ontologies are becoming one of the most important research topics in many domains. They provide very efficient ways for the organisation and management of domain-specific knowledge which can be shared and reused. Research in ontology has focused on many issues such as design methodologies, sharing and reuse, merging, and ontology evaluation. Many concepts from the software engineering field have been borrowed to aid the process of constructing ontologies, and the name ontology engineering is becoming a common term. Recently, the discipline has focused on standardisation for ontology integration and reuse. Many languages have been developed for representing ontologies formally and many ontology development environments have been produced to aid the construction of ontologies. However, there are only a few studies on the evaluation and comparisons of such tools.

[6] Other UMLS concepts can be mapped into symptom, but this one was chosen because it has larger occurrences in Table 3.3.

The most felt burden for non IT specialists is the manual construction of their own ontologies which is a very time-consuming process and a costly one especially in the construction of large and complex ontologies. Recently, several research activities have been undertaken aiming to semi-automate the process of ontology construction from domain-specific corpora through machine learning approaches (Maedche & Staab, 2001), though such work is still in its early stages.

In this work, UMLS and WordNet have been compared to see which one is more suitable for this project. In the study, 80 terms from diabetes-related websites have been extracted and used in the evaluation. The study has shown that UMLS provides a better choice for this project. The details of how UMLS is utilised are explained in chapter 6.

In the next chapter, information extraction will be discussed. Current approaches to information extraction, research challenges, and problems with the current IE technology will be reviewed, together with discussion of how information extraction can help towards the semantic web.

Chapter Four

Review: Information Extraction

"Traditional grammars do not describe the facts of language; rather, they provide hints to the reader who already has, somehow, the requisite 'notion of structure' and general conceptual resources, and can use the hints to determine the expressions of the language and what they mean."

- Chomsky, 1994

4.1 Introduction

The World Wide Web has become an important medium for the dissemination of information related to a wide range of topics. The majority of human information is becoming available on the web very rapidly. In the medical domain, the number of documents related to healthcare is already large and continues to grow at an exponential rate. Online information sources may have relieved health practitioners from the manual process of locating their information from library bookshelves, but have created other problems such as information overload, also dealing with irrelevant and redundant information presented by search engines. Most of today's search engines use statistical techniques over a bag of words to find the keywords that best match the user's query. During the 90s, research in information retrieval (IR) focused on methods for filtering, grouping, organising, and presenting search results to the user. Despite all the advances in the field of IR many of the problems presented by search engines still exist today. The main reason for this is that today's search engines are busy crawling and processing HTML documents that are made to be read and understood by humans and not machines. Once the new layers of the semantic web start to unfold and spread throughout the web then new smart search

agents can process the semantic web in very efficient ways. There are billions of un-semantic web pages in existence today. Annotating the current web with semantic tags manually is deemed impossible. Thus, automating the process becomes very important and currently there are very few research efforts that are moving in this direction.

Information extraction (IE) approaches can be used to make textual information found on the web available in a structured format. The main components of any IE system are the template to be filled by the system and a set of extraction rules that are used to extract relevant information for a particular extraction task (Muslea, 1999). IE is a relatively new field which has evolved during the last two decades due to two factors; the exponential growth of online text documents and the influence of the Message Understanding Conferences (MUC). MUC is an annual[7] competition event started in 1987 and organised by the Defence Advanced Research Projects Agency (DARPA). DARPA has sought the opportunity of extracting specific information from online textual sources. Each MUC conference is focused on different topic and extraction task. Participants are a mix from industrial and academic research labs designing IE systems based on the domain and specifications given by DARPA. IE systems are then evaluated using a domain-specific text corpora and an official scoring program. The first and second MUC were about extracting specific information from messages about naval operations. MUC-3 and MUC-4 focused on extracting specific information from news articles about terrorism (i.e. extracting terrorist names, targeted locations, dates, victims' names, etc.).

IE is different from IR. In IR relevant documents from a large collection of documents are retrieved, while in IE relevant information from document(s) are located and extracted which then are filled inside predefined templates. It seems that one is the opposite of the other. In IR a set of keywords is fed to the IR system to retrieve a set of representative documents, while in IE a set of documents is fed to

[7] Not exactly an annual event. The events took place in the years 1987, 1989, 1991, 1992, 1993, 1995, and 1998.

the IE system to extract the set of keywords (based on the extraction rules). Combining IE and IR provide powerful tools for text processing.

In this chapter current approaches to Information Extraction are reviewed. State-of-the-art IE systems are presented and discussed, then problems with the current approaches and research issues are presented, and finally a discussion about how Information Extraction can help towards the semantic web is presented.

4.2 Approaches to Information Extraction

Most web pages on the Internet are in the form of free text (i.e. unstructured format). Some web pages are semi-structured, which means that the important data in this type of web pages are stored inside tables and lists. Such tables have some degree of structure. As seen in Figure 4.1 each column in the table starts with a heading and each row resembles a record in a database.

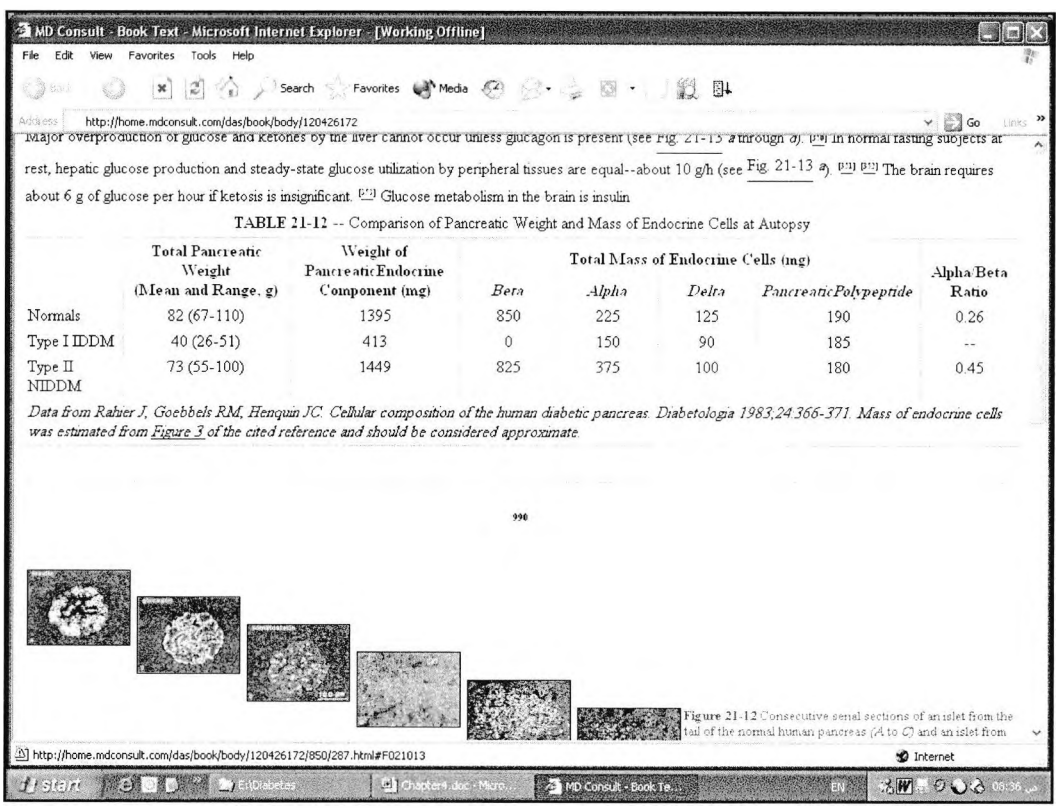


Figure 4.1 *Example of semi-structured data on the web*

Extracting information from semi-structured documents is somewhat easier than extracting information from free text. Figure 4.2 reveals part of the HTML text for the table shown in Figure 4.1. In particular, the important data are shown to be wrapped inside pairs of tags such as <I>...</I> and <TD>...</TD>.

```
<TABLE id=TO21012 cellSpacing=0 cellPadding=3 rules=cols width="100%"
align=center border=1 frame=hsides><A name=TO21012>
  <CAPTION align=top><A name=TO21012><B>TABLE 21-12</B> -- Comparison of
  Pancreatic Weight and Mass of Endocrine Cells at Autopsy</A></CAPTION>
  <THEAD vAlign=bottom>
  <TR>
    <TH vAlign=bottom align=middle width="10%" rowSpan=2><BR></TH>
    <TH vAlign=bottom align=middle width="17%" rowSpan=2><B>Total Pancreatic
      Weight<BR>(Mean and Range, g)</B></TH>
    <TH vAlign=bottom align=middle width="17%" rowSpan=2><B>Weight of
      PancreaticEndocrine Component (mg)</B></TH>
    <TH vAlign=bottom align=middle colSpan=4><B>Total Mass of Endocrine Cells
      (mg)</B></TH>
    <TH vAlign=bottom align=middle width="10%" rowSpan=2><B>Alpha/Beta
      Ratio</B></TH></TR>
  <TR>
    <TH vAlign=bottom align=middle width="10%"><B><I>Beta</I></B></TH>
    <TH vAlign=bottom align=middle width="10%"><B><I>Alpha</I></B></TH>
    <TH vAlign=bottom align=middle width="10%"><B><I>Delta</I></B></TH>
    <TH vAlign=bottom align=middle
      width="10%"><B><I>PancreaticPolypeptide</I></B></TH></TR></THEAD>
  <TBODY vAlign=top>
  <TR>
    <TD vAlign=top align=left width="10%">Normals</TD>
    <TD vAlign=top align=middle width="17%">82 (67-110)</TD>
    <TD vAlign=top align=middle width="17%">1395</TD>
    <TD vAlign=top align=middle width="10%">850</TD>
    <TD vAlign=top align=middle width="10%">225</TD>
    <TD vAlign=top align=middle width="10%">125</TD>
    <TD vAlign=top align=middle width="10%">190</TD>
    <TD vAlign=top align=middle width="10%">0.26</TD></TR>
  <TR>
    <TD vAlign=top align=left width="10%">Type I IDDM</TD>
    <TD vAlign=top align=middle width="17%">40 (26-51)</TD>
    <TD vAlign=top align=middle width="17%">413</TD>
    <TD vAlign=top align=middle width="10%">0</TD>
    <TD vAlign=top align=middle width="10%">150</TD>
    <TD vAlign=top align=middle width="10%">90</TD>
    <TD vAlign=top align=middle width="10%">185</TD>
```

Figure 4.2 *Part of the HTML source that is used to render the table in figure 4.1*

Some special techniques have been developed to deal with semi-structured data on the web; such techniques are called wrapper induction (Kushmerick et al., 1997; Ashish & Knoblock, 1997). Wrappers are designed to look for patterns located around the desired data. Once they are found the surrounded text gets extracted and cleaned. As an example, from figure 4.2 it can be observed that the elements inside the tables are surrounded with the HTML tags <TD> and </TD>, the first tag stands for "Table Data" and the second tag is for the end of table data. These resemble the

boundaries of a cell in a spreadsheet or a field in a database record. The data between the two tags is obviously the data needed to be extracted. The process is usually iterated to cover the whole table.

Information Extraction has emerged from several fields of computer science; these are natural language processing (NLP), computational linguistics, information retrieval, and machine learning. However, it started as an NLP task that is concerned with extracting relevant information from natural text. The extracted information is placed inside predefined templates.

Some of the tools used by computational linguists are being used in Information Extraction to parse and grammatically analyse the extracted pieces of text. These techniques have helped in many ways, for example: the recognition of named entities such as names of persons, locations, organisations, etc., and in finding relationships between entities through the recognition of verbs. These techniques are presented in more details in the next section. IE systems are evaluated using the same evaluation approaches found in IR, namely Precision (P), Recall (R), and F-measures (F), and F1-measures (F1) which are defined as follows:

$$P = \frac{\text{number of correctly predicted answers}}{\text{number of predicted answers}} \tag{4.1}$$

$$R = \frac{\text{number of correctly predicted answers}}{\text{number of true answers}} \tag{4.2}$$

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{4.3}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{4.4}$$

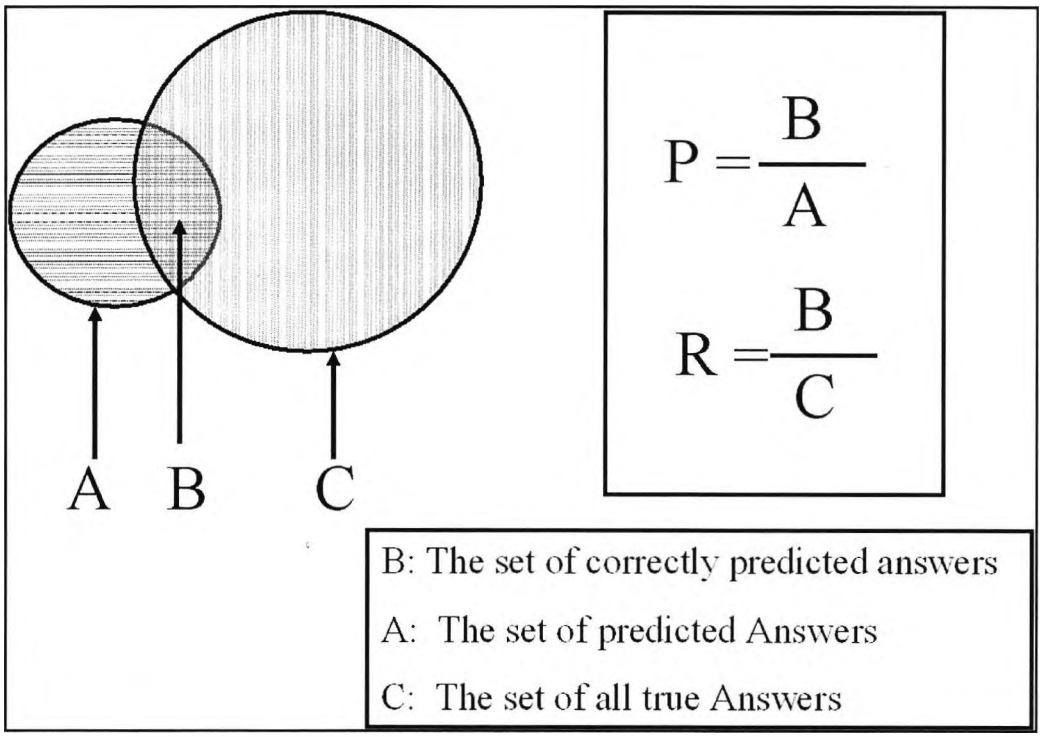


Figure 4.3 *Precision and Recall*

Figure 4.3 presents visualisation of precision P and recall R, assuming that A is the set of all answers that is returned by the IE system part of which is correctly predicted answers (shown here in B), and C is the set of all true answers that ideally should be returned by the system. Then P is the ratio of B over A, and R is the ratio of B over C. F-measure combines both P and R where β indicates a relative value of precision versus recall; it shows how much to favour R over P. F1 is a derivation which treats P and R equally (β = 1). F1 is used in the literature to report results and to compare the different schemes.

More recently IE has evolved from the manual construction of the extracting rules into the automatic generation of these rules by the use of Machine Learning paradigms. Both approaches for Information Extraction are discussed in the following sections.

4.2.1 The Knowledge Engineering Approach

The key element of IE systems is a set of extraction rules. These rules instruct the IE system as to what pieces of text to extract from a given document. The knowledge engineering approach is often called the manual rule-base approach because these rules are constructed manually from domain-specific knowledge. The following scenario presents a simple case.

Example:

Assume the need to extract people's names and their email addresses from web pages. An email expert declares that an email address consists of a name followed by "@" sign then followed by a host name then followed by a dot and finally followed by an extension which is normally made of two or three letters. The knowledge engineer writes extraction rule by constructing a regular expression for the email format as follows:

\b[A-Za-z]+@[A-Za-z]+\.[A-Za-z]{2,3}\b

This regular expression then becomes part of the IE system which is used to find patterns that match this expression. Once an email is located, the surrounding text can be searched for the person's name (or vice versa).

Note that in the above example the regular expression is too rigid and works with letters only. Some email addresses may contain digits and other characters (e.g. underscore) and may have a second extension at the end representing the country (e.g. ali123@go.com.jo), among other variations. The same remark applies when working with dates which usually have many different forms. Knowledge engineers will need to construct many extraction rules to cover all possible cases. The process of constructing extraction rules becomes complex very quickly and is a tedious job.

Early IE systems relied on expert knowledge and manually-constructed sets of rules (Appelt & Isreal, 1993; Soderland et al., 1995). These early systems were expensive to build, were time-consuming, and were very domain specific. IE systems were

developed using the knowledge engineering approach cannot be easily adapted to new domains because, as seen in the above example, each extraction rule is made for a very specific task.

Extracting people's names and their email addresses alone in the above example may not be very useful without other information. This is similar to dates which may need to extract events that occur on these dates along with other information such as people's names, locations, etc. In the case of domain-specific phrases, an IE system needs to recognise named entities that are domain-specific. These are strings that may contain any number of letters and digits and may have no specific patterns such as those found within emails and dates. To solve these problems, NLP methodologies were applied for processing natural text. NLP-based IE systems need to "kind-of" understand natural text. There are six levels of knowledge in understanding natural text (Jurafsky & Martin, 2000):

1. Phonology: the phoneme is the smallest unit of sound;
2. Morphology: word components like prefix and suffix and other word forms;
3. Syntax: grammatical structures of phrases and sentences;
4. Semantics: represent meanings;
5. Pragmatics: usage of different context;
6. World Knowledge: this is background knowledge.

An IE system performing full analysis on the natural text normally starts with tokenising the text and then applies morphological analysis by assigning part-of-speech (POS) and other information to each word such as singulars, plurals, nouns, verbs, prepositions.

The next stage is syntactic analysis where the task is to identify certain groups like noun phrases (NP), verb phrases (VP), and preposition phrases (PP). Figure 4.4 shows the parse tree resulted from parsing the sentence "The doctor injected the patient with insulin".

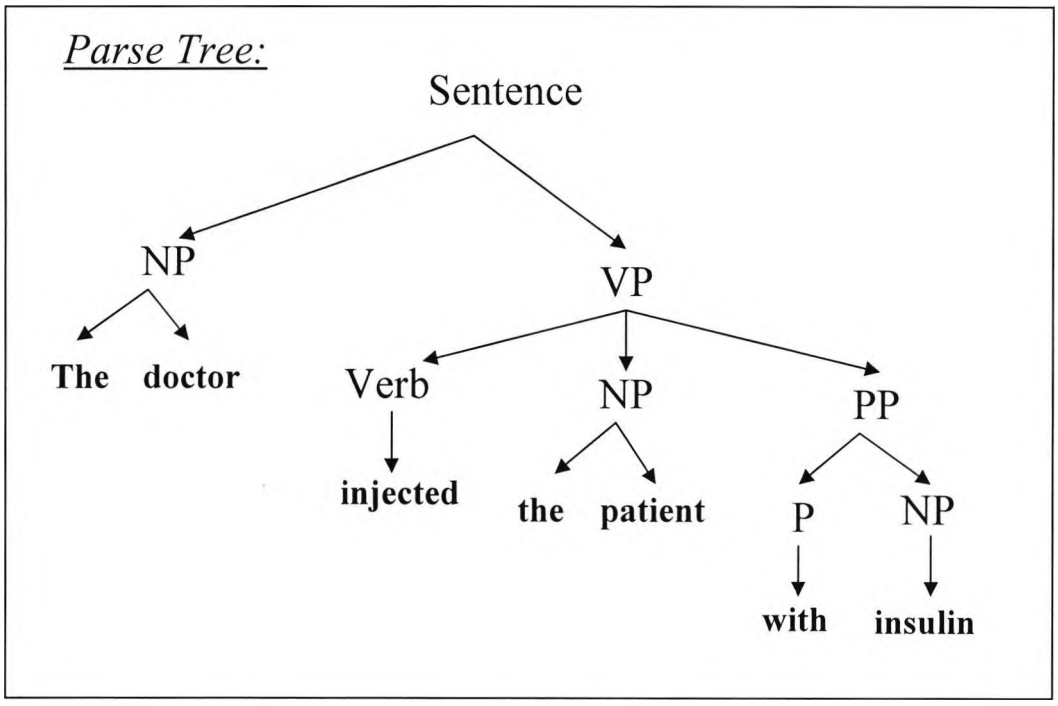


Figure 4.4 *Sentence parsing as part of the text syntactic analysis stage*

This is followed by semantic analysis where other word senses are extracted from special dictionaries and domain-specific Ontologies.

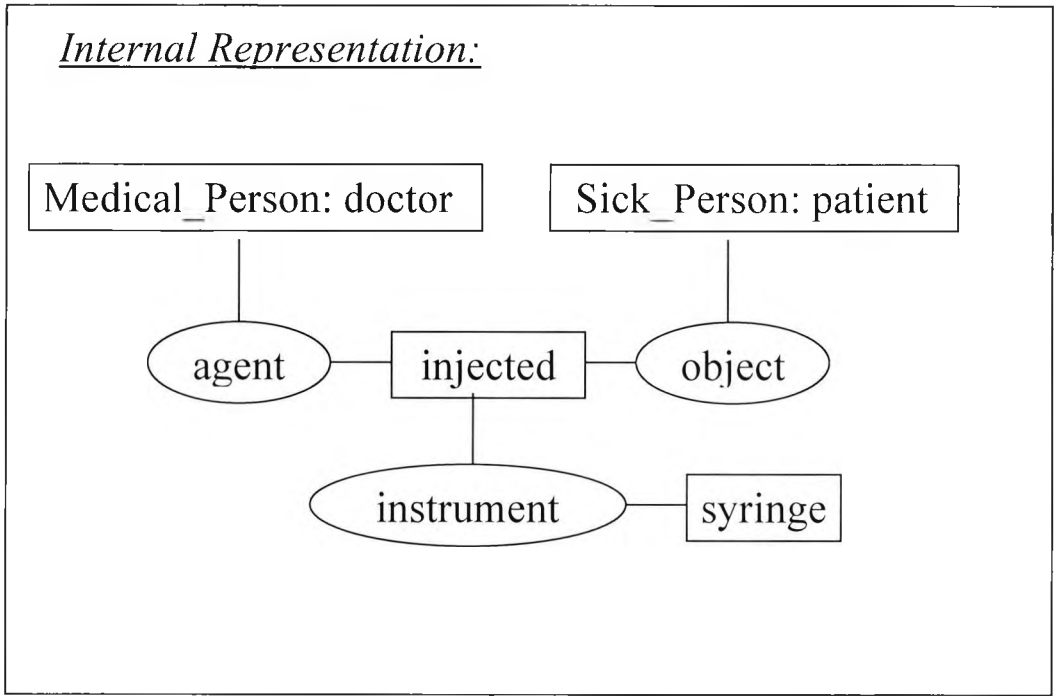


Figure 4.5 *Semantic Interpretation*

In this example, "doctor" is interpreted as "Medical_Person". Figure 4.5 shows one possible interpretation for the sentence "doctor injected patient". Conceptual graph is used to represent the knowledge formally.

The last stage, before extracting the targeted fields, is to expand the representation to see how this text fits within world knowledge. Figure 4.6 shows this expanded view.

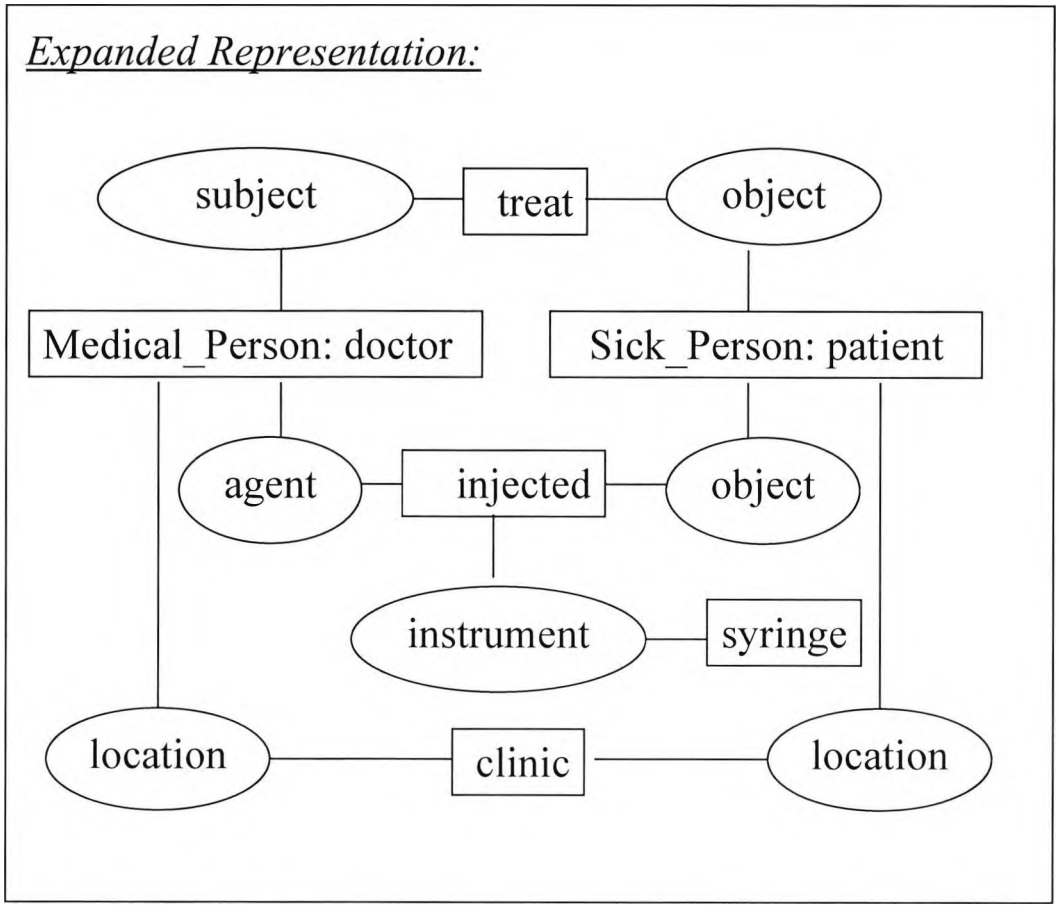


Figure 4.6 *World knowledge interpretation*

Full understanding of natural text is a very complex process and machines are still far from reaching the level of human understanding of natural text. There are several reasons for this:

1. Systems lack the full background knowledge necessary for understanding.

2. The very high complexity of some natural text which requires tremendous computing power that is not available today.
3. Most natural text contains massive ambiguity which needs very deep analysis to be solved. For example, in the sentence "The doctor injected the patient with insulin" a parser may interpret it as "the patient with insulin is injected by the doctor".
4. Some natural text, such as that found in many web documents, lacks correct grammatical sentences and many examples lack complete sentences. Finding sentence boundaries is another problem facing IE systems working on web documents.

For these reasons, today's IE systems tend to skim over text looking for relevant blocks of text and perform partial parsing to identify person names, roles, places, dates, times, etc. Processing only relevant sentences can be crucial to the efficiency of IE systems. Extracting information from a large corpus of text, processing only relevant sentences can also increase the speed of the system. The process of determining the relevance of a sentence or a block of text is called classification, which means classifying a sentence as to whether it is relevant or not. A classifier may also be developed to determine to which class a sentence belongs. We may infer from this discussion that classification can be used for information extraction where the attributes are known and the classifier must find values (tokens) that belong to these attributes. This is discussed in more detail in the next section.

4.2.2 The Machine Learning Approach

The manual construction of information extraction systems is a difficult and a tedious task that requires the engineering of large amounts of knowledge at levels 2 to 6 presented in section 4.2.1. During the last decade, research in IE has shifted towards the automation of rule generation by the use of machine learning paradigms. Machine learning is considered to be a subfield of artificial intelligence and has been inspired by many other fields such as statistics, cognitive science, biology,

information theory, and philosophy. Machine learning is now being applied in many domains ranging from finding protein sequences in genetics, to approving credit applications in banking, categorising documents in information retrieval, tagging part-of-speech and disambiguating word sense in NLP, associating products (market basket) in marketing, clinical diagnosis in expert systems, and many more. Learning algorithms usually take a set of training examples, search for structural descriptions, and output concept descriptions in the form of decision structures which can be used with new unseen data. Decision structures can be expressed as sets of rules. In machine learning there are four different styles of learning (Witten & Frank, 2000):

1. Classification: The learning algorithm induces a classifier based on the training examples (supervised learning). The classifier is to be used with new unseen instances. Some of the most well known methods for classification are decision trees (see Figure 4.7), classification rules, and instant based (i.e. k-nearest-neighbours). All three methods have been used with text classification, for example finding relevant sentences, or to which topic a certain sentence belongs.
2. Numerical Prediction: The examples set and the output classes are numeric quantities. Statistical methods such as linear regression are used to find the weights to be used with the new unseen data to produce the classes.
3. Clustering: Clustering is used when training data are not available (unsupervised learning). Clustering algorithms divide the unclassified collection into groups that falls naturally together. Clustering has been used on a collection of web documents for grouping the collection into separate topics. This is done by using similarity measures to determine the similarity between web documents.
4. Association: This style of learning is used to find associations between the attributes. In other words, the value of any attribute can be predicted and not just the class.

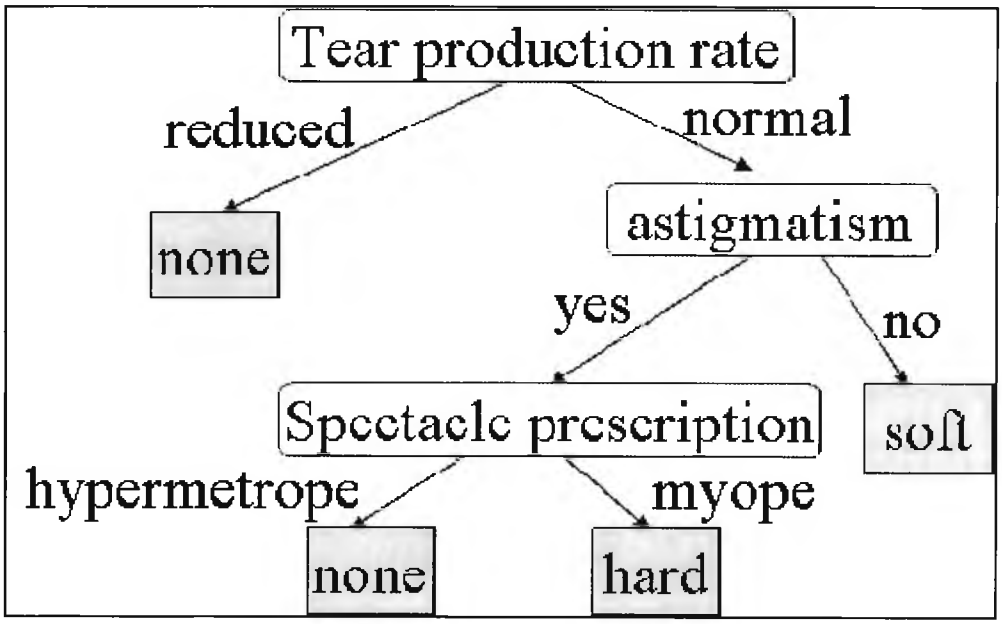


Figure 4.7 *Decision Tree for the contact lenses dataset*[8]

There are several challenging issues facing anyone who wishes to apply learning algorithms such as:

1. Choosing the most appropriate algorithm that performs best for a certain task. There are many learning algorithms that are designed for different tasks and for different types of data.
2. Deciding on the amount of training instances that are sufficient to produce a classifier that performs well on the unseen instances.
3. How to go about testing and validating the new classifier is another issue.

Training examples for IE take the form of annotated text with the information to be extracted. The set of annotated text is then used by an appropriate machine learning algorithm to discover a set of rules that can be applied on a new unseen text. In the following subsections the state-of-the-art in machine learning approaches that have been used with IE are reviewed, namely symbolic relational learning and statistical models.

[8] A well-known dataset used by the machine learning community

4.2.2.1 Symbolic Relational Learning

The utilisation of symbolic relational learning in information extraction systems allows for inducing rules from examples. One of the most used approaches is Inductive logic programming (ILP). ILP is a research area formed at the intersection of logic programming and machine learning (Muggleton, 1991). ILP can be viewed as automatically inferring first-order rules from examples. First-order rules are represented as a collection of first-order Horn clauses, for example PROLOG programs. One approach in the induction of first-order rules works as follows: Given a set of examples E and a set of background knowledge B, find a hypothesis H that satisfies the constraint $B \Lambda H \vdash E$, which is read as "B and H entails E". In this approach, the induction is viewed[9] as simply the inverse of deduction, for example: E follows deductively from B and H. The introduction of background knowledge adds richer description to the training examples and helps in guiding the search for the set of hypotheses H, however, this comes with the price of an increase in complexity as the size of the search space becomes larger. There are many well-understood algorithms for deductive inference described in the literature. A second more common approach to ILP is based on a program called FOIL (Quinlan & Cameron-Jones, 1993). FOIL uses a hill-climbing (general-to-specific) search to learn new rules. FOIL seeks only rules that cover positive examples by applying a performance measure.

4.2.2.2 Statistical Methods

Statistical methodology employs well-known conditional probability techniques such as Naive Bayes and Hidden Markov Models (HMM) to induce unknown values for the attributes in the templates (Charniak, 1993; Jelinek, 1997; Jurafksy & Matrin 2000). Naïve Bayes method assumes that all attributes are equally important and contribute equally and independently to the final target. While this assumption is

[9] This view has been first expressed in 1874 by the nineteenth century economist W. S. Jevons.

unrealistic in real world examples, research has proved that it works surprisingly well in many applications and outperforms many other methods (Mitchell, 1997).

Naïve Bayes is defined as follows: If we have a hypothesis H and evidence E then, Pr(H|E) = Pr(E|H) * Pr(H) / Pr (E), where Pr(H|E) is read the probability of H given E. Pr(H) is called the prior probability, Pr(E) is evaluated as being *one* when the equation is normalised. Sometimes, when an event is equal to *zero,* Pr(H|E) becomes *zero*, in this case a Laplace-estimator is used by adding small weights to the numerator and denominator.

The Hidden Markov Model is another probabilistic model that has been applied successfully to many problems in natural language processing (NLP) such as speech recognition, part-of-speech (POS) tagging, named-entity recognition, and finally information extraction (Freitag & McCallum, 1999). The basic technique of HMM is to learn a generative probabilistic model of the state transition structure from a set of tagged documents. The model is then applied on new unseen documents to determine the target values.

In more detail, an HMM model is composed of:

1. a set of states $Q=\{q_0, q_1,\ldots,q_n\}$, where q_0 is the start state and q_n is the final state
2. a set of transitions between the states $A=\{a_{01},a_{12},\ldots,a_{(n-1)n}\}$, where a_{ii+1} is the transition from q_i to q_{i+1}
3. a set of observation sequence $O=\{o_1, o_2,\ldots,o_t\}$, that is produced as an output by the model.

States are represented as nodes and transition probabilities as edges. When applying HMM to IE, each state is associated with an attribute or a class (i.e. person name, title, etc.) and emits words from the class-specific unigram distribution. Class-specific unigram distribution and state transition probabilities are learned from the training set.

The goal is to find an optimal state sequence with the highest probability for outputting the desired observation sequence. The highest probability for finding the desired path from the start state to the end state can be computed using a variation of the forward algorithm, a dynamic programming algorithm called the *viterbi* algorithm. For more details see (Rabiner, 1989).

One advantage of using probabilistic models such as Naïve Bayes and Hidden Markov Model is that they have a strong statistical foundation; however, they both need large numbers of training examples.

4.3 Information Extraction Systems

During the last decade many IE systems have been developed. In this section, well-known state-of-the-art systems are reviewed.

4.3.1 AutoSlog

AutoSlog is one of the first attempts at relaxing the knowledge engineering bottleneck for IE by auto-generating concept node definitions (also called IE dictionaries) that are capable of extracting information from domain-specific documents (Riloff 1993). These dictionaries used to be built manually (i.e. the knowledge engineering phase) which requires expertise in the domain of discourse and hundreds of hours in construction time. Concept node definitions normally contain syntactic and semantic information that are needed for extracting relevant text. The resultant dictionary in the case of AutoSlog becomes the input of another system called CIRCUS. CIRCUS is an IE system based on the knowledge engineering approach.

The domain used for this work was news stories about terrorist activities in Latin America. Figure 4.8 shows an example of a concept node definition adapted from Riloff (1993).

Id: DEV-MUC4-0657 **Slot filler:** "public buildings"

Sentence: (in la oroya, junin department, in the central peruvian mountain range, public buildings were bombed and a car-bomb was detonated.)

CONCEPT NODE

Name:	target-subject-passive-verb-bombed
Trigger:	bombed
Variable Slots:	(target (*S* 1))
Constraints:	(class phys-target *S*)
Constant Slots:	(type bombing)
Enabling Conditions:	((passive))

Figure 4.8 *Concept node definition generated by AutoSlog*

In the process of generating the IE dictionary, AutoSlog uses triggering words which are called concept anchors, such as "murdered" and "bombed". Concept anchors together with linguistic patterns (see Figure 4.9) are used to build the domain-specific concept node definitions that are used for information extraction.

Linguistic Pattern	***Example***
<subject> passive-verb	*<victim> was murdered*
<subject> active-verb	*<perpetrator> bombed*
<subject> verb infinitive	*<perpetrator> attempted to kill*
<subject> auxiliary noun	*<victim> was victim*
passive-verb <dobj>	*killed <victim>*
active-verb <dobj>	*bombed <target>*
infinitive <dobj>	*to kill <victim*
verb infinitive <dobj>	*threatened to attack <target>*
gerund <dobj>	*killing <victim>*
noun auxiliary <dobj>	*fatality was <victim>*
noun prep <np>	*bomb against <target>*
active-verb prep <np>	*killed with <instrument>*
passive-verb prep <np>	*was aimed at <target>*

Figure 4.9 *Linguistic Patterns that are used by AutoSlog*

A drawback of AutoSlog is that it requires a human expert to filter many irrelevant patterns out of the constructed dictionary.

The system has been tested on 1500 articles that were prepared for the MUC-4 conference. Table 4.1 shows the reported results.

System/Test Set	Recall (%)	Precision (%)	F1-measure
MUC-4/TST3	46	56	50.51
AutoSlog/TST3	43	56	48.65
MUC-4/TST4	44	40	41.90
AutoSlog/TST4	39	45	41.79

Table 4.1 *Empirical evaluation of AutoSlog*

In this table, these results are being compared against MUC-4 results. Considering that the knowledge engineering in MUC-4 was performed manually, AutoSlog's performance was considered a breakthrough.

4.3.2 PALKA

PALKA (Parallel Automatic Linguistic Knowledge Acquisition) is another system for generating linguistic patterns automatically that are used in information extraction (Kim & Moldovan 1993). The generated patterns, called frame-phrasal structures (FP-structures), are constructed by merging meaning frames and phrasal patterns as shown in Figure 4.10. PALKA relies on prior sentence analysis to identify syntactic elements and their relations. Also, it utilises predefined concept hierarchies for mapping the training texts to FP-structures. A correct interpretation will result in producing the desired FP-structure.

The FP-structures are then used to extract information for new unseen news stories in a straightforward manner. PALKA (like AutoSlog) relies on exact words for the FP-structures to get activated.

FP-structure = Meaning Frame + Phrasal Pattern

Meaning Frame:

(BOMBING
agent: ANIMATE
target: PHYSICAL-OBJ
instrument: PHYSICAL-OBJ
effect: STATE)

Phrasal Pattern:

((BOMB) BE HURL AT (PHYSICAL-OBJ))

FP-structure:

(BOMBING
target: PHYSICAL-OBJ
instrument: BOMB
pattern: ((instrument) BE HURL AT (target))

Figure 4.10 *FP-Structure representation used by PALKA*

The system was tested on 500 MUC-4 texts. It was able to extract 220 sentences with "BOMBING" frame creating 67 FP-Structures, and 501 sentences with "KILLING" frame creating 37 FP-Structures. The average creation for the first frame was 30.5% and for the second frame was 6.2%.

4.3.3 CRYSTAL

CRYSTAL is another syntactically based system that works like AutoSlog which generates concept node definitions automatically (Soderland et al., 1995). These generated concept node definitions are fed to another system called BADGER (which works like CIRCUS) for sentence analysis. Figure 4.11 shows one concept node definition that is generated by CRYSTAL. A distinguishing feature of CRYSTAL over AutoSlog is that it only requires a set of documents along with their pre-existing initial set of concept node definitions for training. CRYSTAL uses inductive learning approach while AutoSlog uses Memory-based approach (i.e. lazy approach).

```
CN-type:                Diagnosis
Subtype:                Pre-existing
Extract from Prep. Phrase "WITH"
Passive voice verb
Verb constraints:
        words include "DIAGNOSED"
Prep. Phrase constraints:
        preposition = "WITH"
        words include "RECURRENCE OF"
        modifier class <Body Part or Organ>
        head class <Disease or Syndrome>
```

Figure 4.11 *Concept node definition generated by CRYSTAL*

The concept node definitions generated by CRYSTAL were for the domain of "Hospital Discharges". The system utilised UMLS for the semantic tagging as it requires a semantic hierarchy for the domain and a list of concepts. CRYSTAL extracts both single and multiple slots and operated on semi and free text.

4.3.4 LIEP

LIEP (Learning Information Extraction Patterns) is another syntactic based system that auto-build dictionaries from examples (Huffman, 1995).

```
n_was_named_t_by_c:
        noun-group(PNG,head(isa(person-name))),
        noun-group(TNG,head(isa(title))),
        noun-gourp(CNG,head(isa(company-name))),
        verb-group(VG,type(passive),head(named or elected or appointed)),
        preposition(PREP,head(of or at or by)),

        subject(PNG,VG),
        object(VG,TNG),
        post_nominal_prep(TNG,PREP),
        prep_object(PREP,CNG)
===> management_appointment(M,person(PNG),title(TNG),company(CNG)).
```

Figure 4.12 *Extraction Pattern auto-built by LIEP*

The chosen domain for this system was "management change" where the extracted entities are person name, title, and company name. The output of LIEP is set of patterns that are fed to another system called ODIE (On-Demand-Information-Extraction). ODIE works the same way CIRCUS does for AutoSlog and BADGER for CRYSTAL. Such systems need sets of concept node definitions to use for extracting information. Figure 4.12 shows the extraction pattern auto-built by LIEP. A basic difference between LIEP and AutoSlog is that LIEP has the ability to extract multiple slots extraction rules while AutoSlog is capable of one slot only. Also, LIEP provides real-time tagging of relevant events in the case of no training corpus is available. The patterns generated by LIEP were tested with ODIE. In terms of the F-measurement, the system performed at a level of 85.2%.

4.3.5 RAPIER

RAPIER (Robust Automated Production of Information Extraction Rules) is another system that uses machine learning for inducing extraction rules (Califf & Mooney 1997). The system uses a bottom-up relational learning algorithm inspired by some ideas from ILP (Inductive Logic Programming). It utilises an external POS tagger for identifying syntactical parts. The domain used was computer-related job announcements where information such as job title, employer, salary, and location were extracted. Figure 4.13 shows a rule that has been generated by RAPIER which consists of three parts: Pre-filler pattern, Filler Pattern, and Post-filler Pattern.

Pre-filler pattern:	**Filler Pattern:**	**Post-filler Pattern:**
1) word: in tag: in	1) list: len: 2 tag: nnp	1) word: , tag: , 2) tag: nnp semantic: state

Figure 4.13 A *generated rule by RAPIER*

RAPIER has been tested on 100 documents paired with a filled template using 10-fold cross validation. It has achieved a recall of 63% and precision of 89%. Results

have shown that RAPIER generally outperformed previous systems in terms of precision and also required less human intervention.

4.3.6 WIEN

WIEN (Wrapper Induction ENvironment) is an environment for automatically constructing wrappers (Kushmerick et al., 1997). The methodology used for wrapper induction was somewhat untraditional for the IE community as it requires no linguistic processing (i.e. syntax or semantic processing). All preceding systems required text to be in full grammatical sentences (i.e. news documents). These systems cannot handle tabular data on the web which wrappers can. Wrappers are solely based on learning delimiters that precede and follow the actual data to be extracted. Figure 4.14 presents a pseudo-code procedure for executing a wrapper known as HLRT wrapper and returning the extracted values from page P.

```
ExecHLRT(wrapper<h,t,l1,r1,...,lk,rk>,page P)
  skip past the first occurence of h in P
  while the next occurrence of l1 is before the next occurrence of t in P
        for each <lk,rk> ∈ {<l1,r1>,...,<lk,rk>}
          extract from P the value of the next instance of the k^th attribute
          between the next occurrence of lk and subsequent occurrence of rk
  return all extracted tuples
```

Figure 4.14 *Pseudo-code for executing HLRT wrapper developed for WIEN*

One problem of wrappers is that if a website changes its structure, then the learned wrapper will not work and a new one needs to be induced. Wrappers have been discussed in more detail in section 4.2. WIEN is the first system to construct wrappers automatically.

4.3.7 SRV

SRV (Sequence Rules with Validation) reduces the problem of information extraction to the problem of "slot filling" only (Freitag, 1998). It does not apply any

syntactic analysis and it operates on semi-structured text which makes it suitable to extract information from the web. The algorithm tries to find the best fragment of text to fill a given slot in the template. SRV extracts single slots and treats each field in isolation. Figure 4.15 shows some rules learned by SRV for course number along with text sample for matching.

```
coursenumber:-
  length(=2),
  every(in_title false),
  some(?A [previous_token] in _title true),
  some(?A [] after_p false),
  some(?B [] tripleton true)

 <title> Course Information CS213 </title>
 <h1> cs 213 C++ Programming </h1>
```

Figure 4.15 *SRV's learned rule for course number and a matching text sample*

In this work, the IE problem is looked at as a text classification problem, which means the system tries to find the best value for a given attribute. To do so, it combines three classifiers; rote learner, naïve Bayes, and relational rule learner. As an example, SRV learns personal names from orthographic features (i.e. capitalisation, length of words, etc.).

4.3.8 WHISK

WHISK is the first system to work with tabular, semi-structured, and free text (Soderland, 1999). The system is capable of extracting single slots and multiple slots. Web documents are segmented using the HTML tag <hr>. The system uses different methods for extracting information based on text structures; For tabular data the rules are based on delimiter strings while for free text more steps are involved like syntactic and semantic analysis. Syntactic processing is needed with free text only.

WHISK has addressed a very important type of text that falls between tabular and free text which is often found on web pages. Such text is ungrammatical but does not follow any rigid format. WHISK uses some form of regular expression patterns for extracting facts from relevant text. Figure 4.16 shows an example of this type of text about apartment rental and a rule to extract the number of bedrooms and price.

```
<p>Camden Town - 1 bedroom apartment on thrd flr.<br>
Furnished, heat paid, Close to shopping and schools, <br>
perfect to share, £867/month + 1 month deposit, <br>
call now <br></p>

ID:: 1
Pattern:: *(Digit)'BR'*'£'(Number)
Output:: Rental {Bedrooms $1} {Price $2}
```

Figure 4.16 *Sample text and a pattern generated by WHISK*

WHISK uses supervised learning, a top-down covering algorithm, to induce rules in the form of regular expressions. For evaluation, a ten-fold cross validation on 100 randomly selected announcements has been used.

4.3.9 WEB->KB

This system belongs to newer generation of IE systems in which the problem of IE is viewed as a classification problem (Cavern, 2000). These systems (including this one) rely heavily on machine learning approaches to identify and classify related entities. Another feature of this system is the utilisation of an ontology that describes the domain of discourse which is university related websites, in particular webpages about students, their projects, instructors, and courses. The extracted information is then populated into a knowledge base. Two machine learning approaches have been used; a statistical approach (Naïve Bayes) and a symbolic relational learning approach, both of which have been explained in a previous section. Figure 4.17 shows an example of the relational learning approach. The upper part of the figure shows an extraction rule describing the name of the web page owner and the bottom

part shows an example of an HTML fragment which this rule matches. The underlined portion is the name to be extracted.

```
ownername(Fragment) :- some(B, [ ], in_title, true),
                       length(<, 3),
                       some(B, [prev_token], word, "gmt"),
                       some(A, [ ], longp, true),
                       some(B, [ ], word, unknown),
                       some(B, [ ], quadrupletonp, false)
```

```
<title> Bruce Randall Donald</title>
<h1>
<img src="ftp://ftp.cs.cornell.edu/pub/brd/images/brd.gif">
<p>
Bruce Randall Donald<br>
Associate Professor<br>
```

Figure 4.17 *Extraction rule and HTML fragment that matches it*

The system has achieved approximately 70% accuracy with a coverage of 30%. Their experiments have shown that relational learning performed better than statistical learning in terms of accuracy but was lower in coverage. One reason is that relational learning provides more descriptive rules. Also, these rules were able to take advantage of hypertext.

4.3.10 The Artequakt Project

The Artequakt project is a large European project[10] which aims to implement a system that searches the web and extracts knowledge about artists, populates a knowledge base, and uses it to generate personalised narrative biographies (Allani et. al., 2002). The system reuses several components for automating the knowledge acquisition phase; these components are:

[10] Consists of three separate projects: the Artiste project, the equator IRC project, and the AKT IRC project. See the paper for more information on these projects.

1. An ontology representing the domain of the artist which has been built using Protégé.
2. Apple Pie Parser is used for syntactic analysis.
3. WordNet is used for semantic analysis.
4. GATE is also used for semantic analysis (in particular named entity recognition such as person names, dates, and locations) and to resolve anaphoric references.

The extracted information is saved in XML format and sent to the artist's ontology through the protégé API. The paper does not provide any information about evaluation.

4.3.11 Textpresso

Textpresso is an ontology based information retrieval and extraction system for the domain of biological literature (Muller et. al., 2004). The extraction system extracts information such as title of the literature, authors, abstract, journal, year of publication, etc. Syntactic information about the structure used in the document has been utilised for building the extraction patterns. Textpresso also processes and analyses the full text of the articles. The system utilises a domain specific ontology for categorising sentences. It uses regular expressions in the process of categorisation. Figure 4.18 shows a small set of categories along with their regular expressions and sample text.

Category	Regular Expression	Example
Gene	[A-Za-z][a-z][a-z]-\d+	let-60
Molecular function	[cC]oreceptors?	coreceptor
Cell or cell group	[Hh]ypoderm(aljis)?	hypodermal 3
Organism	[Zz]ebra(-j\s)?[Ff]ish	zebrafish
Allele	n\d\d?\d?\d?	n695
Transgene	[a-z][a-z]?ls[0–9]+	kuIs14

Figure 4.18 *A sample of categories with regular expressions and examples adapted from (Muller et. al., 2004)*

More recently tens of IE systems have been developed for different tasks in different domains, with many of them intended to support the semantic web. Some of these systems have shown improvements, for example, by testing different machine learning algorithms, or even a combination of several methods to improve performance. Other systems have aimed at using smaller sets of examples for training.

4.4 Problems with Current Information Extraction Technology

The machine learning approach has solved some of the problems that exist with the knowledge engineering approach; these are:

1. Elimination of the need to write the extraction rules manually which can be a very time-consuming process.
2. Deep knowledge of NLP is not required.
3. Knowledge of programming (i.e. constructing regular expressions) is also not required.
4. The learning approach is domain-independent.

However, the learning approach has created other problems and these are:

1. The need to construct training examples which could be time-consuming and knowing that most learning algorithms require a large number of examples for greater accuracy.
2. Decreased efficiency in performance. The manually created rules perform better.

Both approaches lack adaptability to new domains[11]. In the knowledge engineering approach a completely new system is needed to process documents belonging to the

[11] More on this issue has been presented in chapter 2.

new domain and in the learning approach a new set of examples belonging to the new domain is needed.

4.5 Summary

In this chapter information extraction has been reviewed. First, a small historical account has been presented along with how the discipline has been evolving. Next, it has been compared with IR, a more mature and closely related discipline. Also, the techniques for evaluating IE systems have been presented. In the main sections of this chapter the approaches to IE have been explored and discussed. Then a quick overview of some of the most-known IE systems has been presented.

Upon close examination of these IE systems, it can be observed that the early ones depend greatly on grammatical structures of sentences extracted from news stories. Systems such as AutoSlog, PALKA, and LIEP, which were developed during the early 90s, utilise part-of-speech and syntactic analysis to locate the desired information. This can be observed by looking at Figures 4.8 - 4.12. Due to the grammatically ill-formatted nature of the web in general, such systems would fail terribly if tried on real web documents. During the second half of the nineties, a second class of IE systems was developed which is based on wrapper induction and regular expression techniques. These systems such as WIEN, WHISK, and Textpresso, use surrounding patterns (i.e. characters surrounding the desired information) to identify target words. These systems would fail on a webpage that has no repetitive patterns. More recent systems utilise metadata and semantic information[12], which classify words found within text to determine its type. Systems like the Artequakt project and this current project depend heavily on the semantic types of words to recognise the deferent entities and to restructure them into attribute-value pairs. Concept based recognition and extraction of information seem

[12] May also use regular expressions (i.e. to find prices and dates), but more biased towards semantic analysis.

to be the direction of the future. Also, it seems more natural, since the same mechanism of recognising entities based on their categories is used by humans.

Finally, the problems of each approach have been discussed. In the next chapter diabetes will be reviewed.

Chapter Five

Application Domain - Diabetes

"Diabetes is a growing and massive silent epidemic that has the potential to cripple health services in all parts of the world."

- Robert Beaglehole, 2004

5.1 Introduction

The need to automate the process of annotating web documents with semantic metadata is a very important step towards the semantic web. The aim of this work is to develop a framework that contributes towards the automatic construction of a semantic layer to a given website. The semantic layer is made of "nuggets of knowledge" that give a fair description about the website and its content. This can be done with the guide of a domain-specific ontology. The domain of diabetes management has been chosen for good reasons.

Recently it has been estimated that over 194 million people worldwide suffer from diabetes mellitus. The numbers of people affected with this silent deadly disease will most likely double by the year 2025[13]. It is a silent disease because at least 50% of people with diabetes are unaware of their condition and is a deadly disease because it is the fourth main cause of death in most developing countries. Worse still, the disease is known to cause a number of complications like blindness, foot problems, heart disease, high blood pressure, kidney disease, nervous system

[13] According to the World Health Organisation (WHO) and the International Diabetes Federation (IDF).

disorder, and strokes. Such complications impose a huge burden on governmental health agencies in terms of human resources and costs.

An early detection and a good management of this chronic disease can severely reduce the risk of complications and therefore reduce the huge burden on governments and patients as well.

In this chapter a brief overview of diabetes mellitus is given. This includes the types of diabetes mellitus, symptoms, diagnosis, complications, and management.
Some 100 diabetes-related websites dealing with different aspects of the disease have been extracted for offline experiments. In this chapter there is also a study of the characteristics of these websites. This study is used to show how these characteristics can help us in the work being described.

5.2 What is Diabetes?

Diabetes mellitus is one of the most common endocrine disorders resulting from hypo-secretion of insulin, decreased effects of insulin, or malfunction of insulin receptors on the target cells (Thibodeau & Patton, 1999). Insulin is a protein hormone that is produced in the beta cells of the pancreatic islets of Langerhans and target many body organs. Insulin regulates glucose, the body's main source of energy, which is taken in by most body cells.

5.3 The Types of Diabetes

There are two major types of diabetes mellitus:

1. Type 1: is also known as insulin-dependent diabetes mellitus (IDDM). This type was also known as juvenile-onset diabetes because it mainly affects young people (11 to 30 years old). The beta cells of people with type 1 diabetes are damaged and completely unable to produce insulin. There are several factors that cause this type of diabetes such as genetics, viral

infections, and autoimmune disorders. Type 1 patients are required to take insulin injections on a daily basis for the rest of their lives.

2. Type 2: is also known as non insulin-dependent diabetes mellitus (NIDDM). This type was also known as maturity-onset diabetes because it affects mainly older people (above the age of 40). Type 2 is the most common type with 90% of the total number of all diabetic patients. The beta cells of people with type 2 are functioning, but insulin is produced in smaller quantities. Another main cause of type 2 diabetes is the loss of some insulin receptors on the target cells which means that these cells receive less insulin; therefore less glucose is consumed and transformed to energy. Type 2 patients are not necessarily required to have insulin injections; in many cases they can live a normal life with an appropriate diet, exercise, and weight control. Some patients may need to take medicines (oral hypoglycaemic agents) that simulate the beta cells to produce more insulin or increase the number of insulin receptors on the target cells. The main factors that cause this type of diabetes are genetic and environmental factors.

5.4 Diabetes Symptoms

The absence of normal quantities of insulin causes an increase in the glucose concentration in the bloodstream. The excessive quantity of glucose in the blood, a syndrome known as hyperglycaemia, flows into urine by osmosis (glycosuria) and the body loses large quantities of water in the process. This causes body dehydration and result in excessive thirst (polydipsia) and the tendency to drink large quantities of water. Passing large amounts of urine (polyuria) is another symptom of diabetes. Depriving cells from utilising glucose for energy causes the body to ask for food (polyphagia), thus hunger is yet another symptom of diabetes. While these symptoms are the most common and are the ones that diabetes practitioners usually look for in patients, there are other symptoms like vomiting, abdominal pain, a fruity odour of the breath, weight loss, and coma. Symptoms in

patients with type 2 can be much harder to recognise than in patients of type 1, and many people with type 2 diabetes are unaware of their condition.

5.5 Diagnosis of Diabetes

Diabetes mellitus can be diagnosed by testing the level of glucose in the blood. In normal people the level of blood glucose stays between 3 and 7 mmol/l throughout the day. Diabetic people on the other hand have unstable levels of blood glucose which can vary from below 3 mmol/l to almost three times the normal level. Diabetic patients can also be diagnosed by testing the level of glucose in the urine. Excessive glucose is drawn from the blood to the urine.

5.6 Diabetes Complications

People with diabetes who are left untreated will develop complications. This is due to the fact that the human body will look for other resources of energy such as fat. The shift from sugar to fat causes an increase of lipid levels in the blood (hyperlipidaemia). This will result in a range of complications that affect many parts of the body. In the following subsections the most common complications are reviewed.

5.6.1 Blindness

Blindness is one the most common complications of diabetes. The disorder is caused by damage in the retina, the part of the eye that intercepts light and forms our vision. Lipid that reaches the retinal blood vessels weakens and sometime damages these vessels causing internal bleeding in the eye. Patients become less sensitive to the light entering the eye and therefore their normal visual functionality is affected. Another common eye disorder for people with diabetes is the rise of blood pressure inside the eye (glaucoma). Symptoms of retinal disorder may take around 30 years to appear.

5.6.2 Heart Disease

Heart disease is considered the number one cause of death for people with diabetes. The formation of lipids, which adhere to the walls of the vessels causes the flow of blood towards the heart to become reduced. Another factor is the excess of glucose in the bloodstream, which causes a stiffening of the blood vessels. Prevention of the blood from reaching the heart at normal levels is the major cause of heart disease.

5.6.3 High Blood Pressure

Surveys have shown that approximately 60% of patients with diabetes are affected with high blood pressure. The real cause of high blood pressure in patients with diabetes is still unknown. However, there are some theories connecting high blood pressure with insulin resistance, weight disorder, and hormone disorder.

5.6.4 Kidney Disease

The kidneys are the main organs of the urinary system. They are responsible for getting rid of waste by filtering the blood plasma. The extra volumes of glucose in the blood affect the kidneys' performance. The unwanted bodies (waste) may stay in the blood plasma while proteins and other nutrition may flow out of the body with the urine instead. The extra efforts that are imposed upon the kidneys can lead to kidney failure.

5.6.5 Nervous System Disorders

Scientists believe that the excess of glucose levels in the blood affects the chemical equilibrium inside the nervous system. This can impact on speech abilities, body movement, and vision.

5.6.6 Strokes

The excess of lipid and sugar in the blood vessels located in the brain can cause internal bleeding and therefore strokes occur. Also depriving brain cells with blood causes them to die and this affects other activities such as vision, the ability to speak and body movement.

5.6.7 Foot Problems

The reduced blood flow towards the feet can cause chronic ulceration. Worse still, it can cause gangrene which leads to amputation of the affected parts.

5.7 Diabetes Management

People with diabetes can live normal lives if they practice good management of their condition. As for patients with type 1 diabetes, they must have regular insulin injections for the rest of their lives. In the past, insulin used for such injections was collected from the pancreas glands of cows and pigs. However, now a new form of insulin is made in the laboratory and its chemical structure resembles that found in humans. It is important for patients with type 1 diabetes to understand the nature of insulin injections. Insulin processed for patients' injections comes in three forms; short acting, medium acting, and long acting. Each form has different characteristics in terms of purity, duration, effect, strength, onset after injection, and source. More information about insulin injections can be found in diabetes-related websites and literature.

Controlling type 2 diabetes can be achieved through diet, exercise, and weight control; some patients may need prescriptions of oral hypoglycaemic agents. The management of type 2 diabetes may sound easy but this is not the case! Normally type 2 diabetes strikes later in life, from 40 years of age onward, which means that a person who has been adapted to a certain life style may find it very hard to change to a new one. Because of this, many people with type 2 diabetes develop complications

and may become insulin-dependent and progress to type 1 diabetes. Another reason for developing complications is that patients may not know about their condition until too late.

As seen in the above sections the main factor for the development of complications is the rise of lipid levels in the blood. Unfortunately, the number of people with obesity has been rising at alarming rates in many countries (increasingly including younger people). Studies have shown that people with obesity have high diabetic rates and are subjected to complications such as heart diseases and stroke.

Many diabetes-related websites publish food recipes that are specially designed for people with diabetes, but this does not replace medical consultation. Each patient has different needs based on his/her own condition and disease progression. Medical consultation alone is not enough. Patients need to be educated about diabetes management in general and about their own condition in particular. It has been found during this study (see next section) that over 70% of diabetes-related websites contain general information about diabetes and some offer more-in-depth educational material about this chronic disease.

The following actions must be taken on regular basis to insure good management of the disease:

1. Establishment of patient's knowledge of illness and educational needs
2. Measurement of height and weight
3. Measurement of blood pressure
4. Testing of urine for ketones and protein
5. Testing blood for glucose
6. Examination of feet
7. Examination of eyes

5.8 Diabetes-related Websites

During the course of this work a set of 100 diabetes-related websites has been collected[14] and their contents and structures were analysed. This set contains around 12000 files and over 200 MB in size.

The most import feature about any website is its purpose. Knowing the purpose of a website gives a good hint on what type of information to expect. For example, if a website's purpose is to sell books then it is expected to find a list of names of books along with other information such as ISBN number, name of author(s), prices, etc.

Determining the purposes of a website makes extracting information much more efficient.

As mentioned in the objectives of this thesis, the process of extracting knowledge from websites can be automated with the guide of ontology. The ontology is needed first to determine the purpose of the website and then to guide the IE system to which pieces of information to extract.

Rather than guessing a list of purposes that a diabetes-related website may have, an empirical study was carried out on the collection and a set of purposes (and other information about the websites) has been extracted. Figure 5.1 shows the header of the columns of the spreadsheet showing the purposes categories. The full list that shows the websites is presented in Appendix A. The following abbreviations have been used in the table:

1. Education (EDU): a website provides educational services, or belongs to an educational institute.
2. Information (INF): provides basic information about diabetes such as definitions, types, symptoms, complications, etc.

[14] Only textual files has been collected such as with HTML, ASP, XHTML sources and no graphical (i.e. GIF, JPG) or other multimedia files.

3. Products (PRO): patients can order diabetic products such as insulin syringes online.
4. Research (RES): provides support for researchers, or the website belongs to research institute.
5. Portal (POR): has many links to other diabetes-related websites. The links are categorised and organised in some hierarchy.
6. Membership (MEM): requires registration to use, or provide some type of membership such as subscription to newsletters.
7. News (NEW): offer news about diabetes-related events.
8. Consultation (CON): offers online consultation to patients, and families of patients.
9. Charity (CHA): accept donations.
10. Society (SOC): a society belongs to some regional community that provides support for its members in particular.
11. Metadata (MET): a web site may have meta-tags imbedded in the HTML source which provides keywords about the content of the website.
12. Date (DAT): shows the date of last modification to the website.

Features 1 to 10 can be considered as the categories or purposes of these websites. All websites have more than one category, for example some websites are educational (EDU), provide general information about diabetes (INF), and accept donations at the same time.

Purpose											
EDU	INF	PRO	RES	POR	MEM	NEW	CON	CHA	SOC	MET	DAT

Figure 5.1 *Headers of columns in the spreadsheet showing purpose categories*

Figure 5.2 shows the percentage of websites for each of the features presented above. From this figure it has been noted that 70% of all websites present information about diabetes in general, and such information is basically the same and recurs many times. Such information is usually not the real purpose of the

website! Therefore, there is no need for this part to be transformed into formal knowledge since it is made only for human consumption and for people who need general information about the disease. However, it should be noted in the semantic layer that such a website contains general information.

Looking again at Figure 5.2 we see that the presence of meta tags comes next with 60% of all websites. However, on close examination of the content of these meta tags it has been noticed that most of them carry general keywords about diabetes and just a very few contain the real purpose of the website.

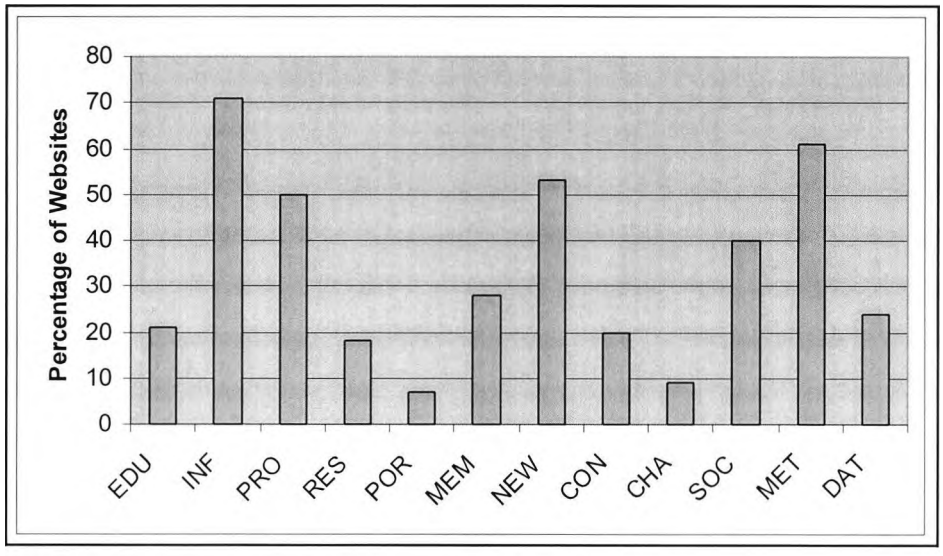


Figure 5.2 *Percentage of websites with categories and other features*

Third comes news (NEW); around 50% of all websites contains news. Most organisations use their websites to disseminate news about local and global events. For example, educational and research oriented websites disseminate news about past and future seminars and conferences. Others disseminate social news about their members, events, dates, success stories, etc.

Next in the figure is products (PRO) with 50% of websites. Probably this feature is one of the most important ones for patients affected with the chronic disease that needs life-long management.

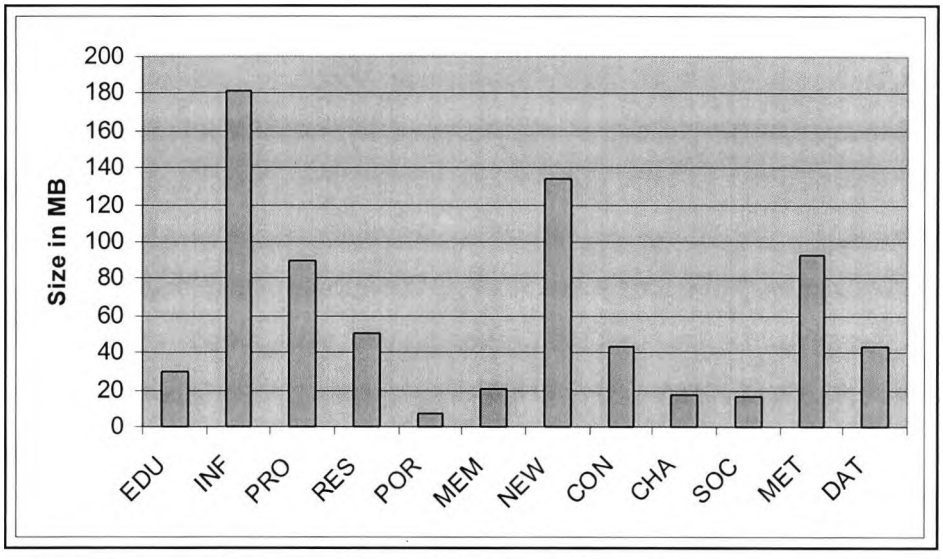


Figure 5.3 *Size in MB of websites holding each of the features*

Figure 5.3 shows the size in megabytes of each category as summed from all websites holding this category. Figure 5.4 shows the total number of files of websites for each category which is similar to the chart presented in Figure 5.3.

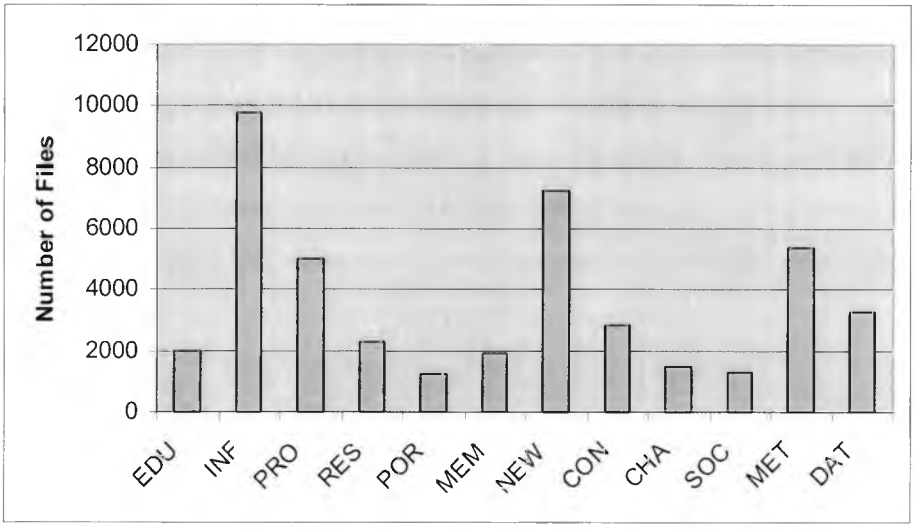


Figure 5.4 *Total numbers of files in websites holding each of the features*

Despite the similarity between these figures there are some differences and for logical reasons. For example, the value of INF in Figure 5.2 is 70%, but in Figure 5.3 it is shown to occupy 180 MB out of 200 MB, that is 90% in size leading to the conclusion that the recurring general information is consuming bytes as well, which

is costly to process by machine. On the other hand, the percentage of society-featured websites is 40%, but Figure 5.3 is showing less than 20MB of the 200MB which is 10% in bytes.

A similar list as that in Figure 5.1 has been developed showing only websites that offer products, see appendix D. The products in this list have been categorised; their abbreviations are as follows:

1. Sff: Sugar free food
2. Nut: Nutrition like herbs, vitamins and minerals.
3. Skn: Skin care (creams and lotions)
4. Ins: Insulin
5. GIT: Glucose monitoring (meters, strips, swabs, lancets)
6. InT: Insulin tools (syringes, pumps, wallets)
7. Bks: Books about diabetes
8. TrK: Travel kits
9. Med: Medicine
10. SW: Software
11. ID: ID bracelets

Figure 5.5 shows the percentage of websites carrying each of the product categories listed above. It has been observed that books about diabetes are the highest carried type of product with 20% of the websites. Next in the figure comes websites presenting insulin-related devices such as syringes and pumps. The third type of product which comes very close to the insulin-related devices is glucose monitoring tools such as meters, strips, and lancets. Each of these products has been noted along with their attributes which will form the templates to be filled by the extraction engine.

This information has been entered in the ontology to be discussed in detail in the next chapter. It has been observed that the purpose of all diabetes-related websites can be categorised as one or other or both of the following:

1. Service providers: these services are shown from Figure 5.1 are education, research, portals, news, consultation, and aid for local communities.
2. Products providers: as listed in appendix D.

As mentioned above the same study has been repeated for the rest of the categories shown in Figure 5.1 (i.e. research, news, etc). A complete hierarchical structure has been drawn to be discussed in next chapter. The same study on other chronic diseases has been established to show their similarity and therefore induce a generalised structure for all medical sites relating to chronic diseases.

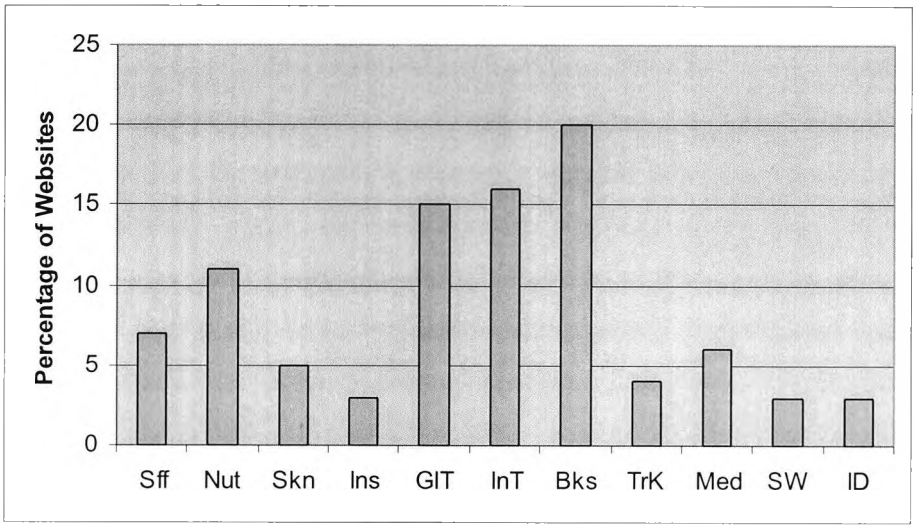


Figure 5.5 *Percentage of websites carrying each product categories*

5.9 Conclusion

In the above section it has been noted that 70% of all diabetes-related website carry general information about diabetes, such as definitions, types, symptoms, and complications. Such information is constructed for human consumption and not machine.

It may be concluded that there are three types of knowledge that make up a website:

1. Knowledge about the website: This includes the title of the website, its IP address, date of last update, author names, etc.
2. Static knowledge: This type is general information about diabetes and repeated in 70% of all diabetes-related websites. This type does not usually change because these are facts about the disease and most likely will not change.
3. Dynamic knowledge: This represents the real purpose of the websites. It has been separated into two main categories: products and services. Each has been analysed using a divide-and-conquer technique. This process derived the ontology to be used as part of the system being developed.

This chapter concludes the introductory part of this work; the next chapter is devoted to the development and implementation of the knowledge extractor system

Chapter Six

System Development and Implementation

We succeed in managing information if we can "[take] data and [analyze] it and [simplify] it and [tell] people exactly the information they want, rather than all the information they could have."

- Jim Gray,
Microsoft Research

6.1 Introduction

One of the main goals of this work is to develop a framework for automating the process of extracting useful information from medical websites. The extracted information is to be saved into a structured format that is ready for the semantic web. This would decrease the burden on web authors from the manual construction of the semantic layer[15]. The proposed framework contains several components starting from pre-processing HTML documents and ending with the final knowledge representation.

In chapter four, the field of information extraction (IE) was reviewed, and the two approaches to IE were presented. It has been shown that IE systems rely on predefined templates. A template contains the fields (attributes) for which their values are to be extracted from a given document. Research has shown that current IE systems are task-specific and hard to adapt to new domains. The implementation of a general IE system that works with any task without the extra work is not feasible at present. In this present work a new approach is introduced. First, an IE system is coupled with two ontologies to guide the system as to which pieces of text

[15] Other uses for the framework are discussed in detail in chapter 9

to extract. The first ontology is the UMLS which is utilised to retrieve concepts for the terms found in the free text. This is done by facilitating the MetaMap transfer (MMTx) system. The second ontology contains common concepts that are shared between many chronic disease-related websites. This ontology is the result of the website analysis that has been carried out during the first part of this research work. A complete listing of the constructed ontology in OWL format can be found in appendix F. The goal of this ontology is to guide the system as to which terms to extract. This approach brings capability and flexibility to the system. In this chapter the development and implementation of the IE system is presented in detail.

6.2 System overview

The ultimate objective of the system is to identify relevant and useful blocks of text embedded within chronic disease-related HTML documents and to transform it into suitable knowledge representation.

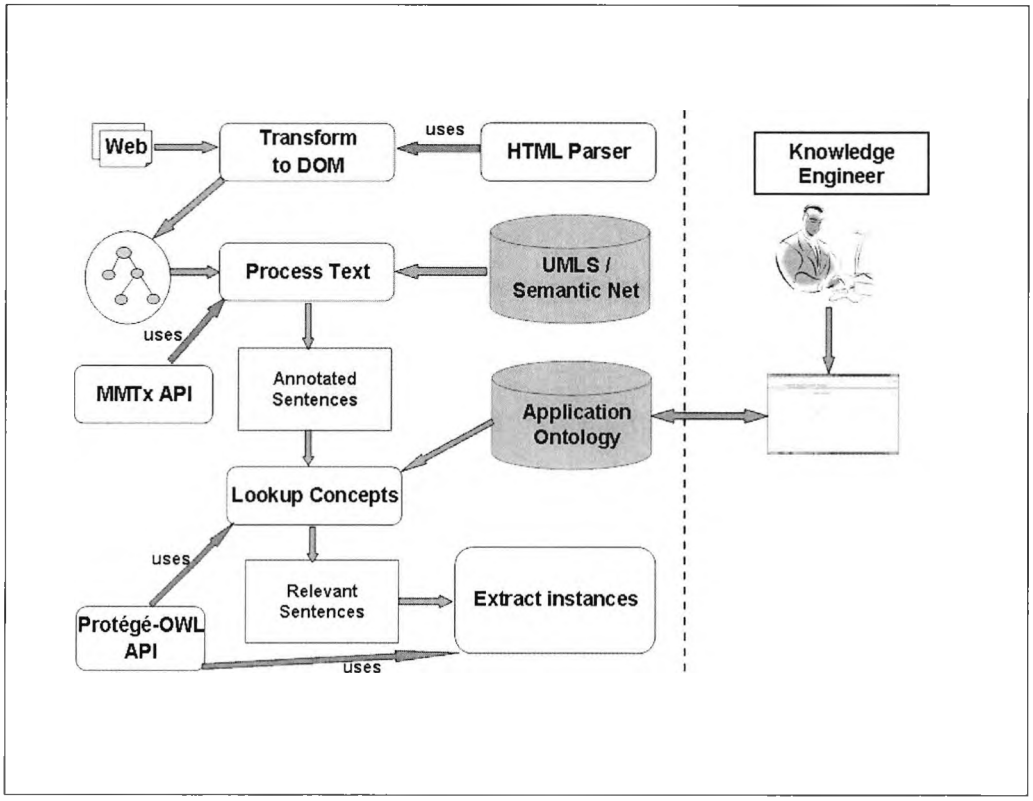


Figure 6.1 *A general view of the proposed system*

In other words, it should identify and extract products and services that these websites offer. Figure 6.1 shows a general view of the Medical Knowledge Acquisition system (MediKA), a system that seeks knowledge in medically-related websites.

The view in this figure shows all inputs, outputs, processes, and external components that the system utilises, which are explained as follows:

a) Inputs of the System:

1. HTML document: To be processed by the system.
2. UMLS semantic net: Used by MMTx API for tagging text.
3. The application ontology: To guide the system as to which pieces of text to extract.

b) Output of the System

The set of extracted instances in the form of attribute-value pairs

c) External Components of the System

1. MMTx API: Tag text with lexical, syntactic, and semantic information
2. Protégé-OWL API: Access the application ontology programmatically. A knowledge engineer may interact with the application ontology externally via the Protégé-OWL interface as shown in the figure above.

d) Processes of the System

1. Pre-process HTML document.

2. Annotate sentences using MMTx tool.
3. Look up marked concepts from the ontology to identify the relevant concepts in the text.
4. Extract attribute-value pairs.
5. Save the extracted instances in a structured form.

The five processing steps that are listed above are performed in that order and show the general approach of the system. Each of these steps will be explained in more detail in the following subsections.

6.2.1 Pre-processing Phase

Most research on information extraction from the web works on a flat HTML document. They treat HTML documents as a sequence of words to be tokenised and grouped (text zoning) for lexical and syntactic analysis. In this sense, web documents are considered noisy streams of data because they are cluttered with images, commercial ads, scripts, and many other irrelevant elements. HTML tags can assist us in many things such as to identify the importance of some text, select relevant features, and solve sentence boundary problems.

Web documents can be better viewed as hierarchical models (Gupta, 2003), where the <HTML> tag is usually set at the root node. In this current work, an HTML document is considered as an object. To preserve the structure of the html document, first a document is transformed into XML using an HTML parser and then into a Document Object Model (DOM). Figure 6.2 shows a small extract from the HTML document (in the top part) and its correspondent DOM representation (in the bottom part). For more details on DOM see chapter two.

```
<TABLE>
   <TBODY>
   <TR>
   <TD>NovoRapid</TD>
   <TD>Insulin Aspart</TD>
   </TR>
   <TR>
   <TD>Humalog</TD>
   <TD>Lispro</TD>
   </TR>
   </TBODY>
</TABLE>
```

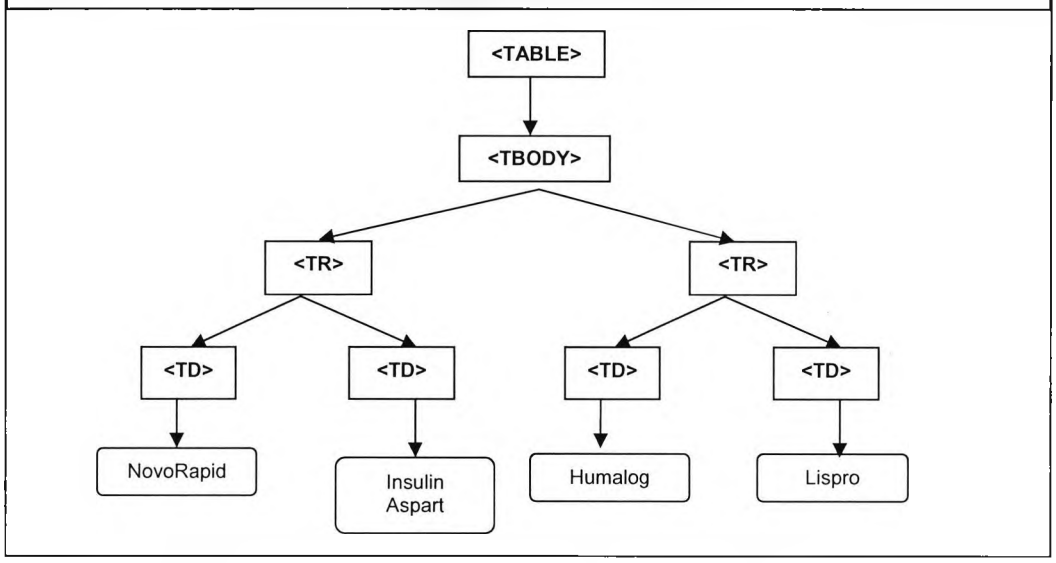


Figure 6.2 *HTML document and its correspondent DOM representation*

A DOM tree groups the content of an HTML document into logical units where each unit contains text that can be viewed as a text zone. The DOM tree is then traversed in a depth-first approach and each text zone in the tree is processed. The java development kit contains several components for processing DOM trees which allow programs to dynamically access and update the content and structure of documents.

6.2.2 Sentence Annotation with MMTx API

The MetaMap Transfer (MMTx) tool is a Java package developed by the National Library of Medicine that maps free text to UMLS metathesaurus concepts. Text is tokenised into sections, sentences, phrases, terms, and words. Noun phrases from

text are mapped to the best matching UMLS concept(s). It should be noted that each UMLS concept has been labelled with one or more UMLS semantic types. These are categories that are defined in the UMLS semantic network. Today there are 135 semantic types and 54 relationships linking these concepts together. In this work, the semantic network is used as an upper level ontology. It has been reported that approximately 99% of all concepts in the UMLS metathesaurus are mapped to these upper level categories. The main relationship between the semantic network concepts is the "isa" relation, a term mostly used in object oriented paradigms that forms a hierarchy between concepts in the form of superclass-subclass hierarchy which is also referred to as inheritance. The other relationships that are defined in the UMLS semantic network are grouped into five categories[16]:

(1) Physically related to.
(2) Spatially related to.
(3) Temporally related to.
(4) Functionally related to.
(5) Conceptually related to.

Figure 6.3 shows an overview of the steps used by MMTx to map text to UMLS semantic network concepts.

First, a document is tokenised into sections, each section is divided into sentences, then each sentence is divided into tokens. Optionally, tokens can be processed with an external Part-Of-Speech (POS) tagger. The tagger can be attached through the programming interface.

[16] Semantic relationships are not tackled in this research work, but are only presented here for completeness

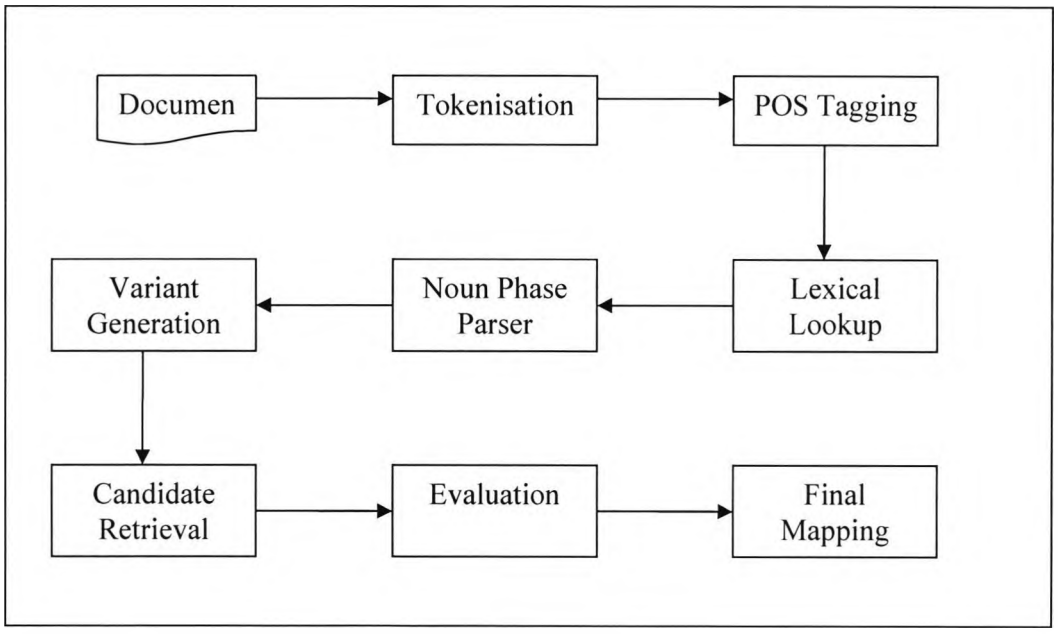


Figure 6.3 *MMTx steps used in the process of text mapping*

The SPECIALIST Lexicon which is part of the UMLS system is used in the next stage, the lexical lookup, where each token in the sentence is tagged with its POS[17]. Next a noun phrase parser is used to tokenise sentences into phrases. A variant generation model is used to derive several variations from each phrase. All variations are used to retrieve candidate UMLS concept strings. Candidates are then evaluated to produce the final best covered concepts.

Figure 6.4 shows one chunk of the resulting output from mapping the string "insulin pump". It can be observed that first the string is marked as a noun phrase. It is also shown that "insulin pump" is a medical device which belongs to the semantic group of physical objects. Each piece of information can be retrieved by calling its own method which is implemented as part of the MMTx API.

As can be observed from the example in Figure 6.4, MMTx can be used to annotate sentences with essential information that is needed for information extraction. MMTx has been developed for the domain of medicine and it is recommended for any application that needs to analyse biomedical text. As mentioned above, MMTx

[17] By default, MMTx uses the MedPost/SKR POS Tagger

is a Java API that can be used in stand-alone applications for biomedical text analysis. It is downloaded as a Java "JAR" file which first needs to be included in the CLASSPATH environment variables.

An object of type "MMTxAPI" is instantiated which contains all necessary methods to process the text. The text can be in the form of a term, sentence, phrase, or even a document. There is a set of parameters that can be set in a configuration file that instructs MMTxAPI how to handle text and what type of information to output. The interested reader can check the NLM website[18] for all the details.

Input:

"Insulin Pump."

Output:

Sentence|17|277|292|Insulin Pumps.

Phrase|43|277|290|Insulin Pumps|Insulin Pumps|2|true|NOUN_PHRASE
1000|C1140609|Insulin pump, device (physical object)|T074|Medical Device|medd
1000|S0293457|Insulin Pumps|insulin pumps

Figure 6.4 *MMTx example for the term "Insulin Pump"*

As a final comment before moving to the next section, it should be noted that MMTx takes care of the lexical, syntactic and semantic analysis phases that are used in many advanced information extraction tasks.

6.2.3 Concept Lookup

Based on the analysis made on the set of diabetes-related websites, three types of information have been identified[19]:

[18] http://mmtx.nlm.nih.gov
[19] The second type of knowledge discussed earlier is now divided into two types

I) Information about the website which includes the following[20]:

1. Title: the name given to the website
2. Creator: the name of person or organisation that made the content of the website.
3. Subject: expressed as a set of keywords that identify the topic (this could be mapped to a classification code from UMLS)
4. Date created: in the form YYYY-MM-DD
5. Date of last update: following the same format described above
6. Identifier: This will be the URL of the website
7. Language: the default is English

II) General information (static) about the disease, such as types, symptoms, complications, diagnosis, etc. This type of information is basically the same and recurred over and over again. Such information is for human consumption and not for the machine. Therefore there is no need for this part to be transformed into formal knowledge. It has been found from analysing the set of websites that this type of information is consuming a large space; finding and skipping this part would increase the efficiency of the system.

III) Information about the purpose of the website (dynamic) which includes the following two categories:

1. Providing services
2. Providing products

Each category shown in the third type (i.e. products and services) has been divided into subcategories. These subcategories are based on the analysis carried out in the previous chapter. The attributes for each subcategory have also been derived and the results were used to construct the ontology. It is important to note that each set

[20] This is a subset taken from the Dublin Core Metadata element set for resource description.

of the related attributes basically makes up one template in the library. The ontology is used to guide the system to the appropriate template to be filled with the relevant information. Templates are modelled as classes (in object-oriented terminology) and the slots of the templates are attributes. In the system implemented here, each attribute is also a class. For example, the class "Product" contains the attribute "Price". The attribute "Price" has been implemented as a class instead of just a primitive data type (i.e. float). This design decision has been made for the following reasons:

1. Class "Price" has a *value* of type float where the amount is stored, the type of *currency* (i.e. £ for UK currency) which is another attribute, a string that contains a "Regular Expression" pattern to be used when searching text for a price, and finally a string contains the *tui* (Semantic Type Unique Identifier) which is used to establish vocabulary control.
2. Each attribute mentioned above has its own methods (i.e. set and get) which are also implemented as part of the class.
3. This makes attributes and methods for classes like "Price" tightly coupled which is more natural from real world point of view.
4. It simplifies the implementation by adapting a "Divide & Conquer" approach.
5. It allows for reusability and extensibility. This means that the class "Price" can be attached to any object where it has price as an attribute (or property in OWL terminology). Also, as a class, it can be extended where other functionality can be added.
6. Following OWL specification where properties are also classes and the attributes are only links. For example, the concept "Product" has a link (i.e. relationship) "has_Price" which links "Product" to "Price".

The same approach has been used with all common attributes, such as "Date", "Address", "Phone", "Email", etc.

At this stage, concepts from the annotated sentences are matched with concepts found in the ontology. This is done by querying the ontology using Protégé-OWL API.

Protégé is an open-source ontology development environment developed in Java by the medical informatics group at Stanford University. It has a flexible, open architecture that allows programmers to develop the so-called plugins. Plugins are used to extend the system's functionality. The Protégé-OWL plugin has been developed, also at Stanford, especially to work with OWL and RDF(s) ontologies. Figure 6.5 shows a screenshot for the main interface.

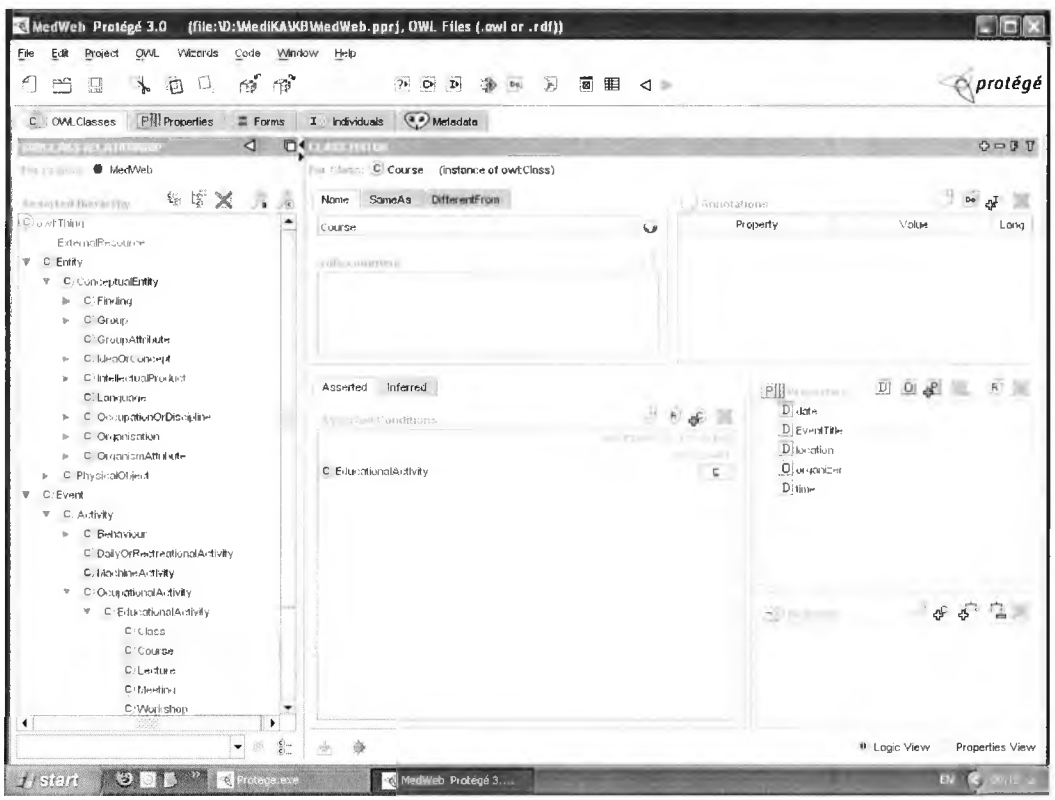


Figure 6.5 *Protégé-OWL main user interface*

It can be noted that it is slightly different from Protégé's main interface (shown in the previous chapter), because it is customised to work with the OWL ontologies which use a somewhat different terminology than the one used in Protégé. For example, Protégé's instances are called individuals in OWL, and OWL's properties

are called slots in Protégé. It should be noted here that OWL's properties are equivalent to 'relations' in object oriented. Also it should be noted that the Protégé-OWL has new facilities and a richer set of operators tailored to OWL ontologies. Probably one of the most important facilities in Protégé-OWL is the possibility to perform reasoning, by setting up a reasoner such as "Racer" for computing subsumption relationships and validating the ontology.

Protégé-OWL provides programmers with an open-source Java library in the form of an Application Programming Interface (API). The API contains classes and methods to load and save OWL files, manipulate and query OWL ontologies, and perform reasoning on the ontology. Best of all, the API hides the low-level details from programmers which are abstracted into interfaces. Programmers need only to understand how to use the set of interfaces with which they can perform just about anything with the ontology. Basically, they provide everything needed for the system under study.

The application ontology has been created using the Protégé-OWL interface. The ontology is saved in OWL format. This means that a knowledge engineer can view, modify, and update the ontology anytime using the Protégé-OWL interface as shown in Figure 6.1.

The system accesses the ontology using the API by first reading the OWL file that is saved on disk and creates an OWL model called "OWLmodel". The object contains all the methods that are needed to access the classes and properties saved in the ontology. While the OWL model has a complete set of methods to add, delete, and modify the ontology, the system explained here only needs to query and traverse the content of the ontology looking for certain classes and properties that match the tagged free text.

The algorithm used for matching concepts from the tagged sentences against the ontology is as follows:

1. Read a line from the set of tagged phrases.
2. Tokenise the line using "|" as a delimiter.
3. If the number of tokens in a line is equal to six, retrieve the fifth token.
4. Query the ontology with the token.
5. If it exists, store the concept in the relevant list of concepts.
6. Store the phrase in the relevant list of phrases.
7. Read the next line from the set of tagged phrases.

It has been observed from the list of tagged phrases which is the output of MMTx, that lines containing concepts have six tokens. The fifth token is the concept name. This observation made extracting concepts much easier to deal with, rather than processing every token looking for concepts.

The new and novel approach of matching concepts, rather than terms, was extremely useful in the context of this work. It is impossible to enumerate all terms that are encountered in HTML documents. Once a set of relevant phrases has been determined, the system moves to the next phase, the extraction of instances, to be presented in the next section.

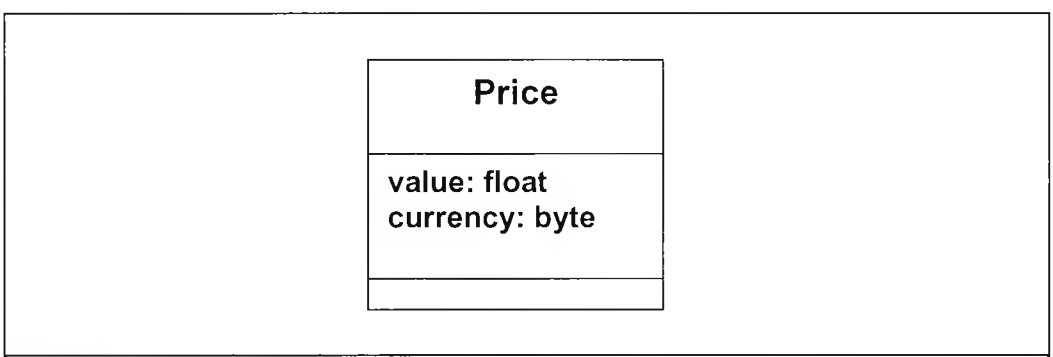
6.2.4 Extraction of Instances

The utilisation of object oriented methodologies in the analysis and design of this work has paid off in many ways. For one thing, implementing templates as objects rather than flat data structures has been extremely helpful in the organisation of the different data items and providing a more natural way (i.e. closer to real-world objects) for implementing such entities. The process of extracting instances has also benefits from the adaptation of object oriented methodology. Once a concept has been located in a webpage and identified as explained in the previous section, an object of that concept is instantiated and the values which were previously obtained from the MMTx-tagged phrases are filled in. This step is known in information extraction as merging, where all the fields that were discovered in the previous steps are now merged together.

Probably the greatest advantage of such an approach, besides perspicuity, is reusability. Once these common objects are fully developed they can become part of a library for information extraction that can be utilised for other domains as well.

6.2.5 Output Structure Format

Another advantage of using object oriented methodologies in this work is the development of the method "toOWL( )" that outputs the instantiated object in OWL format. This has been inspired by the Java built-in method "toString( )" that simply transforms java objects to strings that can be displayed on the screen. This method has been inherited from the superclass and down the hierarchy to every class in the model. Method "toOWL ( )" does the same thing where the data held inside the object are transformed into OWL constructs. For example, refer to Figure 6.6 to see a simple form of class "Price" with two attributes.




```
<owl:Class rdf:ID="Price"/>
 <owl:FunctionalProperty rdf:ID="value">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
   <rdfs:domain rdf:resource="#Price"/>
 </owl:FunctionalProperty>
 <owl:FunctionalProperty rdf:ID="currency">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#byte"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
   <rdfs:domain rdf:resource="#Price"/>
 </owl:FunctionalProperty>
```

Figure 6.6 *UML class Price with its correspondent OWL format*

The first one is "value" of type float and the second one is "currency" of type byte. The upper part of the figure is showing the class in UML format and the bottom part is showing the class in OWL format. Method "toOWL( )" utilises the java StringBuffer object which is used to wrap OWL constructs into strings as shown in the bottom part of Figure 6.6. The constructed string can then be either printed, saved to disk, or sent to another process.

6.3 The Data Model

During the initial study, it has been found that many concepts exhibit common attributes like all products have attributes such as price and brand name. However, some products have different attributes to others and some products have more attributes to others too. For example, a dietary food item has a different set of attributes than an insulin pump, and a glucose meter solution has many fewer attributes than a blood glucose meter.

Special Excel spreadsheets have been designed for this work to aid in collecting information about each product type. The same collection mechanism has been used with all services that have been encountered too. The collected information has been used in the building of the ontology. As has been mentioned before, the ontology basically contains several hierarchies of concepts and the attributes for each concept which are normally represented as a relationship linked to another concept. Concepts in the ontology have been mapped into java classes. Classes form the main data structures for any object-oriented system.

In the object oriented paradigm a product is an object, and all products are classified under some category which is organised in a class hierarchy. Figure 6.7 shows part of the product hierarchy which has been developed for this project. It should be noted in Figure 6.7 the root node contains the most general class. This is class "Product" in this example. Moving down the hierarchy, classes become extended with other attributes and methods and some methods that are inherited from the superclass may be overridden (i.e. their code is replaced with a more specialised

code). This process is called specialisation in object-oriented terminology, which is considered one of the main features of Object-Orientation.

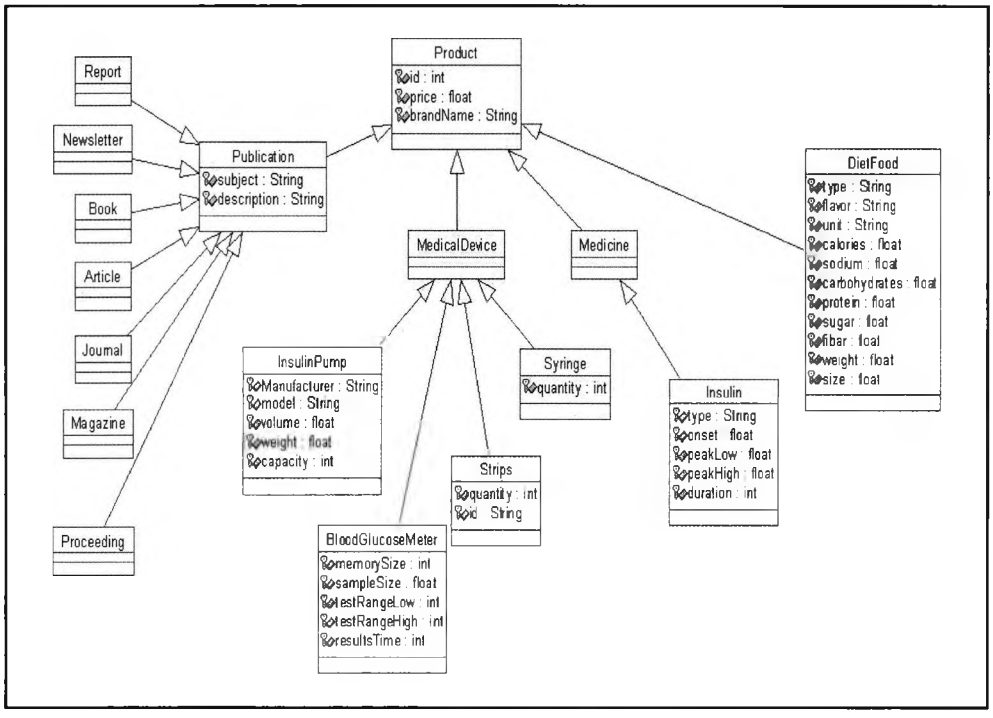


Figure 6.7 *UML class diagram showing part of the product hierarchy*

In the same manner all events such as workshops, conferences, classes, and seminars, which are disseminated on many chronic disease-related websites, also exhibit common attributes such as name of event, date, time, purpose, etc. Figure 6.8 shows class event. Class event is part of the service class hierarchy.

In Figure 6.8, two types of relations are shown: the first one is inheritance which was explained above, and the second one is composition. An example is the relationship between class Event and class Time, which is a part-whole relationship. In other words, class Event has an attribute of "object type" and in this case of Time. In OWL, most properties (attributes) are of this type.

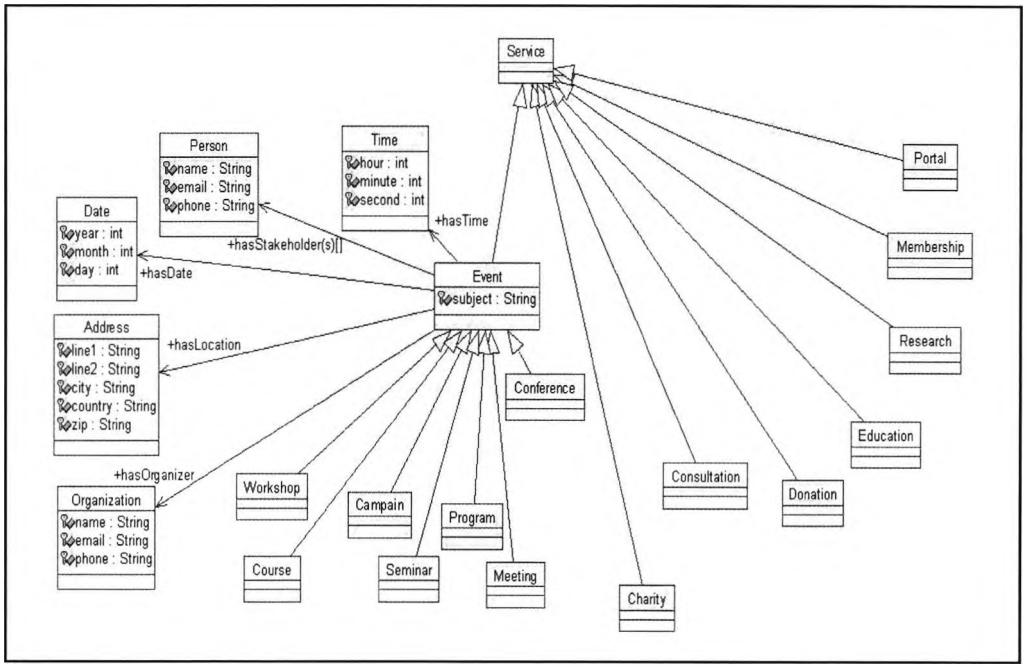


Figure 6.8 *UML class diagram showing the class hierarchy for Service*

6.4 System Menu Design

During the first part of this research project, i.e. the website analysis phase, a java application was built for this purpose. Figure 6.9 shows the user interface of this system which was extended later to accommodate the IE system. The left panel displays the list of downloaded websites. When a user clicks on a folder representing a website, the view is expanded to show subfolders and HTML files in a tree structure. Upon clicking on an HTML file, its content gets displayed in the right panel which works as a browser that is used for visual examination of the file's content. The file size and other information are displayed at the bottom in the status bar.

On top of the frame a menu toolbar is displayed. Choosing the first item "File" allows a drop-down submenu to be displayed where the user can choose to open, save, and print opened files.

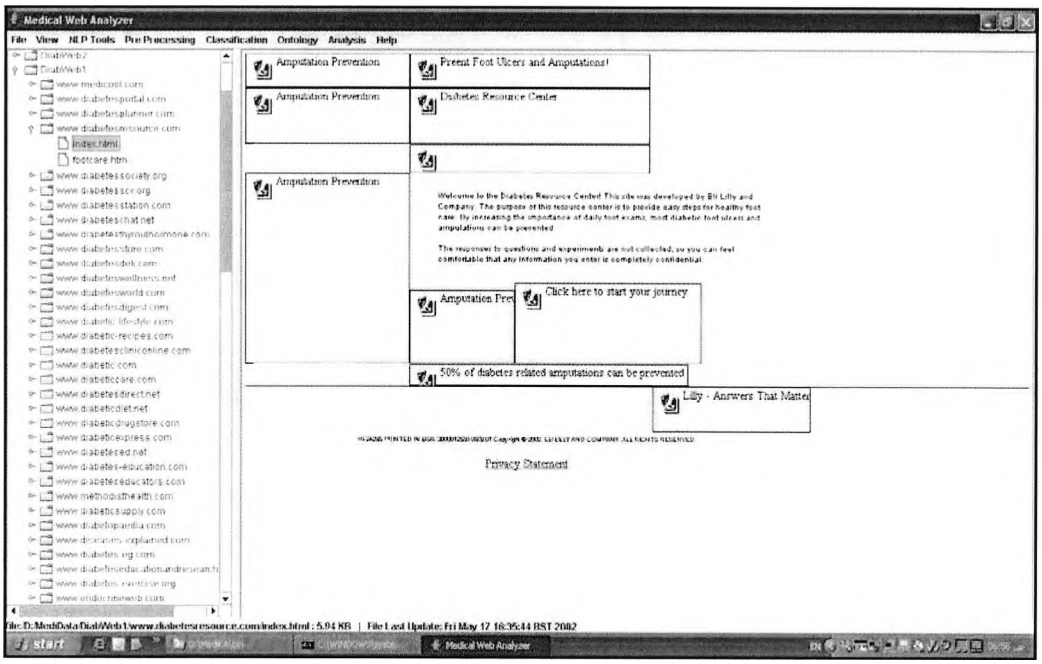


Figure 6.9 *System screenshot*

The second choice on the main menu is the "View" menu which has the following choices:

1. Source: Display the source of the opened HTML file for inspection of features such as the content of the "Meta" tag.
2. Tags: Extract and display all tags in the opened HTML file.
3. Links: Extract and display the links contained in the opened HTML file.
4. Text: Extract and display all Text blocks in the opened HTML file. This is done by traversing the HTML graph grouping and concatenating all text segments into the buffer, then displaying the content of the whole buffer on the screen.
5. Stemming: This feature is an implementation of the Porter stemming algorithm which returns each token to its origin. For example, the word "Played" is changed to "Play". This is an essential feature for comparing different words having the same stem.
6. Tokens: Divide text into tokens which are then displayed in the right side panel.

7. Stopwords: remove all Stopwords from the text. Stopwords are common words that contribute no special meaning to the text such as "the" and "is". A file with around 500 Stopwords has been prepared and used to remove the Stopwords from text. Usually words left after removing Stopwords are words that are considered feature candidates.
8. CountWords: This module counts the number of occurrences of each word in the text (frequency), which can be used to determine the importance of words in the text (See Figure 6.10).

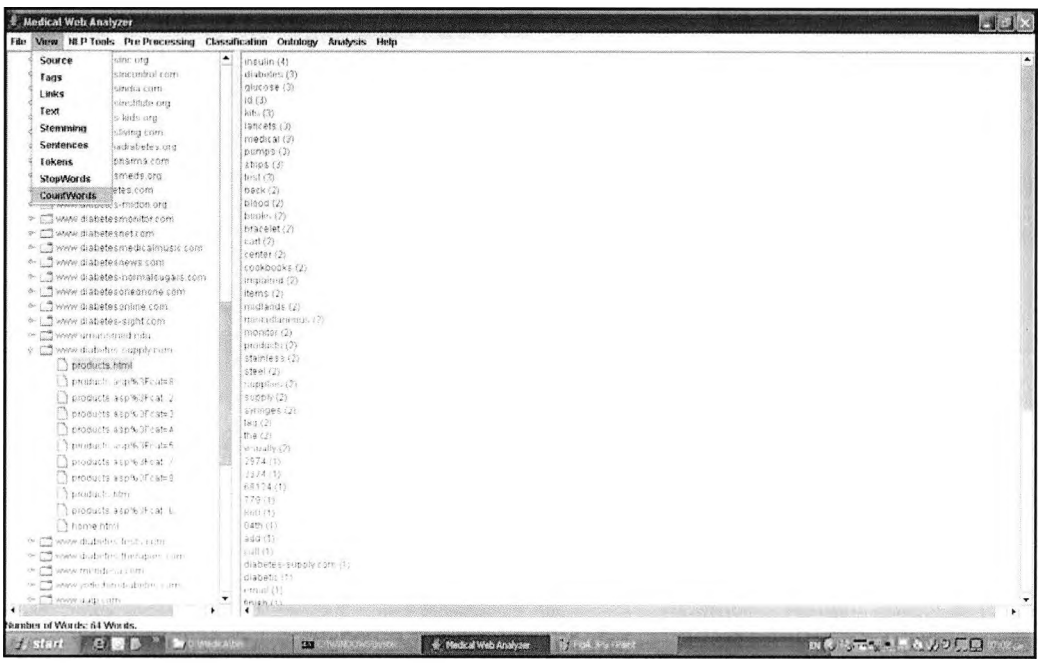


Figure 6.10 *Output showing the list of terms and their counts*

During the design phase, other modules were added and tested in turn. For example, the next submenu is "NLP Tools" which consists of a POS Tagger, a parser, and an NE recogniser that are needed for natural language processing. These are open source modules that were also developed at Stanford and are being reused in this research project.

The next item in the menu is "Pre-processing". It contains the modules for pre-processing the HTML page such as extracting prices, dates, times, emails, etc. These items are extracted using regular expressions that have been developed for

this purpose. The next item in the menu is "Classification". Module "MMTx" can be called explicitly for performing syntactic and semantic analysis on the text. The right side panel in Figure 6.11 shows an example run of this module.

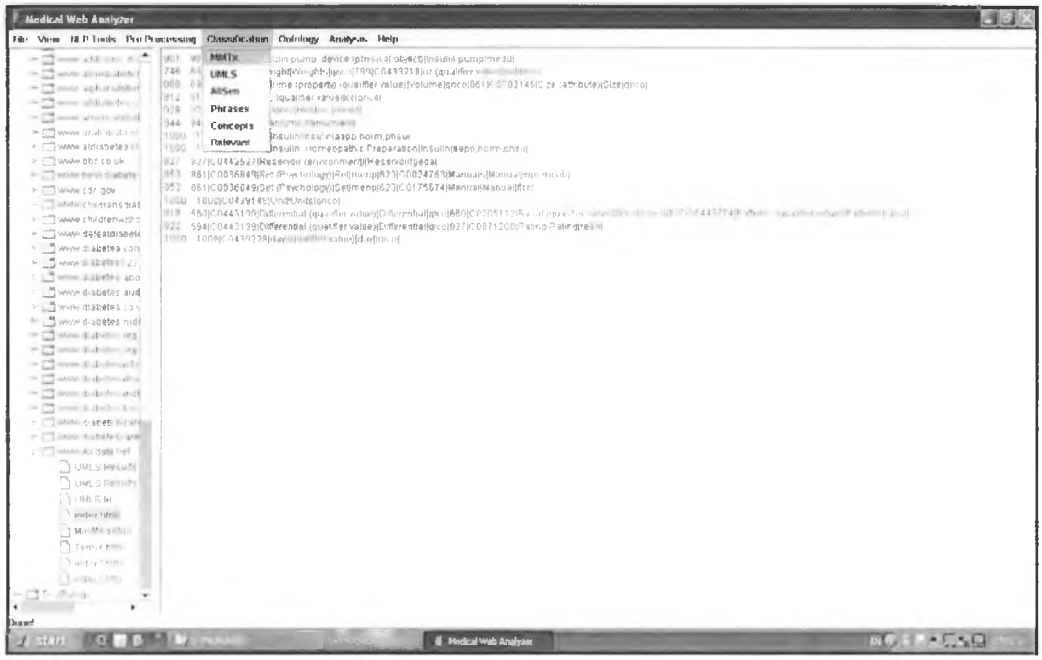


Figure 6.11 *Output of MMTx after processing HTML document*

The module "UMLS" displays more information and a very detailed analysis. The module "AllSen" extracts all sentences from the HTML document. The module "Phrases" extracts the list of phrases. Figure 6.12 shows a list of extracted phrases along with other information such as the type of phrase (i.e. Noun Phrase, Verb Phrase, etc). The next module is "Concepts" which displays all items on the page after classification. The last item under this menu is "Relevant" which extracts the relevant concepts by utilising the ontology and displaying the pairs as shown in Figure 6.13. The algorithm for extracting relevant pairs has been explained in the previous section.

It should be emphasised here that this system interface has been designed simply for analysing websites and for testing the various components of the framework. A friendlier user interface would be needed in the event that this system was to be fully developed.

Figure 6.12 *Output showing extracted phrases*

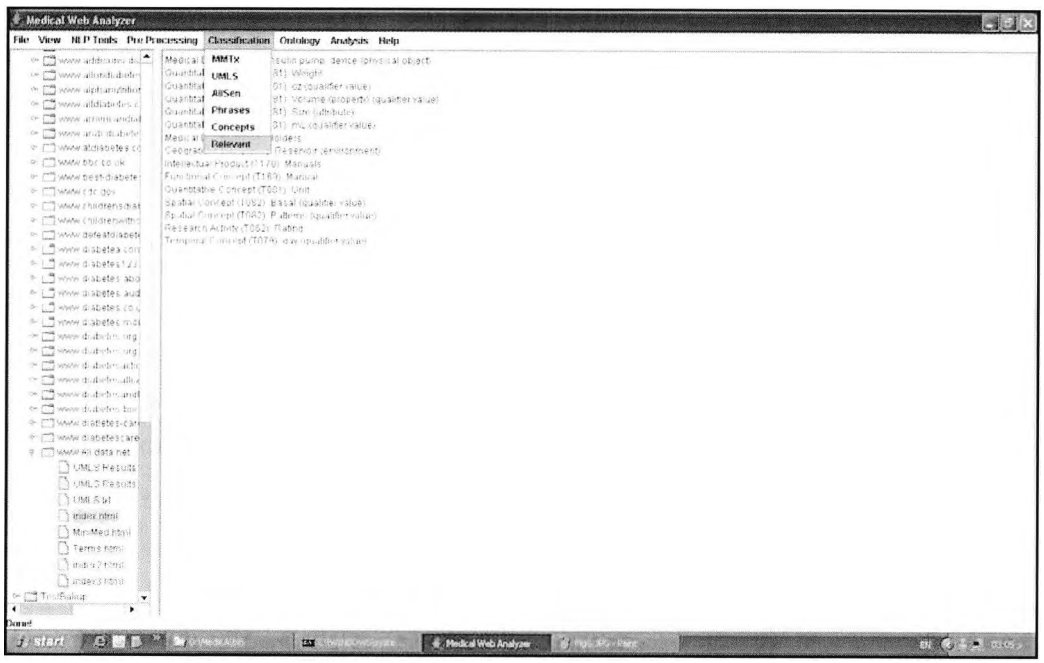


Figure 6.13 *Output showing relevant pairs*

The next menu item is "Ontology" where the system can interact with the ontology via the protégé-OWL API to display classes, properties, and instances. Figure 6.14 displays the list of classes in the ontology.

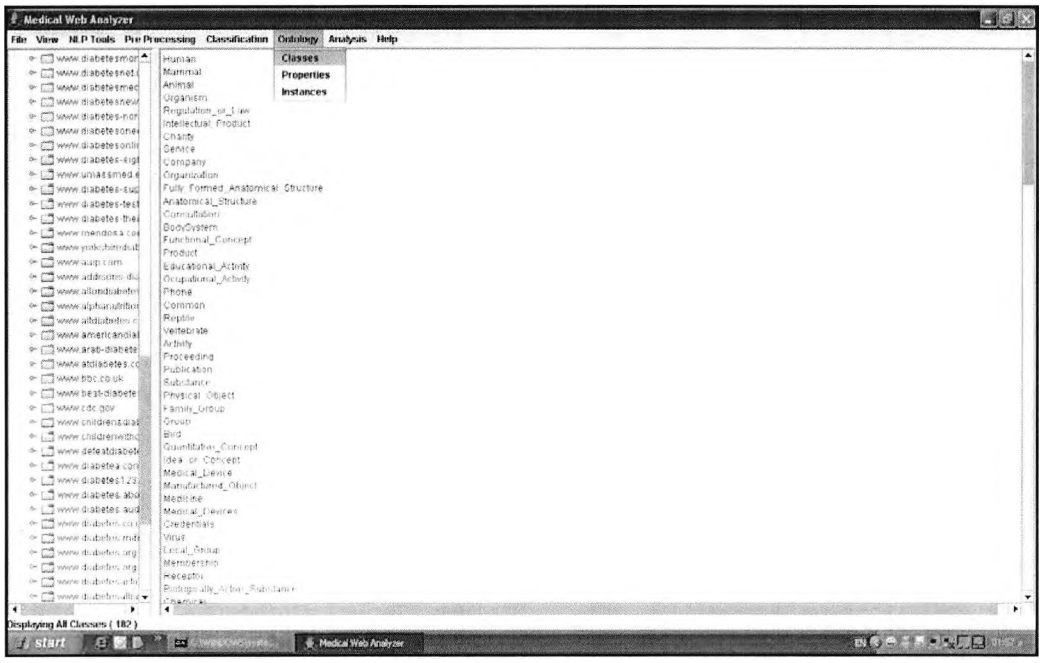


Figure 6.14 The *list of classes outputted by interacting with Protégé-OWL API*

The next item is "Analysis", where statistical and conceptual analyses are carried out on the web page to determine the category. Figure 6.15 shows the output of a conceptual analysis where concepts found on the page are compared with concepts from the ontology.

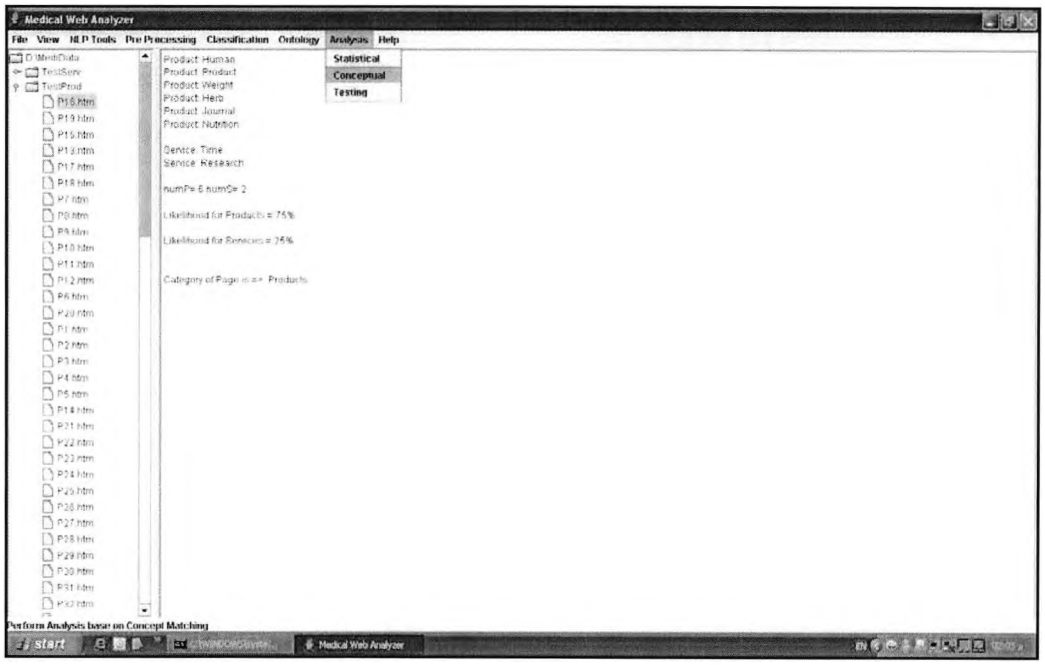


Figure 6.15 *Output from the conceptual analysis for determining the category*

6.5 Conclusion

In this chapter, the development and implementation of an object oriented information extraction system has been presented in detail. All inputs and outputs of the system have been discussed. All the components of the system have been shown. How these components interact with each other to accomplish their tasks has also been presented.

The system has served two purposes. The first one is to aid in the analysis of the chronic disease-related websites that were collected from the web. The second one is to provide a test bed for extracting and structuring relevant information that can be used very constructively in populating the semantic web.

It may be concluded that knowing the purpose of the website can be very effective in determining how to process it. Knowing what type of information is presented in a webpage helps in determining what attributes to look for and therefore guides the extraction and structuring processes.

In the next chapter the methods that have been used in testing the system will be presented. This will be followed by a discussion of the results.

Chapter Seven

Experimental Testing

"Are the laws of acceleration and of the composition of forces only arbitrary conventions? Conventions, yes; arbitrary, no--they would be so if we lost sight of the experiments which led the founders of the science to adopt them, and which, imperfect as they were, were sufficient to justify their adoption"

- Henri Poincaré, *1905*

7.1 Introduction

In the previous chapter, the system under study has been discussed in terms of design and development based on the objectives that were set for this work. The methodology used in building the system is an iterative waterfall approach whereby the building process is made up of a number of stages; these are requirements specifications, design, implementation, and testing. Upon the completion of each stage, the previous stages are reviewed and modifications are made to achieve better results. This process has been carried out in incremental steps. Tasks are added until the current system has been completed. The objective of this approach is to achieve robust and reliable results. During the building process a variety of test cases have been carried out in an iterative fashion.

In this chapter the experiments that have been carried out will be presented in detail with analysis of the results being discussed in the next chapter.

7.2 Meeting the Objectives

The aim of the experiments is to evaluate how well the resulting system behaves under different circumstances in meeting the thesis set for this work. More specifically, what does the system extract from different samples of HTML documents? This can be explained in terms of input/output description as follows:

Input:

HTML document related to chronic disease.

Output:

Attribute-value pairs of the *relevant* text based on the classification of the HTML document (i.e. service or product).

Two sets of sample web documents have been randomly collected from the web for evaluation. The two sets contain examples where their answers are known in advance and they will be used to determine the accuracy of the system.

The content of these two sets are as follows:

1. Product-related web documents:

These documents have been extracted from different chronic disease-related web sites and contain many different types of related products such as insulin pumps, meters, herbs, books, etc.

2. Service-related web documents:

These documents have been extracted also from different chronic disease-related websites and contain specifically events such as announcements for workshops, classes, seminars, etc.

The output explained above can be seen as labelling specific values with the correct attributes. Thus, it is actually a classification problem in two phases:

1. The classification of the web document, for example: does it contain a product or a service provided by a chronic-disease related website?
2. The classification of values in the relevant text as to what attributes they belong to.

The performance of the system depends on two main factors:

1. The performance of the MMTx tool and how well it behaves with different samples, taking into consideration that the MMTx tool is one of the main components of the system under study.
2. The content of the Ontology that guides the system as to what to extract from a given text. This is another main component as explained in the previous chapter.

Both factors have been subjected to evaluation and are explained in more detail in the next sections. Some standard evaluation techniques that are used in IE systems have been carried out. These are explained in the following section.

7.3 Experimentation with MMTx

The system as explained in the previous chapter is mainly made up of several "open-source" components. Such components have been used reliably and successfully in many research projects. One of these components is the MMTx API. The API is made up of several java classes where each class is designed to carry out certain tasks. The architecture of the MMTx API has been discussed in detail in the previous chapter. However, it should be mentioned here that the primary goal of the MMTx system is to process and analyse text by mapping text to concepts from the UMLS Semantic Network. In other terms, the tool classifies each phrase and assigns a category from the list of predefined upper categories of the semantic

network. Without this tool, the process of classifying the various pieces of text found in web pages would have been very difficult to achieve. In chapter three another tool has been experimented with, namely "WordNet", which also can be used to map text to concepts. Both WordNet and MMTx have been subjected to evaluation. The results are discussed in the next chapter.

7.4 Experimenting with the Ontology

Protégé-OWL has been used in this project first to maintain and edit the ontology, and secondly to interact with the ontology programmatically. This tool is designed to make the process of working with the ontology easy and efficient. Protégé-OWL has been reviewed in chapter three in detail.

The ontology plays a major role in this work, therefore it is important to ensure that a high utilisation is achievable. There are two criteria that need to be checked:

1. Validity of the Ontology

An excellent facility provided by Protégé-OWL is that it allows for the installation and the use of an external DIG compatible reasoner. In this work a reasoner called RACER (Haarslev & Möller, 2001) has been installed and used for testing the ontology throughout the development phase. Racer provides two main services:

a) Subsumption testing:
   To test whether or not one class is a subclass of another class.
b) Consistency Checking:
   Based on the conditions of a class it can be checked as to whether or not it is possible for the class to have any instances.

2. Correctness of the Ontology

Correctness of the ontology can be achieved by experimenting with the IE System to see if the results have been satisfactory. This will be explained in more detail in the next section.

7.5 Experimentation and Evaluation of the IE System

Evaluations of IE systems are performed using the same measures used with information retrieval (IR) systems. However, the difference is that IR systems are evaluated based on the determination of the relevance to the query, while IE systems are evaluated based on the determination of the relevance to the template to be filled by the system.

In IR, Recall (R) is the percentage of relevant documents that have been retrieved by the system in relation to all relevant documents in the collection. Precision (P) is the fraction of documents retrieved by the system that are relevant to the query. These two measures are often combined to give the F measure:

$$F = \frac{(\beta^2+1)PR}{(\beta^2 P+R)} \qquad (7.1)$$

where β indicates a relative value of precision versus recall. Based on the above discussion, Precision and Recall can be given in the following equations:

Precision = The set of correctly predicted instances / The set of predicted instances

Recall = The set of correctly predicted instances / The set of all true instances

Two subsets of HTML pages are manually extracted from chronic disease-related websites. Each subset contains pre-determined instances that cover one the predefined categories as explained in section 7.2 above.

Each set is processed by the system. The output is then examined. Results are presented in the next chapter.

7.6 The Sets of Asthma and Hypertension – related Websites

Two more sets of websites were downloaded. One set contains asthma-related websites, and the other set is for hypertension-related websites. A similar analysis to the one that was carried out on the set of diabetes-related websites has been performed. The goal was to find common structures among websites related to chronic diseases in general, and also to see whether such generalisations could be applied on the ontology.

The two subsets have undergone similar analysis to the one performed on the diabetes-related websites. The results of the three analyses are then compared and presented in chapter 8.

7.7 Conclusion

The research that has been carried out in this project has two phases. The first one has involved studying several chronic disease-related websites. The reason was to extract the common concepts. These concepts were then entered into an ontology. The second phase consisted of building an IE system that utilises the ontology for extracting information from unseen websites. In this chapter the experiments that have been carried out have been explained. In the next chapter results from these experiments will be presented in detail.

Chapter Eight

Analysis of Results and Assessment of the System

"An approximate answer to the right question is worth a good deal more than an exact answer to an approximate question."

- J. W. Tukey, 1962

8.1 Introduction

In the last chapter it has been explained that the system has undergone a range of experimental testing. There are two main tasks that have been carried out during the previous phase; the first one is related to the analysis of websites where the results are discussed in the next section. The second is the work that has been done on information extraction in which the results from experiments that were carried out are discussed in section 8.3. Finally, some concluding remarks will be presented in the last section.

8.2 Websites Analysis

In the first part of this work a set of 100 diabetes-related websites has been collected from the Internet, and has been carefully analysed. The process of selecting the websites was made by collecting links, mostly from the first few pages returned by the Google search engine. Google ranks websites based on popularity; people looking for information about diabetes are most likely to consult one of these websites. This makes them good candidates for the analysis. The main categories found in these websites were collected and recorded in spreadsheets specially designed for this purpose. Table 8.1 shows the main categories with some examples.

Categories are usually pointed out in menus found on the main webpage. Each category has been studied. Concepts and attributes found in each category are collected and recorded. Based on the analysis of these websites it may be asserted that every chronic disease-related website has a purpose mainly providing services or products (or both). Services offered by these websites can be classified based on these categories such as education, research, portals, local and global support through memberships, etc.

Purpose Category	Examples
Products	Dietary food, Medicine, Medical devices such as meters
Education	Organise seminars and conferences Provide publications such as articles and books Hold classes and workshops Provide assistance for educators
Membership	Subscription to newsletters Participation in special programmes and activities Aid in disease control and management
General Information	Symptoms, complications, treatments
Research	Research involvement, research support
Portal	An organised hierarchy of links to other related websites
Online consultation	E-mail, online chatting, online forums
Others	Recipes, news, accept charities and donations, organise campaigns, service local groups

Table 8.1 *Main categories for purposes found on the websites*

In the same manner, products can also be classified into main categories as shown in Table 8.2, such as nutrition, skin care products, publications, sugar free food, medical devices, etc.

Two more sets of websites have been downloaded and analysed in the same way. The first one is for asthma and the second one is for hypertension. The extracted concepts from all three sets have been compared and analysed. The intention has been to find common concepts and their attributes among chronic disease-related websites in general. It has been found that most of these websites exhibit many common structures as will be presented shortly.

Knowing the purpose of a website gives a good hint as to what type of information to expect. For example, if a website's purpose is to sell books then it is expected to find a list of names of books along with other information such as ISBN numbers, name of author(s), prices, etc. The study resulted in the construction of an ontology covering a set of services and products that these websites offer.

Product types	Examples
Special foods	Sugar free food, nutrition, herbs, vitamins, minerals
Medical devices	Glucose monitoring meters, strips, lancets, insulin syringes, insulin pumps
Medicines	Insulin, glotrole, skin care
Publications	Books, magazines, special articles
Software	Disease management, research support
Others	Travel kits

Table 8.2 *Categories for product types with examples*

Most websites have more than one category, for example some websites are educational, provide general information about the disease, and accept donations at the same time. Figure 8.1 shows the percentage of websites for each chronic disease and the features presented in tables above. The abbreviations used in Figure 8.1 are explained in Table 8.3.

From this figure it has been noted that approximately 70% of all websites present information about each disease in general. This type of information is basically the same and recurred over and over again. Such information is usually not the real purpose of the websites. Therefore, there is no need for this part to be transformed into formal knowledge since it is made only for human consumption and for people who need general information about the disease.

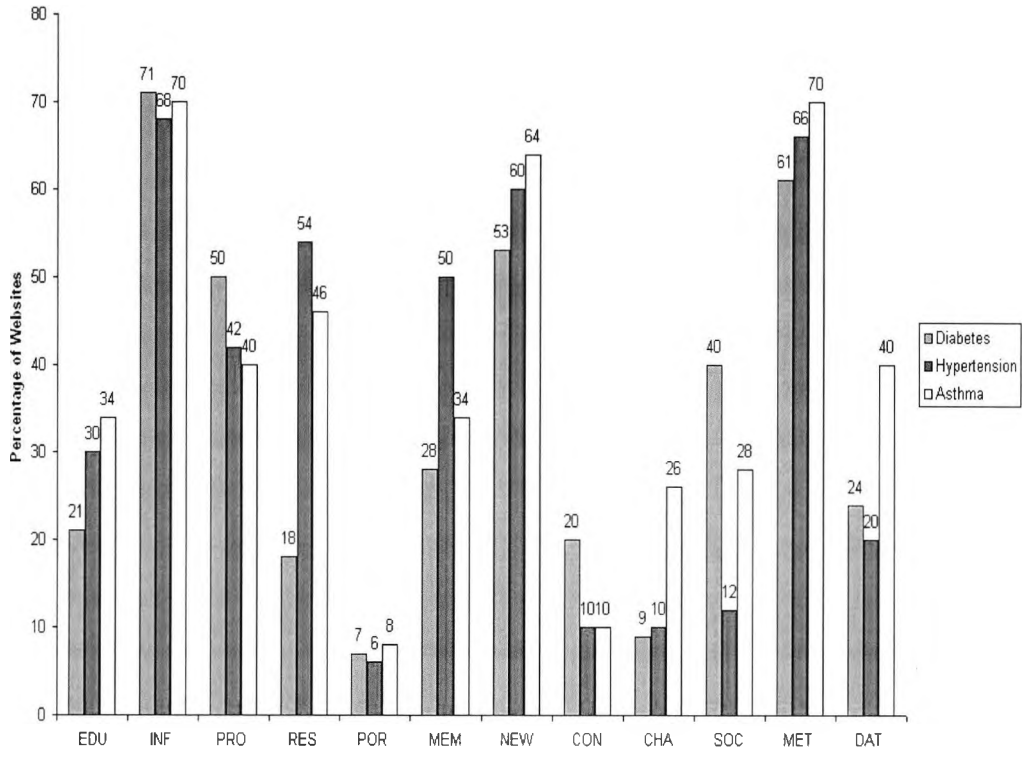


Figure 8.1 *Comparisons of features among the three sets*

Most features when compared between the three diseases are near in values of website percentages. This suggests that related websites exhibit common structures. Some of the variants between values of the same feature are justifiable, for example it seems that more research activities have been carried out with hypertension, a higher percentage of websites offers diabetes-related products, and more diabetes-related societies exist.

ABBREVIATION	EXPLANATION
EDU	Education
INF	Information
PRO	Products
RES	Research
POR	Portal
MEM	Membership
NEW	News
CON	Consultation
CHA	Chatting
SOC	Society
MET	Meta
DAT	Date

Table 8.3 *The list of abbreviations used in figure 8.1 with explanation*

The list of product types in the three sets has been studied extensively. The types were generalised into four main categories that are common to all; these are publications (Pub), medical devices (MD), medicine (Med), and nutrition (NH). Other products are not common between the three. For example, asthma has products concerned with the environment such as air purifiers, mould treatment, and allergy bedding. Diabetes also has some products like sugar free food and skin care products. Figure 8.2 shows the percentage of websites carrying the common products. Publications have close percentages between the three. Hypertension-related medicine has a higher percentage than the others, but for fewer medical devices, and diabetes has more nutrition products than the other two.

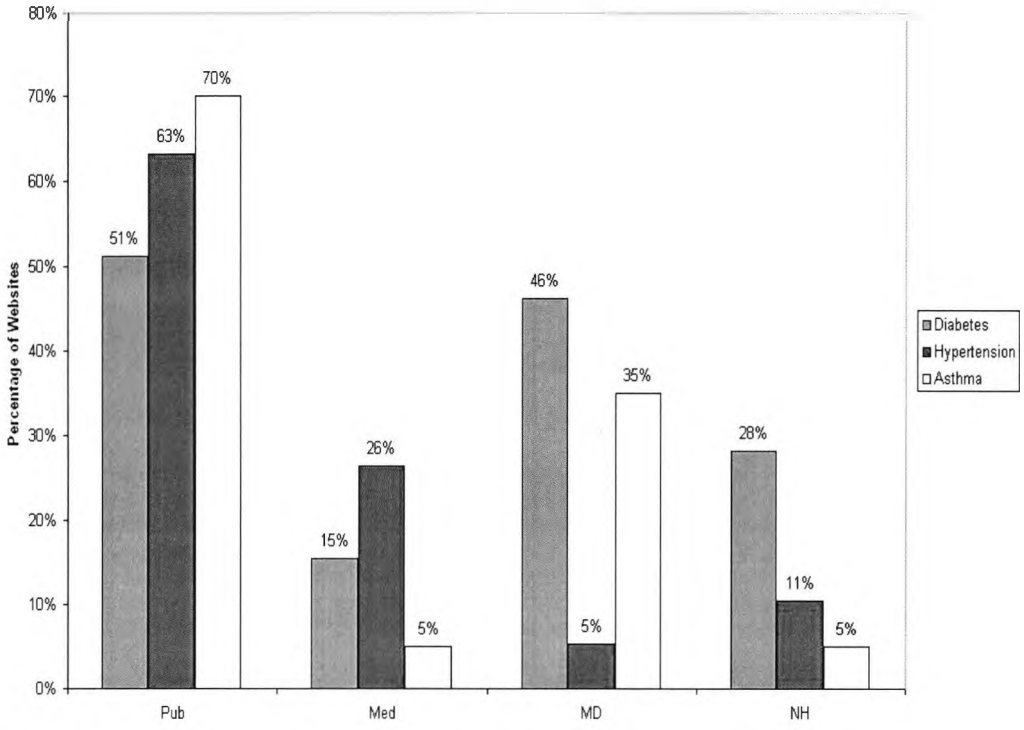


Figure 8.2 *Comparisons of features related to products*

Figure 8.3 shows a comparison of some features found in education-related websites. These features are activities like workshops and conferences, publications like books and journals, support groups, stakeholders such as professionals and patients. Some offer memberships, and some hold research-related activities. It can be observed that the values in each group are comparatively close. Table 8.4 explains the abbreviations used in this figure.

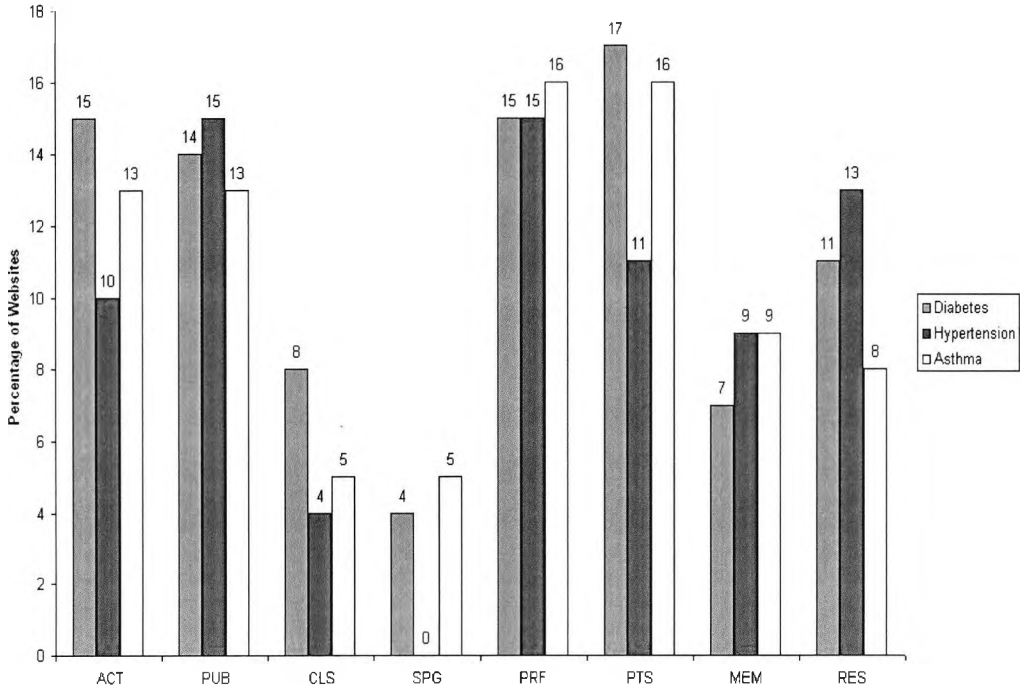


Figure 8.3 *Comparisons of features related to education among the three sets*

ABBREVIATION	EXPLANATION
ACT	Activities
PUB	Publications
CLS	Classes
SPG	Support Groups
PRF	Professional
PTS	Patients
MEM	Memberships
RES	Research

Table 8.4 *The list of abbreviations used in Figure 8.3 with explanations*

Based on this study, the types of information found on chronic disease-related websites can now be grouped into three categories as follows:

1. Information about the website: This includes the title of the website, its IP address, date of last update, etc. This category has been discussed in chapter two under "Website-About Knowledge".
2. Static information: This is general information about the diseases such as definitions, types, symptoms, and complications and is repeated in 70% of all three sets.
3. Dynamic information: This represents the real purpose of the website and has been separated into two main categories, namely products and services which have been discussed in detail.

8.3 IE System Analysis

IE systems are made up of several components that are used to process documents. These are:

1) Tokenisation - zoning.

2) Lexical analysis - part of speech tagging.

3) Syntactic Parsing - shallow parsing.

4) Semantic analysis.

The system uses the MMTx tool for tokenisation and annotating text with lexical and semantic information as detailed in chapter 6. In order to evaluate the performance of the MMTx tool in this work, first it has been compared with WordNet. WordNet is defined as a machine-readable lexical database that organises words into syntactic categories. In a previous experiment, 80 diabetes-related terms

have been tested with both WordNet and MMTx (shown in Table 3.2). These terms were categorised into two types:

1. Directly-related terms (i.e. hyperglycaemia, HbA1C).
2. Indirectly-related terms (i.e. autoimmune, obesity).

The semantic types of these terms were retrieved from WordNet and UMLS. Approximately 50% of the directly-related terms and 25% of the indirectly-related terms were not found in WordNet, but all 80 terms have been mapped to the correct concepts by the MMTx tool. More discussion on this issue has been presented in chapter 3.

MMTx has been put through extensive experimentation to ensure correct results with the overall IE system. The following problems were found during this experimentation:

1) **Misclassified abbreviations:**

During the experimentation it has been found that MMTx misclassifies some abbreviations, for example in the sentence "This insulin pump can hold up to 300 u of insulin." MMTx classified the abbreviation 'u' as "Uranium" when it should be "Unit".

2) **Skipping numbers and dates**:

Also, in the above sentence, it has been noticed that the number "300" has been skipped and no classification was given in the output. In another test that contains dates, it also has been found that the dates were skipped.

3) **Missing named entities:**

MMTx provides excellent facilities for processing and mapping health-related terms to concepts. However, it falls short with other types of named entities, such as persons' names and locations. Although this may not be needed when extracting product information, it is required when extracting information about activities such as conferences and workshops.

A workaround for the first and second problems has been implemented. This is done by pre-scanning the text before passing it to MMTx. In the pre-scanning phase abbreviations are identified and looked up from a table stored in a pre-prepared file. Also, during the pre-scanning phase, numbers and other patterns such as currencies (i.e. prices), dates, times, phone numbers, and emails are identified using regular expressions. The third problem remains a task for future work. However, as an initial step, some open source modules, such as the Stanford named entity recogniser has been plugged in and tested for future experimentation.

8.4 Empirical Evaluation

During the experimentation phase, two sets of HTML documents with the answers known in advance were prepared for testing and evaluation. The sets contained products and educational activities collected from the web and the three chronic disease-related samples. The Products sample contained Special Foods (SF), Medical Devices (MD), Medicines (MED), Publications (PUB), and Others (OTH). The system was able to give correct classification of the categories on 93% of all HTML documents in the input sets This means that the system gave an answer of unknown (UNK) only to 7% of these documents.

		Actual				
		SF	MD	MED	PUB	OTH
Predicted	SF	**14**	1	0	1	0
	MD	0	**21**	2	0	0
	MED	0	1	**23**	1	1
	PUB	2	5	0	**19**	0
	OTH	0	4	5	2	**23**
	UNK	0	3	3	1	0

Table 8.5 *The actual and predicted categories produced by the system*

Table 8.5 shows the actual number of each category (diagonal) in bold. For example, in the sample set there are 14 pages containing special foods, 21 pages containing medical devices, etc. These add up to a total of 100 pages. The same table also shows the predictions made by the system. For example, under the MD column there is one page predicted incorrectly as SF, 21 predicted correctly as MD, one predicted incorrectly as MED, five pages predicted incorrectly as PUB, etc. The total number of pages that were classified correctly is 68 out of 100. which means that the accuracy (Precision) is 68%. Looking back at the IE systems discussed in section 4.3, their accuracy ranged from 23% to 72% which indicates that our system competes very well with the state-of-the-art systems.

8.5 Conclusion

The study has shown that chronic disease related websites exhibit many common concepts which resulted in the construction of the ontology to guide in extracting information for new unseen websites. Also, the study has resulted in the development of a platform for information extraction that utilises the ontology. In the next chapter, a detailed discussion about a number of issues that have emerged from the project will be presented .

Chapter Nine

Discussion

"Every piece of the puzzle that doesn't fit gets you closer to the answer."

- Cynthia Copeland Lewis, 1994

9.1 Introduction

The subject of this work has been the development of a novel methodology for populating the semantic web with common knowledge found on chronic disease-related websites. The idea has been to maximise the amount of useful information[21] that can be extracted from the current web. The extracted information is then classified and structured into nuggets of knowledge ready for the semantic web.

During the course of this work three sets of chronic disease-related websites have been downloaded, studied, and analysed. The results of this study were reported in the last chapter. Based on these results, an ontology has been constructed that contains the most common structures that were found on these websites. An IE system has been developed that utilises the ontology for guidance as to what information to extract from a given webpage. The system utilises an outstanding array of open source research components such as MMTx and Protégé-OWL, which have been used in many successful research projects around the world. The system has gone through experimental testing with the results being presented in the previous chapter.

[21] Useful information such as products and services that can be of great benefits to the stakeholders

In this chapter, the need for the right information for patients with chronic diseases has been elaborated, followed by some scenarios of how the semantic web can be utilised to retrieve useful and precise information by stakeholders. Next, some issues and assumptions about the data that have been collected for this research work are discussed. Also, issues related to the methodologies that have been used, such as weaknesses, other means of improvement, differences and similarities between this work and other related works, also will be considered. Finally, the value of this work and its usability, along with the potential for transferability to other domains, and what other work can benefit from this project will also be discussed.

9.2 Utilisation of the Semantic Web: Future Outlook

The prolonged course of illness from chronic diseases such as diabetes, hypertension, and asthma results in decreased quality of life for many people around the world. Chronic diseases also impose a huge burden on governments in terms of human resources and costs. Many patients have turned to the Internet to learn more about their chronic diseases, to buy products such as medical devices and medicine, to seek online consultation, to subscribe to newsletters, and even get information about classes, workshops and other activities, related to their conditions, that are close by. Chronic disease-related websites offer many valuable resources that can aid patients with their life-long management of their illness, but many patients may be unaware of such services. Most patients looking for information on the Internet turn to search engines normally with simple one- or two-keyword queries. After that they are faced with hundreds of thousands of HTML pages to browse. Many people end up just collecting the first few websites and may not obtain adequate results as expected. Most information is still buried or hidden away from casual users or patients.

The semantic web, once fully developed, will allow users, for example, to ask questions and obtain precise answers to questions such as the following:

1. "Where can I find an insulin pump with a capacity larger than 200 units and a weight of less than 120 grams?"
2. "I need the cheapest book about Asthma with a chapter on alternative medicine"
3. "Is there any workshop on diabetes management in London next month that is intended for pharmacists?"
4. "I need to know the types of food that lower blood pressure and contain low amounts of sugar"
5. "Where is the closest centre that provides asthma therapy for children?"

A special software, a web agent, will interpret such questions into machine understandable queries that work like querying a database using the powerful SQL language. For example, the first question might be interpreted as follows: "SELECT insulin_pump WHERE capacity >200 AND weight <120 FROM some_table". Some queries may not be so obvious such as the last one in the list above! In this case the web agent software must have some information saved about the user such as her/his address, age, browsing habits, etc. so agents can be personalised to adapt to users' needs. These agents will certainly need to work with some form of search engines that traverse the semantic web collecting knowledge into a knowledge base or a database. This is also referred to as parsing and processing ontologies[22]. There has been some limited research in this direction. Probably the most prominent example in this area is Google's version for a search engine for the semantic web which they called "Swoogle". In general, such issues are still open for more research.

The possibilities and the different ways of utilising the web with meaningful information being made available to machines become endless, and is also a fertile ground for further research. Patients with any type of disease can benefit from such a future web, however patients with chronic diseases will benefit the most, since most chronic diseases require life-long management. The current web does a great job in linking pieces of related information, displaying images, sounds, etc. in

[22] One of the definitions of the semantic web is a globally distributed collection of ontologies.

human-understandable ways. Of course, websites having good, understandable design enable information to be easily found and hyperlinks are easy to follow. Such a website does very well in relation to what it is intended to do, and that is to display information for users to read, listen to, or watch. However, since HTML tags are only meant for rendering information on the screen, machines can do very little with such meaningless symbols.

9.3 Decisions and Assumptions about the Collected Data

The decision to choose diabetes, asthma, and hypertension for this work was based on their common occurrence and the need for life-long management. After deciding on the three chronic diseases, the next decision was to decide on how to go about collecting representative sets of related websites. It has been assumed that people who are looking for information about these diseases will most likely use search engines such as Google or AltaVista and then navigate through the first few websites. Most popular search engines rank websites based on their popularity. This indicates that downloading the first 100 diabetes-related websites should give a well-presented set of data to work with. This is also true for asthma and hypertension.

When extracting data, several assumptions have been made concerning the contents of the collected web documents, which are as follows:

1. A webpage may contain no instances of products and no instances of services.
2. A webpage may contain one or more instances of products.
3. A webpage may contain one or more instances of services.
4. One instance of a product may span more than one webpage.
5. One instance of a service may span more than one webpage.
6. A webpage may contain more than one instance of products and services.
7. A webpage may contain several links, where each link leads to another page containing instances.

In theory, it is possible that any of the above cases can occur on the web, but in reality only the first three have been observed and therefore have been considered for this work. During the analysis of the websites no instances of the fourth, fifth, or sixth have been encountered. The last case is not considered since each physical page (file) is processed separately and all the data are reached eventually. It is important to note that all the processing has been carried out offline. So, this issue has not been considered for the following two reasons. First of all the order of processing pages is not important for this work. Secondly, the fifth and sixth cases above state that it is very rare that one instance spans more than one page which supports the claim that processing pages belonging to some website in a certain order is irrelevant to this work. However, as will be discussed in a subsequent section, search engines may benefit from this work, in which the seventh case above becomes important to consider since pages are most likely to be collected with an online crawler.

9.4 Difficulties in Processing the Collected Data

Processing HTML documents that others have written is not an easy task. Some web developers design simple and straightforward pages, while others depend heavily on embedding scripts. Some web developers work with text and images separately, while others use images to draw fancy text on web pages. This means that information can be very hard for programs to interpret and in fact may need some specialised software to recognise it. Some developers link their pages with documents written in other formats such as PDF, PS, RTF, etc. while others stick with HTML and hardly use any other formats.

Fortunately, the number of non HTML formatted files that were encountered during this research was insignificant, therefore dealing with other formats has not been discussed here. For the interested reader, it should be mentioned that there are some open source programs that can read text in any of the formats mentioned above and

translate it to HTML. Also, some search engines offer the option of transforming PDF and PS documents to HTML with the click of a mouse button.

9.5 Common Structures: Hypothesis and Analysis

It has been hypothesised that related websites exhibit common structures. To the author's knowledge, such a hypothesis has never been addressed nor proved before. Based on the analysis made on the collection of chronic disease-related websites, it is suggested that related websites do exhibit common structures. This has been articulated clearly in Figures 8.1 - 8.3. An ontology has been built firstly for the domain of diabetes websites and then it has been edited to cover the asthma and hypertension-related websites. The methodology used in building the ontology is based on well-defined principles that have been presented by well-known researchers in the field. This methodology has been discussed in detail in chapter 3.

The IE system that has being developed during this research work started as a java program used in the analysis of the collected websites. Some special excel spreadsheets have also been designed and prepared to enter the collected data (see Appendices A, B, and C for more details). Once the data were entered into the spreadsheets, they were plotted and analysed, the results being presented in chapter 8. To ensure high accuracy of the collected data, a sample from the same set of websites has been reselected several months later and the data have been recollected again and then compared with the previously collected ones. The differences between the two were insignificantly small.

9.6 Methodological Issues and Related Work

Most web pages on the Internet are in the form of free text. Besides the semantic web, extracting and restructuring information found on the web has become an important task in many fields such as information retrieval and web content mining (Mathias & Jean-Pierre, 2001; Muller et al., 2004). Search engines that utilise the extraction and structuring of information can gain capabilities far beyond those with

simple keyword search techniques. Extracting attribute-value pairs allows search engines to resolve word meaning ambiguities. A simple example is the word "bank" which could mean the bank of a river or a bank of money. This is done by classifying terms based on the surrounding words or topic identification. Also, it has been used in the development of domain-specific and focused search engines.

Some researchers are working on mining web content. For example, there is the mining of abstracts from the medical literature with an aim of finding previously undiscovered knowledge. The field of data mining, a more mature field, requires that data are prepared in structured format before processing. This implies that to apply normal data mining techniques successfully on web content, first it should be transformed into structured format. There have been many projects that have attempted this in relation to web content mining. These are just a few examples showing the benefits of extracting information from free text on the web and transforming it into a structured form.

Information Extraction (IE) has started as an NLP task that is concerned with extracting relevant information from natural text. The two main ingredients for any IE system are a set of extraction rules and a specific template to be filled. The extraction rules are used to find the value for each slot in the template. The end product is actually a set of attribute-value pairs. Today, state-of-the-art approaches in IE are based on machine learning, where the extraction rules are mined from training samples. The training samples are normally tagged instances with labels to be learnt from, thus these approaches are called supervised. They eliminate the need to write extraction rules manually. However, they require training samples to be supplied and also carry the cost of a decrease in efficiency. This means that the manually created extraction rules perform better. The main problems with these rule-based systems are: first they are expensive and time-consuming to build, and second they are very domain specific, and could not be easily adapted to new domains. Both approaches require a predefined template to be filled by the system.

A challenging task to be tackled in this project was to adapt and enhance IE technologies to meet the requirements set for this research work. The resultant system is not completely adaptable to any specific task. However, it is meant to be used with the most common tasks based on the results of the study that was made on the sets of the chronic disease-related websites. Also, since it is object based, it should be easy to customise it and extend it to work with some more specific tasks.

It should be emphasised that the main objective has been to determine the most efficient and shortest possible ways to populate the semantic web with the most common and most needed knowledge that should fulfil stakeholder needs. To fulfil this aim the following issues have been addressed:

9.6.1 Providing Common Set of Concepts and Extraction Rules

The first is the development of sets of object–based data structures that cover the most common categories as specified by the ontology for the chronic disease-related websites. According to the earlier study that was made on the three sets of chronic disease-related websites, the two most common and general concepts found were products and services. Other concepts have been categorised under these two concepts, thus forming hierarchies of concepts. The study has shown, as presented in the last chapter, that 50% of chronic disease-related websites provide products. Every product has a set of attributes such as brand name and price. Attributes for each product type has been determined. These form the slots to be filled when extracting information from a new unseen webpage.

Earlier IE systems that have been developed were very task-specific and based on themes that had been planned by DARPA for the MUC conferences. For example, the first and second MUC were about extracting specific information from messages about naval operations. MUC-3 and MUC-4 focused on extracting specific information from news articles about terrorism (i.e. extracting terrorist names, targeted locations, dates, victims' names, etc.). MUC-6 and MUC-7 focused on business events that had been extracted from Wall Street Journal text. Other systems

were developed later for other tasks. Table 9.1 shows some of the well-known IE systems and the type of information that has been extracted.

Researcher(s)	Type of Information Extracted
Riloff 1993	Terrorist Event Descriptions
Kim & Moldovan 1993	Terrorist Event Descriptions
Huffman 1995	Corporate Management Changes
Soderland et al. 1995	Hospital Discharge Reports
Soderland et al. 1997	Weather Forecast Reports
Kushmerick et al. 1997	Country Codes
Califf & Mooney 1997	Computer-related Job Postings
Freitag 1998	Course and Research Projects
Fukuda 1998	Gene and Protein Names
Craven 1999	Molecular Interactions
Craven 2000	Computer Science Departments
Alani et al. 2003	Information about artists
Rzhetsky 2004	Molecular Pathway Data

Table 9.1 *IE Systems and type of information that has been extracted*

Some of the early systems that were influenced by MUC adopted the same topics that were chosen by DARPA and were of interest to the agency. To the author's knowledge, no research work has been devoted to find common knowledge in any domain that can be of great benefit to many stakeholders as explained previously.

The ontology that was constructed provides an efficient method for organising concepts, their attributes, and the relationship between these concepts.
It has been found that many concepts share the same or some attributes. It should be noted, however, that some of these attributes are also concepts. For example, Figure 9.1 shows the most common attributes and the percentage of each occurring in other concepts.

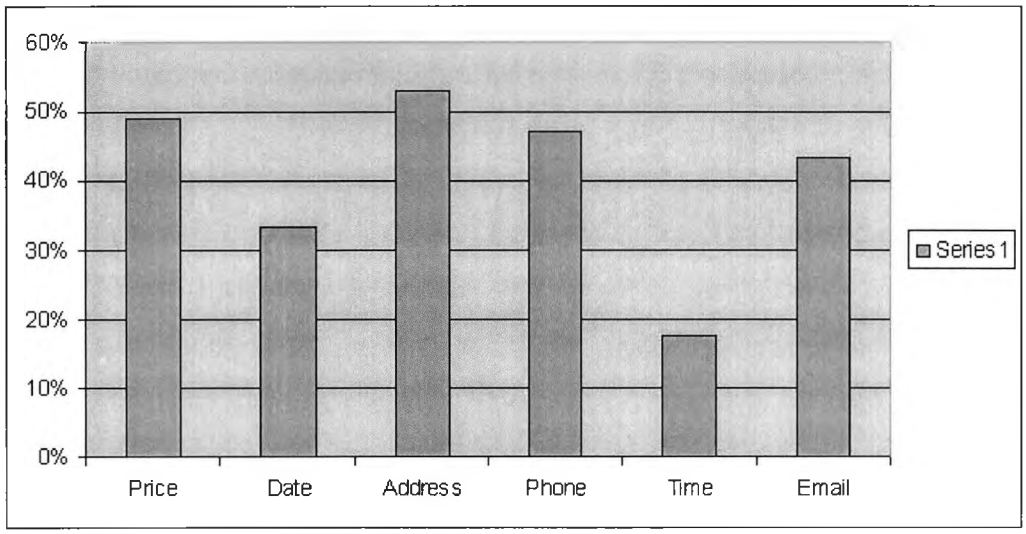


Figure 9.1 *Most common attributes and their percentage of occurrence in other concepts*

For example, the most common attribute for any product is its price. Many IE systems that have been reviewed use regular expressions to extract entities like dates, prices, emails, etc. In this work the nature of these entities has been studied well, and each entity has been encapsulated inside an object along with their own properties and methods. Normally, such methods are used to set and get the values stored in each of the entity's properties. Each entity contains a method that consists of regular expressions for extraction. Not all entities can be extracted with regular expressions. For example, personal names and locations are strings that cannot be extracted with regular expressions. Probably, the simplest way to recognise named entities is by searching a database that contained previously known examples.

Researchers have turned to other methods for named entity recognition, which is an active research field. Today there are many algorithms in existence that benefit from some hints such as word capitalisation and other grammatical structures. Other researchers have used machine learning methods to learn structures for discovering named entities that exist in their documents. Some researchers have studied the nature of medical text where some medical terms exhibit a special format that can benefit from special treatment. The problem in this research work has been minimised because the medical terms have been processed by MMTx and classified

highly correctly. Most medical terms that were encountered on chronic disease-related websites are terms like the names of medical devices, medicines, special foods such as herbs, vitamins, etc.

Because of this, the error rate has been very small when experimenting with web pages that contain products, and conversely very high when experimenting with web pages that contain events. Many phrases that belong to events which contain personal names and locations (i.e. addresses) have been classified "Unknown Phrases". This is an issue which has been left for future work.

9.6.2 Object Oriented Modelling for Information

Programming in an object-oriented language such as java provides a more natural paradigm for modelling real world objects. Concepts saved in the ontology have been mapped to java classes and most importantly their attributes and relationships have been preserved as well. Classes in java are blueprints that describe real word entities such as products and services. Once they are instantiated they become objects, where the attributes are usually populated with their values. Hence, it is exactly like filling the slot of a frame. Viewing the data model for an information extraction system from this perspective simplifies many aspects of the implementation of such a system. It is exactly a divide-and-conquer approach; an object of type price[23] has all the related attributes such as the value, the currency type, a vocabulary control number that links it to its semantic type, and also all related methods. Other entities, such as email and date, have been designed in the same way.

Adapting the object-oriented paradigm in the design and implementation of this project has greatly influenced the handling of many issues, such as the output. For the semantic web, OWL has become the standard language for representing knowledge. To comply with the standards, a method named "toOWL" has been implemented in every class of the data model. This method transforms objects into

[23] Note that while price is an attribute of product, it is also a concept on its own.

an OWL format that can be displayed on the screen, written to disk, or sent to another process in the same computer or across the globe.

At the time when many of the early IE systems were developed, the semantic web was unknown. Their primary interest was only to extract information into some type of structure such as a record in a database.

9.6.3 Concept Matching

Many IE systems facilitate a lexicon that contains terms that represent the domain of discourse. These terms usually trigger some action to be taken once they are found in text. This means these IE systems are term-based systems. Their efficiency depends vitally on the coverage of such lexicons and dictionaries. For very large domains, or like in the case of this current research work where the domain(s) is products and services found on chronic disease-related websites, it become impossible to cover all terms.

To overcome this problem a novel approach has been developed which is based on concept matching instead of term matching. The method has been presented in chapter 6 and is illustrated in the following example.

"*ID: BPH2*
Name: Blood Pressure Monitor
Description: Easy to Use. . . press Start - in seconds, your blood pressure and pulse are displayed.
Price: $99.98"

The paragraph is first processed by MMTx and some terms are tagged with their semantic concepts. For example, the phrase "Blood Pressure Monitor" is mapped to

the concept "Medical Device" and the value "$99.98" is mapped to "Price"[24]. The ontology is then queried to determine the category based on the given concepts. In the above example these concepts are matched with the category "Product".

A needed improvement is the requirement to refine the set of rules to return what type of product and instantiate the correct type. Currently, only a few attributes have been tested. More work is needed in this area.

9.7 Other issues

During the course of this research work many issues have opened up that are worthy of further investigation. Some are related directly to this work while others are related to the semantic web in general. These are as follows:

9.7.1 Semantic Search Engines

Like today's search engines that traverse and process the current web, different types of search engines will be needed to traverse and process the semantic web. Upon examining Swoogle, it seems that it only searches semantic web documents for terms! This shows that this area is still in its first steps of development.

It is probably worth researching methods for the development of a health-related semantic web search engine. According to Swoogle, there exist currently over one and a half million semantic web documents. As a preliminary study, it may a good starting point to use Swoogle to find and collect a set of health-related semantic web documents, then study their structure and investigate methods to query such documents in a way that closely parallels the scenarios that were presented early in this chapter.

[24] Prices are matched using regular expressions during the preprocessing phase since MMTx does not handle numerical values.

9.7.2 Work in Other Languages

A concept is a concept, no matter what language it is expressed in. However, extracting information in languages other than English is different because IE depends on NLP which is language dependent. For example, the first step in information extraction is tokenisation or term zoning. In some languages, especially those of Asia, this can be a problem where some words are literary connected together to form another entity with a richer meaning.

During the process of collecting chronic disease-related websites, some websites written in other European languages, such as Spanish and German, have been discarded. The number of non-English websites is increasing. Investigating IE in other languages at some point will become essential. Or is it best just to translate the text into English and then proceed with English IE? Would this achieve the same results? Does a non English health-related website differ from an English one in terms of the treatment of health-related terms? Are there any standard vocabularies that have been established and used? Etc.

9.7.3 Semantic Data Mining

A more mature field than information extraction is the field of data mining, which has achieved great success during the past decade. Data mining is the process of discovering new knowledge in databases, i.e. in already structured data. Since the semantic web is an extended version of the current web and a structured one, this means that a new field, that we may call "Semantic Web Mining", must emerge that is to apply current data mining methodologies and maybe research new techniques to harvest the semantic web for new knowledge. This would also apply for medical knowledge which should benefit very greatly from such a research.

9.7.4 Work on Other Domains

One of the main objectives of this work was to find common structure in chronic related websites. It is therefore worth looking at other domains in more depth and especially those of a medical nature to see if there are any commonalities that can be discovered. Such examination may benefit other domains.

9.7.5 Semantic Web Tools

The current web has revolutionised the information world in so many ways. During the past decade many tools have been developed for the web in so many different domains. What would be the status and function of such tools with the existence of the semantic web? What added value can the semantic web bring to these tools?

One potential benefit from this project includes the development of tools based on this work that would aid the webmaster in adding a semantic layer to their websites and thereby accelerating the semantic web. An example would be an interactive development environment that could intelligently supervise the webmaster in the process of constructing semantic web documents based on their HTML documents.

9.7.6 Current Search Engines

The current web will not go away with the evolution of the semantic web because the semantic web is another layer(s) that will coexist with the current web. In other words, other files with different extensions live in the same space with the current web. These files hold semantic information about the current web. In this case the current search engines will not go away either. However, they may access these other files to gain some added value. Today, there are many domain-specific search engines and some of them are for the medical domain. How and what is the best way for these types to gain more efficiency?

9.7.7 The E-Worlds in the New Era

During the last few years the world has witnessed the birth of many new concepts. One of these is the e-world, including the likes of e-commerce, e-health, e-government, e-learning, etc. Many issues will rise concerning each one of them with the evolution of the semantic web. In other words, how can each one benefit from the semantic web? Would the addition of a web with meaning have a great impact, such as for example, enhancing the current process that is done today?

9.7.8 Working with Images

Another issue worth looking at is the processing of web content other than text, most importantly processing images. In particular, there is the need to find suitable structures for the management and retrieval of images that reside on chronic disease websites. A picture is worth a thousand words, but how can we give a meaning to pictures residing on the semantic web? Today, image processing on the web is one of the hot research topics that attracts many researchers.

Looking into the crystal ball one can see that the current web could indeed be brought to its full potential in so many ways. The possibilities become seemingly endless with the new technologies. The acceleration of the development of the semantic web becomes vital to its success and more research is needed to bring the semantic web up to the level where these things start to happen and results are felt.

This current research work is directed towards this goal of moving the world faster into the semantic web and to accelerate the process. In the worst case scenario whereby the semantic web should fail, such a system would at least represent added value to existing search engines.

9.8 Conclusion

The web has become one the world's primary sources for information. In the medical domain, the web has become a valuable source of information for patients and professionals alike. The semantic web aims to extend the web into machine understandable form. Therefore, the automation of knowledge acquisition from the current web is becoming an important step towards this goal. The aim of this work was twofold; first to learn what types of information exist in chronic disease-related websites, and secondly how to extract and structure such information into machine understandable form. It has been shown that these websites exhibit many common concepts which resulted in the construction of the ontology to guide in extracting information for new unseen websites. Also, the study has resulted in the development of a platform for information extraction that utilises the ontology. While further work is still needed in this part, the experiments to date have shown encouraging results.

Chapter Ten

Conclusion

"The foregoing considerations lead us to the very important conclusion, that matter is essentially force, and nothing but force; that matter, as popularly understood, does not exist, and is, in fact, philosophically inconceivable."

- Alfred Russel Wallace

10.1 Meeting The Objectives

In chapter one, seven objectives were identified for this research project. In the following, the seven objectives are presented in terms of the extent to which each one has been met.

10.1.1 Web Analysis

The first objective was to study and analyse health-related web documents, in particular three sets of websites related to diabetes, asthma, and hypertension. The aim was to study their structures in order to identify what types of knowledge should be extracted, and more interestingly, to what extent common and shared knowledge exists among chronic disease- related websites.

The study has been carried out as planned. The main concepts from these websites have been extracted along with the attributes for each concept. A full analysis with the results has been reported in chapter 8 and discussed further in chapter 9.

10.1.2 Ontology Development

Based on the results of the first objective, the second one was to build an ontology to guide the information extraction (IE) system as to which features should be extracted from a given website. This work has also been carried out. Analysis of the websites has revealed a set of common categories which have exposed the purpose for each site. These fall under two main headings, services and products. Under each one, a list of subcategories also has been identified along with the attributes for each one. For example, if a website is selling insulin pumps, then it is expected to find properties such as models, weights, capacities, prices, etc which are essentially the attributes for this product type. During this study, methodologies for ontology development have been studied and used in engineering the proposed ontology. Also, in the process the roles and types of ontologies have been explored and reported in chapter three.

In ontology engineering, it has been stated that it is good practice to examine the possibility of reusing exiting ontologies rather than starting from scratch. To accomplish this part of the work, two candidates, WordNet and UMLS, have been compared. The results were also reported in chapter 8. UMLS has being used in this work as an upper-level ontology with great success. A full discussion about this issue has also been presented in chapter 9.

10.1.3 Information Extraction

The next objective was to review state-of-the-art approaches in information extraction, and to identify problems with these approaches when applied to web documents.

An IE approach has been utilised to extract semantic entities to be used to enrich the semantic web.

The two main IE methodologies identified were the knowledge engineering approach and the machine learning approach. Each approach has been studied in depth and problems have been identified and reported in chapter 4.

10.1.4 The New Framework

The next objective was to design and develop a framework which consists of the components that are needed for the extraction system. Rather than implementing each component from scratch which may take long time to accomplish, open-source components, which have been developed in research labs and used by many other researchers successfully, are considered and reused in this work. An initial prototype of the system has been implemented and tested. The development of the system has been reported in chapter 6, the testing procedures presented in chapter 7, and the results reported in chapter 8.

10.1.5 System Output

The next objective was to explore ways for representing the output structures. Some standards were looked at and considered such as XML and RDF. More discussion on these standards can be found in chapter 2. During the last few years the web ontology language (OWL) has been standardised and is now considered as the official language for the semantic web. Examination of OWL has revealed that it would be a good choice to formally represent the ontology. Fortunately, the health informatics research team at Stanford who developed Protégé have also developed Protégé-OWL, a plugin for managing ontologies in OWL format. Protégé-OWL has been used in this research work successfully. Firstly, it was used to construct and edit the ontology; secondly, to access the ontology programmatically from the IE system; and thirdly, to save it in OWL format.

10.1.6 System Applicability

The system has been applied specifically to the domain of diabetes and to other chronic diseases such as asthma and hypertension. However, there is the need to test it with other domains.

10.1.7 System Evaluation

The system has gone through comprehensive evaluation as shown in chapter 7, with the results of the evaluation being presented in chapter 8 and discussed in detail in chapter 9.

10.2 Contributions

During the first year of this work a survey on e-Health was performed where the main issues, problems, and opportunities have been identified and published (Eljinini et al., 2003). The survey has established the need for web content mining of health related information. The paper outlined the need for such information to be extracted and transformed into a structured form before any successful web content mining can be performed.

It should be stated here that the whole solution to the problem described in this work is essentially novel. During the first part of this research work, three sets of websites related to diabetes, asthma, and hypertension have been downloaded and some time was spent studying and analysing these sets. This has resulted in producing a group of common categories based on the real purpose of each website. Part of this work also has been published in Eljinini et al. (2006).

The study has shown that similar websites exhibit common structures. The structures revealed can be of benefit to several fields, spanning both information and computer science and health informatics:

10.2.1 The Semantic Web

In previous chapters, the importance of enriching the semantic web has been elaborated and discussed. The development of a "global" system that can harvest knowledge from arbitrary websites and make it ready for the semantic web is simply impossible at this time. The use of current IE technologies for this task also is not feasible. This research work has demonstrated that common knowledge exists in many related websites which can form the seed for more knowledge acquisition towards the semantic web.

10.2.2 Information Retrieval

Recently, much of the research work in information retrieval (IR) has been concept based. By integrating conceptual models to work with existing IR technologies, new models can be developed that benefit from richer structures. The set of common structures that has been presented in this research work can be very useful and added value to the development of concept-based information systems.

10.2.3 Information Extraction

It has been stated in the literature that the knowledge engineering approach is more efficient than the machine learning approach. However, the knowledge engineering approach is harder to build, needs domain-specific expertise, and requires more time. Some researchers have suggested the need for a hybrid approach that can benefit from both approaches, but no study has been carried out in this direction so far. This research work has revealed common sets of concepts with attributes that can be reused successfully in a hybrid approach that uses both methodologies.

From an application point of view, adapting object oriented paradigms in modelling the data has shown that such a model can be reused and extended very efficiently.

10.2.4 Data Mining

The field of data mining has matured during the past decade and has been used successfully in many projects. Today, the web has been reviewed as the premier source of information. Most information on the web is in free form. The automation of information extraction and structuring is a necessary first step toward successful web content mining. The results obtained from this research work can be of great benefit to researchers in web content mining, where they can be used effectively to harvest new knowledge for the web.

10.2.5 Health Informatics

The same issues that have been discussed above can be applied to health informatics in specific cases, since the data that have been used in this research work relate to the domain of health informatics.

It should be pointed out that the components which have been used in this research work, such as MMTx and Protégé-OWL, have been developed by teams from the health informatics domain. Such components have been utilised in a novel way to serve the purpose of this research project.

The developers of the MMTx MetaMap system have stated that common uses of MMTx include information extraction tasks, but no research has explored this issue in further detail. In this study MMTx was subjected to a full evaluation. The strengths and weaknesses have been reported and can be of benefit to other researchers working on text analysis projects.

During the course of this project the literature has been searched on the Internet – including journals - to attempt to find a similar study, but none was found. The results and conclusions of this present analysis could be of benefit to other health informatics researchers who are engaged in extracting and analysing information from other medically-related websites. The ontology developed here can form a common, sharable ontology that can be extended easily to fit their work.

It has been suggested in chapter 9 that IE systems rely on domain-specific dictionaries to identify relevant terms in text. In large domains, it becomes very hard to manually build such dictionaries that produce good coverage. To overcome this problem a novel approach has been developed which is based on concept matching rather than term matching.

10.3 Future Work

The study has concentrated on the domain of chronic diseases and in particular diabetes, asthma, and hypertension. Future work should first look at other types of health related websites in particular and other domains in general. One reason is to see how much of the studied and exploited information is covered in other domains. Another reason is to see how the developed system would behave and adapt to other domains.

Another future task would be to establish a repository for the object oriented data model to be shared, used, and extended by other researchers and developers for maximum benefit.

Currently, the IE system uses a set of small manually engineered rules that cover a small number of attributes for testing. The system should be extended in two ways: first, by adding more rules to cover more attributes; second, and more importantly, is to extend the system into the hybrid approach by utilising some machine learning algorithms.

In chapter two a discussion about the importance of having information about the website such as date of last update, etc. was included. It is worth investigating methodologies for extracting such knowledge into a predefined template (i.e. the Dublin Core Standards).

Finally, this research work has introduced a novel method for locating and filtering relevant phrases. During experimentation, the system gave excellent results compared with other state-of-the-art systems, which offers encouragement for further research work in this very important field.

References

Alani H, Kim S, Millard DE, Weal MJ, Hall W, Lewis PH, Shadbolt NR, 2003. "Automatic ontology-based knowledge extraction and tailored biography generation from the web", *IEEE Transactions on Intelligent Systems,* 18(1): 61-65.

Appelt D, Israel D, 1993. "*Introduction to Information Extraction Technology*", A Tutorial prepared for IJCAI-99, Stockholm-Sweden.

Ashish N, Knoblock C, 1997. "Wrapper generation for semi-structured internet sources", *ACM SIGMOD Rec., 26(4): 8-15.*

Bechhofer S, Horrocks I, Goble C, Stevens R., 2001. "OilEd: A reason-able ontology editor for the semantic web", in: *KI-2001: Advances in AI, LNAI 2174*, Berlin: Springer, 396-408.

Berners-Lee T, 1999. *Weaving the Web,* San Francisco: Orion Business Books.

Berners-Lee T, Hendler J, Lassila O, 2001. "The semantic web", *Scientific American*, 284(5): 28-37.

Bodenreider O, Rindflesch T, Burgun A, 2002. "Unsupervised, corpus-based method for extending a biomedical terminology", in: *Proceedings of the ACL'2002 Workshop on Natural Language Processing in the Biomedical Domain,* Philadelphia: Association for Computational Linguistics, 53-60.

Bodenreider O, McCray A, 2003. "Exploring semantic groups through visual approaches", *Journal of Biomedical Informatics,* 36: 414-432.

Bodenreider O, 2004. "The Unified Medical Language System (UMLS): integrating biomedical terminology ", *Nucl. Acids Res.*, 32: 267-270.

Brin S, Page L,1998. "The anatomy of a large-scale hypertextual web search engine", *Computer Networks and ISDN Systems,* 30(1-7): 107-117.

Broekstra J, Klein M, Decker S, Fensel D, Harmelen F, Horrocks I, 2001. "Enabling knowledge representation on the web by extending RDF schema", Computer Networks, 39(5): 609-634.

Burgun A, Bodenreider O, 2001. "Mapping the UMLS semantic network into general ontologies", in: *Proceedings of AMIA Annual Symposium 2001,* Washington, DC: Heinley&Belfus, 86-90.

Califf M, Mooney R, 1997. "Relational learning of pattern-match rules for information extraction", *Working Notes of {AAAI} Spring Symposium on Applying Machine Learning to Discourse Processing*, Menlo Park, CA: AAAI Press, 6-11.

Charniak E, 1993. *Statistical Language Learning.* Cambridge, Mass.: MIT Press.

Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell T, Nigam K, Slattery S, 2000. "Learning to construct knowledge bases from the world wide web", *Artificial Intelligence,* 118: 69-113.

Craven M, Kumlien J, 1999. "Constructing biological knowledge bases by extracting information from text sources". in: *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99),* Heidelberg: AAAI Press, 77-86 .

Decker S, Fensel D, Harmelen F, Horrocks I, Melnik S, Klein M, Broekstra J, 2000. "Knowledge representation on the web", in: *International Workshop on Description Logics (DL'00)*, Aachen: Springer Verlag, 89-97.

Eljinini MA, Carson ER, Sarhan NA, 2003. "Internet technologies for healthcare and patient management: opportunities and issues", in: *Proceedings of the 5th Jordanian International Electrical & Electronic Engineering Conf. (JIEEEC 2003)*, Amman- Jordan, 61-65.

Eljinini MA, Sarhan NA, Carson ER, 2006. "Towards the semantic web: extracting common concepts from chronic disease - related websites", in: *Proceedings of the International Medical Informatics and Biomedical Engineering Conf. (IMIBE'06),* Amman-Jordan, 118-123.

Elkin P, Brown S, Husser C, Bauer B, Wahner-Roedler D, Rosenbloom S, Speroff T, 2006. "Evaluation of the Content Coverage of SNOMED CT: Ability of SNOMED Clinical Terms to Represent Clinical Problem Lists", *Mayo Clin. Proc.*, 81(6):741-748.

Fensel D, Horrocks I, Harmelen F, Decker S, Erdmann M, Klein M, 2000. "OIL in a nutshell", in: *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modelling, and Management,* Berlin: Springer-Verlag, 1-16.

Freitag D, 1998. "Information extraction from HTML: application of a general learning approach", in: *Proceedings of the 15th Conference on Artificial Intelligence*, Madison-WI: AAAI Press, 517-523.

Freitag D, McCallum A, 1999. "Information extraction with HMMs and shrinkage", in: *AAAI-99 Workshop on Machine Learning for Information Extraction*, Orlando-FL: AAAI Press, 31-36.

Fridman Noy N, Hafner D, 1997. "The state of the art in ontology design", *AI Magazine*, 18(3): 53-74.

Fridman Noy N, Grosso W, Musen MA, 2000. "Knowledge-acquisition interfaces for domain experts: an empirical evaluation of Protégé-2000", in: *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering,* Chicago-IL.

Fridman Noy N, Sintek M, Decker S, Crubezy M, Fergerson R, Musen M, 2001. "Creating semantic web contents with Protégé-2000", *IEEE Transactions on Intelligent Systems* 16(2): 60-71.

Fridman Noy N, Chugh A, Liu W, Musen MA, 2006. "A Framework for Ontology Evolution in Collaborative Environments", in: *5th International Semantic Web Conference*, Athens, GA: LNCS 4273.

Fukuda K, Tsunoda T, Tamura A, Takagi T, 1998, "Toward information extraction: Identifying protein names from biological papers", in: *Proceedings of the Pacific Symposium on Biocomputing '98 (PSB'98)*, Maui-Hawaii, 707-718.

Gondy L, Hsinchun C, 2001. "Meeting medical terminology needs-the ontology-enhanced medical concepts mapper", *IEEE Transactions on Information Technology in Biomedicine*, 5 (4): 261-270.

Gruber TR, 1993. "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, 43: 907-928.

Guarino N, 1998a. "Formal ontology and information systems", in: *Proceedings of FOIS'98,* Amsterdam: IOS Press, 3–15.

Guarino N, 1998b. "Some ontological principles for designing upper level lexical resources", in: *Proceedings of First International Conference on Language Resources and Evaluation*, Granada-Spain, 527-534.

Guarino N, Giaretta P, 1995. "Ontologies and knowledge bases: towards a terminological clarification". in: N Mars (ed.) *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing 1995,* Amsterdam: IOS Press, 25-32.

Gupta S, Kaiser G, Neistadt D, Grimm P, 2003. "DOM-based content extraction of HTML documents". in: *Proceedings of the 12th International World Wide Web Conference,* Budapest: ACM Press, 207-214.

Haarslev V, Möller R, 2001. "RACER system description", in: *Proceedings of the International Joint Conference on Automated Reasoning, IJCAR'2001*, Berlin: Springer-Verlag, 701-706 .

Huffman S, 1995. "Learning information extraction patterns from examples", in: *IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing,* Menlo Park, CA: AAAI Press, 127-142.

Jelinek F, 1997. *Statistical Methods for Speech Recognition.* Cambridge, Mass.: MIT Press.

Jurafsky D, Martin J, 2000. *Speech and Language Processing*. Upper Saddle River-NJ: Prentice-Hall, Inc. Pearson Higher Education.

Kim J, Moldovan D, 1993. "PALKA: a system for lexical knowledge acquisition", in: *Proceedings of the 2nd International Conference on Information and Knowledge Management*, New Your: ACM Press, 124–131.

Knight K, Luk S, 1994. "Building a large knowledge base fro machine translation", in: *Proceedings of the 12th nation conference on AI (vol. 1)*, Menlo Park, CA: AAAI Press, 773-778.

Knublauch H, Fergerson RW, Fridman Noy N, Musen MA, 2004. "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications", in: *Third International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, 229-243.

Kushmerick N, Weld DS, Doorenbos R, 1997. "Wrapper induction for information extraction", in: *Proceedings of the 15th International Joint Conference on Artificial Intelligence,* San Francisco: Morgan Kaufmann Publishers Inc, 729-737.

Lenat DB, 1995. "CYC: a large-scale investment in knowledge infrastructure". *Communications of the ACM,* 38(11): 33-38.

Maedche A, Staab S,2001. "Ontology learning for the semantic web", *IEEE Transactions on Intelligent Systems,* 16(2): 72-79.

Manola F, Miller E, 2003. *RDF Primer*, W3C Proposed Recommendation 15 December 2003.

Mathias G, Jean-Pierre C, 2001. "Toward a structured information retrieval system on the web: automatic structure extraction of web pages", in: *International Workshop on Web Dynamics*, London-UK.

McCray A, Srinivasan S, Browne A, 1994. "Lexical methods for managing variation in biomedical terminologies", in: *Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care*, Washington, DC: Heinley&Belfus, 235-239.

Mitchell T, 1997. *Machine Learning*, New York: McGraw-Hill.

Miller G, 1995. "WordNet: a lexical database for English", *Communications of the ACM* 38(11): 39-41.

Muggleton S, 1991. "Inductive logic programming", *New Generation Computing* 8(4): 295-318.

Müller HM, Kenny EE, Sternberg PW, 2004. "Textpresso: an ontology-based information retrieval and extraction system for biological literature", *Public Library of Science (PLoS) Biology* 2(11): e309.

Muslea I, 1999. "Extraction patterns for information extraction tasks: a survey". in: *Proceedings of AAAI'99: Workshop on Machine Learning for Information Extraction*, Menlo Park, CA: AAAI Press, 1-6.

Quinlan J, Cameron-Jones R, 1993. "FOIL: A midterm report", in: *Proceedings of the European Conference on Machine Learning,* Vienna: Springer-Verlag, 3-20.

Rabiner L, 1989. "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE,* 77(2): 257-286.

Rector A, Gangemi A, Galeazzi E, Glowinski A, Rossi-Mori A, 1994. "The GALEN CORE model schemata for anatomy: towards a re-usable application-independent model of medical concepts", in: *Medical Informatics Europe, MIE'94*, Lisbon, Portugal, 229-233.

Rector A, Solomon WD, Nowlan W, Rush T, 1995. "A terminology server for medical language and medical information systems", *Methods of Information in Medicine,* 35: 147-157.

Rector A, Bechhofer S, Goble C, Horrocks I, Nowlan W, Solomon WD, 1997. "The GRAIL concept modelling language for medical terminology", *Artificial Intelligence in Medicine,* 9: 139-171.

Riloff E, 1993. "Automatically constructing a dictionary for information extraction tasks", in: *Proceedings of the 11th National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press, 811-816.

Rzhetsky A, Iossifov I, Koike T, Krauthammer M, Kra PB Morris M, Yu H, Ariel Dubou EP, Weng W, Wilbur JW, Hatzivassiloglou V, Friedman C, 2003. "GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data", *Journal of Biomedical Informatics* 37(1): 43-53.

Soderland S, Fisher D, Aseltine J, Lehnert W, 1995. "CRYSTAL: inducing a conceptual dictionary", in: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, San Francisco: Morgan Kaufmann, 1314-1319.

Soderland S, 1997. "Learning to Extract Text-based Information from the World Wide Web", in: *Proceedings of Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, 251-254.

Soderland S, 1999. "Learning information extraction rules for semi-structured and free text", *Machine learning 34: 233-272.*

Sowa J, 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Pacific Grove, CA: Brooks/Cole Thomson Learning.

Srinivasan P, Mitchell J, Bodenreider O, Pant G, Menczer F, 2002. "Web crawling agents for retrieving biomedical information", in: *Proceedings of the International Workshop on Bioinformatics and Multi-Agent Systems (BIXMAS 2002),* Bologna, Italy, July 2002.

Sure Y, Erdmann M, Angele J, Staab S, Studer R, Wenke D, 2002. "OntoEdit: collaborative ontology development for the semantic web", in: *Proceedings of the International Semantic Web Conference 2002 (ISWC 2002),* Berlin: Springer-Verlag, 221-235.

Sure Y, Staab S, Angele J, 2002. "OntoEdit: guiding ontology development by methodology and inferencing", in: *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics*, Irvine, CA: Springer, 1205-1222.

Thibodeau G, Patton K, 1999. "*Endocrine system*", in: *Anatomy & Physiology*, St. Louis, Missouri: Mosby Inc.

Uschold M, Gruninger M, 1996. "ONTOLOGIES: principles, methods, and applications", *Knowledge Engineering Review* 11(2): 93-155.

Witten I, Frank E, 2000. *Data Mining: Practical Learning Tools and Techniques with Java Implementation,* New York: Academic Press.

Appendix A – List of Diabetes Related Websites

	Purpose												
Website URL	EDU	INF	PRO	RES	POR	MEM	NEW	CON	CHA	SOC	MET	DAT	NF
www.aaip.com	X					X	X						9
www.addisons-diabetes.gkznet.com		X					X						10
www.allondiabetes.com		X	X	X		X							100
www.alphanutrition.com		X	X										712
www.altdiabetes.com			X									X	144
www.americandiabetes.com		X	X									X	71
www.arab-diabetes.com				X		X							24
www.atdiabetes.com		X			X								868
www.bbc.co.uk		X					X						1239
www.best-diabetes-cookbooks.com			X										51
www.cdc.gov	X	X		X			X					X	147
www.childrensdiabetesfdn.org	X	X	X			X	X				X	X	41
www.childrenwithdiabetes.com		X	X				X	X				X	34
www.defeatdiabetes.org		X		X			X	X	X			X	981
www.diabetea.com		X	X										9
www.diabetes.about.com		X			X	X	X						20
www.diabetes.audit-commission.gov.uk		X									X		23
www.diabetes.co.uk		X	X			X	X				X		79
www.diabetes.mdmercy.com	X	X		X				X			X		36
www.diabetes.org	X	X	X								X		190
www.diabetes.org.uk	X	X		X		X	X		X	X	X		32
www.diabetes123.com		X	X				X					X	46
www.diabetesaction.org	X	X		X		X			X				231
www.diabetesalliance.org			X			X	X	X		X	X		12
www.diabetesandtravel.com			X								X		55
www.diabetes-book.com		X	X	X			X				X		97
www.diabetes-care.com		X	X								X		40
www.diabetescareandtreatment.com			X			X							27
www.diabeteschat.net		X					X	X			X		3
www.diabetescliniconline.com		X	X		X	X	X	X	X		X	X	90
www.diabetesdek.com		X	X			X	X				X		28
www.diabetesdigest.com		X	X				X				X		89
www.diabetesdirect.net			X			X	X				X		33
www.diabetesed.net	X		X								X		14
www.diabetes-education.com	X	X					X			X	X		21
www.diabeteseducationandresearchcenter.org	X	X					X		X				53
www.diabeteseducators.com	X							X			X		6
www.diabetes-eg.com		X			X	X	X			X		X	60
www.diabetes-exercise.org	X					X		X					28
www.diabetes-explained.co.uk		X					X	X				X	18
www.diabetesforecastlive.com		X					X	X					163
www.diabetesforum.net		X					X	X			X		442
www.diabeteshealing.com		X	X				X				X		77
www.diabetes-herb.com			X	X							X		49
www.diabeteshomecare.com			X								X		23
www.diabeteshopefoundation.com				X			X		X				35
www.diabetesinc.org	X		X									X	6
www.diabetesincontrol.com	X		X				X				X		2

Website URL	EDU	INF	PRO	RES	POR	MEM	NEW	CON	CHA	SOC	MET	DAT	NF
www.diabetesindia.com		X					X				X		140
www.diabetesinstitute.org		X					X		X		X		53
www.diabetes-kids.org		X					X			X	X	X	1
www.diabetesliving.com		X		X	X		X				X		125
www.diabetesmedicalmusic.com			X	X							X		215
www.diabetesmeds.org						X							14
www.diabetes-midon.org	X	X					X			X	X	X	170
www.diabetesmonitor.com		X	X				X				X	X	90
www.diabetesnet.com		X	X								X		8
www.diabetesnews.com							X				X		3
www.diabetes-normalsugars.com		X	X				X				X		59
www.diabetesoneonone.com		X				X	X	X		X	X	X	10
www.diabetesonline.com			X			X	X				X		123
www.diabetesplanner.com	X	X					X	X			X		105
www.diabetesportal.com		X		X	X			X			X		54
www.diabetesresource.com		X	X					X					2
www.diabetesscv.org	X	X				X				X			13
www.diabetes-sight.com													31
www.diabetessociety.org	X	X				X				X			13
www.diabetesstation.com		X	X	X							X		134
www.diabetesstore.com			X								X		284
www.diabetes-supply.com			X			X					X	X	12
www.diabetes-tests.com		X										X	25
www.diabetes-therapies.com		X				X					X		2
www.diabetesthyroidhormone.com		X									X		5
www.diabeteswellness.net		X					X	X					14
www.diabetesworld.com		X					X	X			X		14
www.diabetic.com		X	X				X			X			129
www.diabeticcare.com		X	X				X				X		96
www.diabeticdiet.net			X			X		X			X		4
www.diabeticdrugstore.com			X								X	X	9
www.diabeticexpress.com			X								X		956
www.diabetic-lifestyle.com		X					X				X		631
www.diabetic-recipes.com		X					X				X		273
www.diabeticsupply.com		X	X			X					X		15
www.diabetopaedia.com		X	X			X	X				X		48
www.diseases-explained.com		X	X								X		18
www.endocrineweb.com		X					X				X	X	19
www.endocrinologist.com		X								X		X	32
www.endotext.org		X					X				X	X	14
www.ezdiabetes.com		X	X										11
www.hamotdiabetes.com	X		X								X		38
www.idd-diabetes.com		X		X									23
www.louisianadiabetes.org		X	X	X		X		X	X	X	X		1
www.medicapharma.com			X								X		97
www.medicool.com		X	X								X		101
www.mendosa.com		X	X				X					X	425
www.methodisthealth.com	X	X				X	X	X		X		X	818
www.msdiabetes.org		X					X		X	X			4
www.mydiabetes.com		X				X	X					X	1
www.umassmed.edu	X			X			X				X		2
www.yorkshirediabetes.com		X					X			X			1
COUNT	21	71	50	18	7	28	53	20	9	40	61	24	12053
Total Size (MB)	30	181	90	51	7	21	134	43	18	16	92	43	
Number of Files	1975	9747	4994	2286	1217	1888	7229	2835	1480	1317	5411	3264	

Appendix B – List of Asthma Related Websites

	Purpose									
Website URL	EDU	INF	PRO	RES	POR	MEM	NEW	CON	CHA	SOC
allergy.edoc.com			1	1		1	1			
asthma.about.com		1			1		1			
cure-your-asthma.com			1							
dir.yahoo.com					1					
elitemedical.com			1							
infoforyourhealth.com		1			1					
internalmed.wustl.edu	1			1						
kidshealth.org		1			1		1			
medlineplus.nlm.nih.gov	1		1	1		1	1			
www.aaaai.org	1	1		1		1	1			
www.aafa.org	1	1	1	1		1	1		1	1
www.aanma.org	1	1	1				1		1	
www.air-purifiers-usa.biz			1			1	1			
www.allergy.org.au	1			1			1			
www.allergy-asthma-sinusitis.com		1	1					1		
www.allergybegone.com		1	1							
www.allergysupply.com			1							
www.allergyweb.com		1		1			1			1
www.allernet.com		1				1	1	1		
www.alphanutrition.com		1	1			1				
www.amicrawler.com			1							
www.asthma.org.au	1	1	1	1			1	1	1	1
www.asthmaandschools.org		1								
www.asthmaaustralia.org.au	1	1		1		1			1	1
www.asthmacontrol.com		1								
www.asthmaeducatorsvic.org.au	1					1				1
www.asthmamoms.com		1					1			
www.asthmansw.org.au	1	1		1		1	1		1	1
www.asthmanz.co.nz	1	1	1	1			1		1	1
www.asthmaqld.org.au	1	1		1		1	1		1	1
www.asthmaresearch.org.au		1		1			1		1	
www.asthmatas.org.au		1				1	1		1	1
www.asthmaticbronchitis.com		1					1	1		
www.asthmawa.org.au	1	1		1			1		1	1
www.asthme-reality.com		1		1						
www.betterhealth.vic.gov.au		1				1	1			1
www.drugstore.com			1							
www.efanet.org		1				1	1			
www.lung.ca		1		1			1		1	1
www.lungdiseasefocus.com		1					1			
www.lungnet.org.au	1	1		1			1		1	1

www.lungusa.org		1	1	1		1	1		1	
www.meddean.luc.edu	1			1			1			
www.mhcs.health.nsw.gov.au		1		1			1			
www.myasthma.com	1	1				1	1	1		1
www.nationalasthma.org.au		1	1	1			1			
www.nhlbi.nih.gov	1	1		1			1			
www.nlm.nih.gov		1	1	1		1	1			
www.nqnet.com			1							
www.portablenebs.com			1							

Appendix C – List of Hypertension Related Websites

	Purpose									
Website URL	EDU	INF	PRO	RES	POR	MEM	NEW	CON	CHA	SOC
cardiologychannel.com		1	1					1		
dir.yahoo.com					1					
encarta.msn.com		1								
hyper.ahajournals.org			1	1		1	1		1	
hypertension.newstrove.com		1				1	1			
intl.ahajournals.org		1	1	1	1	1	1		1	
kidshealth.org		1				1				
obgyn.uihc.uiowa.edu	1	1		1			1			
rehd.med.upenn.edu	1	1		1						
www.about-hypertension.com		1					1			
www.advancedclinicalresearch.com				1						
www.ahajournals.org		1	1	1		1	1			
www.ash-us.org	1	1		1		1	1			1
www.bloodpressure.com		1	1			1	1	1		
www.cardiologychannel.com		1	1			1	1			
www.cdc.gov		1	1	1			1			
www.cosehc.org	1				1	1				1
www.dietitian.com		1						1		
www.emedicine.com	1	1		1		1	1			
www.hdcn.com	1	1		1		1				
www.hdi-pulsewave.com			1	1			1			
www.highbp.com						1		1		
www.hypertensionandhealth.com			1				1			
www.hypertensionfoundation.org	1	1		1						
www.hypertensionhelp.com		1					1	1		
www.hypertensiononline.org	1		1			1				
www.ihrfoundation.org	1	1		1		1	1		1	
www.ishib.org	1	1		1			1			1
www.mco.edu	1	1		1		1	1			
www.med.umich.edu	1			1			1			
www.medicinenet.com		1				1	1			
www.mjmoore.co.uk		1							1	
www.nature.com			1	1		1				
www.nhlbi.nih.gov	1	1		1		1	1			
www.nlm.nih.gov		1		1			1			
www.novartis.com		1	1	1			1			
www.ontarget-micardis.com			1	1		1	1			1
www.phassociation.org	1					1	1			1
www.pha-uk.com		1				1	1		1	1
www.phcentral.org		1	1			1	1	1	1	
www.pinc.com			1	1		1	1			

www.pphcure.org		1	1								
www.pphinstitute.com		1					1				
www.pulmonary-hypertension-treatments.com		1	1								
www.sciencedirect.com			1	1							
www.vascular-disease.net		1	1								
www.wfubmc.edu	1			1			1				
www.bpassoc.org.uk		1	1	1		1	1		1	1	
www.bpmonitoring.com			1	1		1					
www.drbloodpressure.com		1		1							

Appendix D – List of Diabetes Related Websites with Products

Website URL	Sff	Nut	Skn	Ins	GIT	InT	Bks	TrK	Med	SW	ID
www.allondiabetes.com				X							
www.alphanutrition.com		X									
www.altdiabetes.com		X	X								
www.americandiabetes.com				X	X	X	X				
www.best-diabetes-cookbooks.com							X				
www.childrensdiabetesfdn.org							X				
www.diabetea.com		X									
www.diabetes.co.uk						X	X				
www.diabetesalliance.org					X	X	X				
www.diabetesandtravel.com								X			
www.diabetes-book.com							X				
www.diabetes-care.com					X	X					
www.diabetescareandtreatment.com					X	X					
www.diabetesdek.com							X				
www.diabetesdigest.com							X				
www.diabetesdirect.net					X	X	X		X		
www.diabetesed.net							X				
www.diabeteshealing.com		X									
www.diabetes-herb.com		X									
www.diabeteshomecare.com					X	X			X		
www.diabetesinc.org							X				
www.diabetesincontrol.com					X				X		
www.diabetesmedicalmusic.com								X		X	
www.diabetesnet.com		X			X	X	X		X		
www.diabetes-normalsugars.com							X				
www.diabetesonline.com										X	
www.diabetesstore.com	X				X	X	X				
www.diabetes-supply.com					X	X	X				
www.diabetic.com	X	X				X	X	X	X	X	
www.diabeticcare.com	X	X				X	X				
www.diabeticdiet.net	X										
www.diabeticdrugstore.com	X				X						X
www.diabeticexpress.com	X	X	X		X	X					
www.diabeticsupply.com				X	X	X					
www.diseases-explained.com							X				
www.ezdiabetes.com	X		X				X				
www.hamotdiabetes.com		X	X		X	X	X				X
www.medicapharma.com									X		
www.medicool.com		X	X		X	X		X			X
COUNT	7	11	5	3	15	16	20	4	6	3	3

Sff: Sugar Free Food, **Nut:** Nutritions (Herbs, Vitamines, Minerals), **Skn:** Skin Care, **Ins:** Insulin
GIT: Glucose Monitoring (Meters, Strips, Swabs, Lancets), **Bks:** Books, **InT:** Insulin Tools (Syringes, Pumps, Wallets).

Appendix E – Semantic Types used in MMTx

Abbreviation	Unique Identifier (TUI)	Full Name
aapp	T116	Amino Acid, Peptide, or Protein
acab	T020	Acquired Abnormality
acty	T052	Activity
aggp	T100	Age Group
alga	T003	Alga
amas	T087	Amino Acid Sequence
amph	T011	Amphibian
anab	T190	Anatomical Abnormality
anim	T008	Animal
anst	T017	Anatomical Structure
antb	T195	Antibiotic
arch	T194	Archaeon
bacs	T123	Biologically Active Substance
bact	T007	Bacterium
bdsu	T031	Body Substance
bdsy	T022	Body System
bhvr	T053	Behaviour
biof	T038	Biological Function
bird	T012	Bird
blor	T029	Body Location or Region
bmod	T091	Biomedical Occupation or Discipline
bodm	T122	Biomedical or Dental Material
bpoc	T023	Body Part, Organ, or Organ Component
bsoj	T030	Body Space or Junction
carb	T118	Carbohydrate
celc	T026	Cell Component
celf	T043	Cell Function
cell	T025	Cell
cgab	T019	Congenital Abnormality
chem	T103	Chemical
chvf	T120	Chemical Viewed Functionally

chvs	T104	Chemical Viewed Structurally
clas	T185	Classification
clna	T201	Clinical Attribute
clnd	T200	Clinical Drug
cnce	T077	Conceptual Entity
comd	T049	Cell or Molecular Dysfunction
crbs	T088	Carbohydrate Sequence
diap	T060	Diagnostic Procedure
dora	T056	Daily or Recreational Activity
dsyn	T047	Disease or Syndrome
edac	T065	Educational Activity
eehu	T069	Environmental Effect of Humans
eico	T111	Eicosanoid
elii	T196	Element, Ion, or Isotope
emod	T050	Experimental Model of Disease
emst	T018	Embryonic Structure
enty	T071	Entity
enzy	T126	Enzyme
evnt	T051	Event
famg	T099	Family Group
ffas	T021	Fully Formed Anatomical Structure
fish	T013	Fish
fndg	T033	Finding
fngs	T004	Fungus
food	T168	Food
ftcn	T169	Functional Concept
genf	T045	Genetic Function
geoa	T083	Geographical Area
gngm	T028	Gene or Genome
gora	T064	Governmental or Regulatory Activity
grpa	T102	Group Attribute
grup	T096	Group
hcpp	T068	Human-caused Phenomenon or Process
hcro	T093	Health Care Related Organisation
hlca	T058	Health Care Activity
hops	T131	Hazardous or Poisonous Substance

horm	T125	Hormone
humn	T016	Human
idcn	T078	Idea or Concept
imft	T129	Immunological Factor
inbe	T055	Individual Behaviour
inch	T197	Inorganic Chemical
inpo	T037	Injury or Poisoning
inpr	T170	Intellectual Product
invt	T009	Invertebrate
irda	T130	Indicator, Reagent, or Diagnostic Aid
lang	T171	Language
lbpr	T059	Laboratory Procedure
lbtr	T034	Laboratory or Test Result
lipd	T119	Lipid
mamm	T015	Mammal
mbrt	T063	Molecular Biology Research Technique
mcha	T066	Machine Activity
medd	T074	Medical Device
menp	T041	Mental Process
mnob	T073	Manufactured Object
mobd	T048	Mental or Behavioural Dysfunction
moft	T044	Molecular Function
mosq	T085	Molecular Sequence
neop	T191	Neoplastic Process
nnon	T114	Nucleic Acid, Nucleoside, or Nucleotide
npop	T070	Natural Phenomenon or Process
nsba	T124	Neuroreactive Substance or Biogenic Amine
nusq	T086	Nucleotide Sequence
ocac	T057	Occupational Activity
ocdi	T090	Occupation or Discipline
opco	T115	Organophosphorus Compound
orch	T109	Organic Chemical
orga	T032	Organism Attribute
orgf	T040	Organism Function
orgm	T001	Organism

orgt	T092	Organisation
ortf	T042	Organ or Tissue Function
patf	T046	Pathological Function
phob	T072	Physical Object
phpr	T067	Phenomenon or Process
phsf	T039	Physiological Function
phsu	T121	Pharmacological Substance
plnt	T002	Plant
podg	T101	Patient or Disabled Group
popg	T098	Population Group
prog	T097	Professional or Occupational Group
pros	T094	Professional Society
qlco	T080	Qualitative Concept
qnco	T081	Quantitative Concept
rcpt	T192	Receptor
rept	T014	Reptile
resa	T062	Research Activity
resd	T075	Research Device
rich	T006	Rickettsia or Chlamydia
rnlw	T089	Regulation or Law
sbst	T167	Substance
shro	T095	Self-help or Relief Organisation
socb	T054	Social Behaviour
sosy	T184	Sign or Symptom
spco	T082	Spatial Concept
strd	T110	Steroid
tisu	T024	Tissue
tmco	T079	Temporal Concept
topp	T061	Therapeutic or Preventive Procedure
virs	T005	Virus
vita	T127	Vitamin
vtbt	T010	Vertebrate

Appendix F - The Application Ontology in OWL Format

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:sn="http://www.umlsLex.nlm.nih.gov/SemNet#"
   xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Centrally_Acting_Agent">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Antihypertensive_Agent"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Charity">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Service"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Company">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Organization"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Consultation">
   <rdfs:subClassOf>
    <owl:Class rdf:about="#Service"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Flaxseed">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Nutrition"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Phone">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Common"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Proceeding">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Publication"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Medical_Device">
   <rdfs:subClassOf>
```

```
    <owl:Class rdf:ID="Product"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Low_Carb">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Dietary"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="Special_Food"/>
    <owl:Class rdf:ID="Recipe"/>
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="Credentials">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Local_Group">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Organization"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Membership">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Service"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Person">
  <owl:disjointWith>
    <owl:Class rdf:about="#Organization"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Product"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Service"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Agency">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Organization"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Email">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Beta_Blocker">
  <rdfs:subClassOf>
```

```
    <owl:Class rdf:about="#Antihypertensive_Agent"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Newsletter">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Publication"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Travel_Kit">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Product"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Skin_care">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Medicine"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Video">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Publication"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Portal">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Service"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Service">
  <owl:disjointWith rdf:resource="#Person"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Product"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Organization"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Weight">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Educator">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Protein">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Nutrition"/>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Ace_Inhebitor">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Antihypertensive_Agent"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Mistletoe">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Herb"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Cholesterol">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Nutrition"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Lancet">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Medical_Accessory"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Dietary">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Special_Food"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Vitamin">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Special_Food"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Institute">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Organization"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Book">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Publication"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Publication">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Product"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ISBN not shared between all publications!  Use id instead.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Workshop">
```

```
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Event"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Calcium_Channel_Blocker">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Antihypertensive_Agent"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Sugar">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Carbohydrate"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Link">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Location">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Time">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Nebulizer">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Medical_Delivery_Device"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Enterprise">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Organization"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Rauwolfia">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Herb"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Fat">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Nutrition"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Medicine"/>
    <owl:Class rdf:about="#Medical_Device"/>
  </owl:unionOf>
```

```
</owl:Class>
<owl:Class rdf:ID="Meeting">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Event"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Diuretic">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Antihypertensive_Agent"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Special_Food">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Product"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Herb">
  <rdfs:subClassOf rdf:resource="#Special_Food"/>
</owl:Class>
<owl:Class rdf:ID="Size">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Glucose_Meter_Solution">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Medical_Accessory"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Coenzyme">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Nutrition"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Capacity">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Author">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Name">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:about="#Medical_Accessory">
  <rdfs:subClassOf rdf:resource="#Medical_Device"/>
</owl:Class>
<owl:Class rdf:ID="Angiotensin_Receptor_Blocker">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Antihypertensive_Agent"/>
  </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:ID="Conference">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Event"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Syringe">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Medical_Delivery_Device"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Carbohydrate">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Nutrition"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Price">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Sugar_Free">
  <rdfs:subClassOf rdf:resource="#Dietary"/>
</owl:Class>
<owl:Class rdf:about="#Event">
  <rdfs:subClassOf rdf:resource="#Service"/>
</owl:Class>
<owl:Class rdf:ID="Calcuim">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Nutrition"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Volume">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Course">
  <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>
<owl:Class rdf:ID="Research">
  <rdfs:subClassOf rdf:resource="#Service"/>
</owl:Class>
<owl:Class rdf:ID="Aerochaber">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Medical_Monitoring_Device"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Journal">
  <rdfs:subClassOf rdf:resource="#Publication"/>
</owl:Class>
<owl:Class rdf:ID="Program">
```

```
  <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>
<owl:Class rdf:ID="Patient">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Webmaster">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:about="#Organization">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:resource="#Service"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="provide_service"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Person"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Product"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Service"/>
</owl:Class>
<owl:Class rdf:about="#Medical_Delivery_Device">
  <rdfs:subClassOf rdf:resource="#Medical_Device"/>
</owl:Class>
<owl:Class rdf:ID="Peak_Flow_Meter">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Medical_Monitoring_Device"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Mineral">
  <rdfs:subClassOf rdf:resource="#Special_Food"/>
</owl:Class>
<owl:Class rdf:ID="Software">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Product"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Physician">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Report">
  <rdfs:subClassOf rdf:resource="#Publication"/>
</owl:Class>
<owl:Class rdf:ID="Coleus_Forkohlii">
  <rdfs:subClassOf rdf:resource="#Herb"/>
```

```
</owl:Class>
<owl:Class rdf:ID="Insulin_Pump">
  <rdfs:subClassOf rdf:resource="#Medical_Delivery_Device"/>
</owl:Class>
<owl:Class rdf:about="#Product">
  <owl:disjointWith rdf:resource="#Organization"/>
  <owl:disjointWith rdf:resource="#Service"/>
  <owl:disjointWith rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Ingredient">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Glucose_Monitoring_Meter">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Medical_Monitoring_Device"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Magazine">
  <rdfs:subClassOf rdf:resource="#Publication"/>
</owl:Class>
<owl:Class rdf:ID="Statin">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Antihypertensive_Agent"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Nutrition">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Hathorne">
  <rdfs:subClassOf rdf:resource="#Herb"/>
</owl:Class>
<owl:Class rdf:ID="Family_Member">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Strip">
  <rdfs:subClassOf rdf:resource="#Medical_Accessory"/>
</owl:Class>
<owl:Class rdf:ID="ID_Bracelet">
  <rdfs:subClassOf rdf:resource="#Product"/>
</owl:Class>
<owl:Class rdf:ID="Measurement_Unit">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:ID="Vitamin_C">
  <rdfs:subClassOf rdf:resource="#Nutrition"/>
</owl:Class>
<owl:Class rdf:ID="Insulin">
  <rdfs:subClassOf>
```

```
    <owl:Class rdf:about="#Medicine"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Recipe">
  <rdfs:subClassOf rdf:resource="#Product"/>
</owl:Class>
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Service"/>
    <owl:Class rdf:about="#Product"/>
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="Class">
  <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>
<owl:Class rdf:ID="Support_Group">
  <rdfs:subClassOf rdf:resource="#Organization"/>
</owl:Class>
<owl:Class rdf:ID="Fat_Free">
  <rdfs:subClassOf rdf:resource="#Dietary"/>
</owl:Class>
<owl:Class rdf:ID="Instructor">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Speaker">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Education">
  <rdfs:subClassOf rdf:resource="#Service"/>
</owl:Class>
<owl:Class rdf:ID="Seminar">
  <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>
<owl:Class rdf:ID="Article">
  <rdfs:subClassOf rdf:resource="#Publication"/>
</owl:Class>
<owl:Class rdf:ID="University">
  <rdfs:subClassOf rdf:resource="#Organization"/>
</owl:Class>
<owl:Class rdf:ID="Practitioner">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Date">
  <rdfs:subClassOf rdf:resource="#Common"/>
</owl:Class>
<owl:Class rdf:about="#Antihypertensive_Agent">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Medicine"/>
```

```
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Student">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="Glucose_Related">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Medicine"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Donation">
  <rdfs:subClassOf rdf:resource="#Service"/>
</owl:Class>
<owl:Class rdf:about="#Medicine">
  <rdfs:subClassOf rdf:resource="#Product"/>
</owl:Class>
<owl:Class rdf:about="#Medical_Monitoring_Device">
  <rdfs:subClassOf rdf:resource="#Medical_Device"/>
</owl:Class>
<owl:Class rdf:ID="Insulin_Pump_Cartridge">
  <rdfs:subClassOf rdf:resource="#Medical_Accessory"/>
</owl:Class>
<owl:Class rdf:ID="Fiber">
  <rdfs:subClassOf rdf:resource="#Carbohydrate"/>
</owl:Class>
<owl:Class rdf:ID="Campain">
  <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>
<owl:ObjectProperty rdf:about="#provide_service">
  <rdfs:range rdf:resource="#Service"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Price">
  <rdfs:range rdf:resource="#Price"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Product"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Journal">
  <rdfs:domain rdf:resource="#Article"/>
  <rdfs:range rdf:resource="#Journal"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="uses_Strips">
  <rdfs:domain rdf:resource="#Glucose_Monitoring_Meter"/>
  <rdfs:range rdf:resource="#Strip"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_brandName">
  <rdfs:range rdf:resource="#Name"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="has_Time">
  <rdfs:range rdf:resource="#Time"/>
  <rdfs:domain rdf:resource="#Event"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="povide_Event">
  <rdfs:domain rdf:resource="#Education"/>
  <rdfs:range rdf:resource="#Event"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Email">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Organization"/>
        <owl:Class rdf:about="#Person"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#Email"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="is_produced_by">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Organization"/>
        <owl:Class rdf:about="#Person"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:domain rdf:resource="#Product"/>
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:ID="is_made_by"/>
  </owl:equivalentProperty>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Publisher">
  <rdfs:range rdf:resource="#Company"/>
  <rdfs:domain rdf:resource="#Publication"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Size">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Glucose_Meter_Solution"/>
        <owl:Class rdf:about="#Insulin_Pump_Cartridge"/>
        <owl:Class rdf:about="#Medicine"/>
      </owl:unionOf>
```

```
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#Size"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Capacity">
  <rdfs:domain rdf:resource="#Insulin_Pump_Cartridge"/>
  <rdfs:range rdf:resource="#Capacity"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="is_member_of">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Local_Group"/>
        <owl:Class rdf:about="#Support_Group"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Services">
  <rdfs:domain rdf:resource="#Membership"/>
  <rdfs:range rdf:resource="#Service"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Organization">
  <rdfs:range rdf:resource="#Organization"/>
  <rdfs:domain rdf:resource="#Event"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="provide_product">
  <rdfs:domain rdf:resource="#Organization"/>
  <rdfs:range rdf:resource="#Product"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="is_sponsered_by">
  <rdfs:range rdf:resource="#Organization"/>
  <rdfs:domain rdf:resource="#Research"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#is_made_by">
  <owl:equivalentProperty rdf:resource="#is_produced_by"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Organization"/>
        <owl:Class rdf:about="#Person"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="has_Nutrition">
  <rdfs:range rdf:resource="#Nutrition"/>
  <rdfs:domain rdf:resource="#Recipe"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="sell_product">
  <rdfs:range rdf:resource="#Product"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="provide_Publication">
  <rdfs:range rdf:resource="#Publication"/>
  <rdfs:domain rdf:resource="#Education"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="results_In">
  <rdfs:range rdf:resource="#Time"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Glucose_Monitoring_Meter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="is_provided_by">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Organization"/>
        <owl:Class rdf:about="#Person"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:domain rdf:resource="#Service"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Ingredient">
  <rdfs:range rdf:resource="#Ingredient"/>
  <rdfs:domain rdf:resource="#Recipe"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Cost">
  <rdfs:range rdf:resource="#Price"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Event"/>
        <owl:Class rdf:about="#Membership"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_Authors">
  <rdfs:domain rdf:resource="#Publication"/>
  <rdfs:range rdf:resource="#Author"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="work_at">
```

```
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range>
   <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
     <owl:Class rdf:about="#Agency"/>
     <owl:Class rdf:about="#Company"/>
     <owl:Class rdf:about="#Enterprise"/>
     <owl:Class rdf:about="#Institute"/>
     <owl:Class rdf:about="#University"/>
    </owl:unionOf>
   </owl:Class>
  </rdfs:range>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="has_Subscription">
  <rdfs:domain rdf:resource="#Membership"/>
  <rdfs:range rdf:resource="#Publication"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="has_Phone">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="#Phone"/>
  <rdfs:domain>
   <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
     <owl:Class rdf:about="#Organization"/>
     <owl:Class rdf:about="#Person"/>
    </owl:unionOf>
   </owl:Class>
  </rdfs:domain>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="has_Link">
  <rdfs:range rdf:resource="#Link"/>
  <rdfs:domain rdf:resource="#Portal"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="has_Sample_Size">
  <rdfs:domain rdf:resource="#Glucose_Monitoring_Meter"/>
  <rdfs:range rdf:resource="#Size"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="has_Description">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Recipe"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="date">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Date"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="has_Type">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

```
  <rdfs:domain rdf:resource="#Insulin"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="zip">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Location"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="time">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Time"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_Onset">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Insulin"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_URL">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
  <rdfs:domain rdf:resource="#Link"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sMonth">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Date"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="city">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Location"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="year">
  <rdfs:domain rdf:resource="#Date"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="weight">
  <rdfs:domain rdf:resource="#Weight"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="email">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Email"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tui">
```

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain>
   <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
     <owl:Class rdf:about="#Common"/>
     <owl:Class rdf:about="#Organization"/>
     <owl:Class rdf:about="#Person"/>
     <owl:Class rdf:about="#Product"/>
     <owl:Class rdf:about="#Service"/>
    </owl:unionOf>
   </owl:Class>
  </rdfs:domain>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="country">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Location"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_Duration">
  <rdfs:domain rdf:resource="#Insulin"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="second">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Time"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_Number_Of_Pages">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Book"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="id">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Product"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="volume">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#Volume"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_Quantity">
  <rdfs:domain>
   <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
```

```
        <owl:Class rdf:about="#Strip"/>
        <owl:Class rdf:about="#Insulin_Pump_Cartridge"/>
        <owl:Class rdf:about="#Syringe"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_Topic">
  <rdfs:domain rdf:resource="#Event"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="cedentials">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Credentials"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="has_Location">
  <rdfs:range rdf:resource="#Location"/>
  <rdfs:domain rdf:resource="#Event"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="Test_Range">
  <rdfs:domain rdf:resource="#Glucose_Monitoring_Meter"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="is_retrieved_from">
  <rdfs:domain rdf:resource="#Product"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="#Link"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Credentials">
  <rdfs:range rdf:resource="#Credentials"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_End_Date">
  <rdfs:domain rdf:resource="#Event"/>
  <rdfs:range rdf:resource="#Date"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Preperation_Time">
  <rdfs:range rdf:resource="#Time"/>
  <rdfs:domain rdf:resource="#Recipe"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
```

```
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="name">
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 <rdfs:domain rdf:resource="#Name"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="value">
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 <rdfs:domain rdf:resource="#Price"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Unit">
 <rdfs:range rdf:resource="#Measurement_Unit"/>
 <rdfs:domain>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Ingredient"/>
    <owl:Class rdf:about="#Nutrition"/>
    <owl:Class rdf:about="#Special_Food"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:domain>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="Number_of_Tests">
 <rdfs:domain rdf:resource="#Glucose_Monitoring_Meter"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="line2">
 <rdfs:domain rdf:resource="#Location"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="line1">
 <rdfs:domain rdf:resource="#Location"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="capacity">
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 <rdfs:domain rdf:resource="#Capacity"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Date">
 <rdfs:range rdf:resource="#Date"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >Date of Production</rdfs:comment>
```

```
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Product"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Start_Date">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Event"/>
  <rdfs:range rdf:resource="#Date"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Peak">
  <rdfs:domain rdf:resource="#Insulin"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Number_Of_Servings">
  <rdfs:domain rdf:resource="#Recipe"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Start_Page">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Article"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Edition_Number">
  <rdfs:domain rdf:resource="#Book"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_End_Page">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Article"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Address">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Organization"/>
        <owl:Class rdf:about="#Person"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#Location"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="month">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
```

```
  <rdfs:domain rdf:resource="#Date"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hour">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#Time"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="currency">
  <rdfs:domain rdf:resource="#Price"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#byte"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="day">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Date"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="unit">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Size"/>
        <owl:Class rdf:about="#Volume"/>
        <owl:Class rdf:about="#Weight"/>
        <owl:Class rdf:about="#Capacity"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_subject">
  <rdfs:domain rdf:resource="#Publication"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="minute">
  <rdfs:domain rdf:resource="#Time"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="has_Name">
  <rdfs:range rdf:resource="#Name"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Organization"/>
        <owl:Class rdf:about="#Person"/>
```

```
        <owl:Class rdf:about="#Product"/>
        <owl:Class rdf:about="#Ingredient"/>
        <owl:Class rdf:about="#Link"/>
       </owl:unionOf>
     </owl:Class>
   </rdfs:domain>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >online</rdf:first>
    <rdf:rest rdf:parseType="Resource">
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >print</rdf:first>
    </rdf:rest>
  </owl:oneOf>
</owl:DataRange>
<owl:AllDifferent/>
<Measurement_Unit rdf:ID="oz"/>
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <Measurement_Unit rdf:about="#oz"/>
  </owl:distinctMembers>
</owl:AllDifferent>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:parseType="Resource">
      <rdf:rest rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >k</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >ml</rdf:first>
          </rdf:rest>
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >u</rdf:first>
        </rdf:rest>
      </rdf:rest>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >g</rdf:first>
    </rdf:rest>
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >kilo</rdf:first>
  </owl:oneOf>
```

```
  </owl:DataRange>
</rdf:RDF>
```