Finding Hidden Semantics of Text Tables

SALEH A. ALRASHED

Ph.D. 2004

Cardiff University
Wales

Cardiff School of Computer Sciences

Finding Hidden Semantics of Text Tables

Saleh A. Alrashed

Doctor of Philosophy in Computer Science
August 2004

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed............. Saleh A. Al-Rashed ....(candidate)

Date......28/9/2004.........

Statement 1

This thesis is the result of my own investigation, except where otherwise stated. Other sources are acknowledged by explicit references. A bibliography is appended.

Signed.............. Saleh A. Al-Rashed ....(candidate)

Date......28/9/2004..........

Statement 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for interlibrary loan, and for the title and summary to be made available to outside organisations.

Signed............. Saleh A. Al-Rashed ....(candidate)

Date......28/9/2004.........

Abstract

Combining data from different sources for further automatic processing is often hindered by differences in the underlying semantics and representation. Therefore when linking information presented in documents in tabular form with data held in databases, it is important to determine as much information about the table and its content. Important information about the table data is often given in the text surrounding the table in that document. The table's creators cannot clarify all the semantics in the table itself therefore they use the table context or the text around it to give further information. These semantics are very useful when integrating and using this data, but are often difficult to detect automatically. We propose a solution to part of this problem based on a domain ontology. The input to our system is a document that contains tabular data and the system aims to find semantics in the document that are related to the tabular data. The output of our system is a set of detected semantics linked to the corresponding table. The system uses elements of semantic detection, semantic representation, and data integration.

Semantic detection uses a domain ontology, in which we store concepts of that domain. This allows us to analyse the content of the document (text) and detect context information about the tables present in a document containing tabular data. Our approach consists of two components: (1) extract, from the domain ontology, concepts, synonyms, and relations that correspond to the table data. (2) Build a tree for the paragraphs and use this tree to detect the hidden semantics by searching for words matching the extracted concepts. Semantic representation techniques then allow representation of the detected semantics of the table data.

Our system represents the detected semantics, as either 'semantic units' or 'enhanced metadata'. Semantic units are a flexible set of meta-attributes that describe the meaning of the data item along with the detected semantics. In addition, each semantic unit has a concept label associated with it that specifies the relationship between the unit and the real world aspects it describes. In the enhanced metadata, table metadata is enhanced with the semantics and representation context found in the text. Integrating data in our proposed system takes place in two steps. First, the semantic units are converted to a common context, reflecting the application. This is achieved by using appropriate conversion functions. Secondly, the semantically identical semantic units, will be identified and integrated into a common representation. This latter is the subject of future work.

Thus the research has shown that semantics about a table are in the text and how it is possible to locate and use these semantics by transforming them into an appropriate form to enhance the basic table metadata.

Acknowledgements

I would like to start by praising God the most gracious most merciful for providing me with the faith to undertake this work.

- It is a great pleasure to express my sincere thanks to my supervisors, Professor W.A. Gray and Professor N. Fiddian, for their expert guidance and encouragement throughout the period of my research.
- Thanks also are due to other members of the school for their help, especially Mrs. M. Evans and Mrs H. Williams.
- My sincere thanks also go to the Royal Saudi Air Force, Saudi Arabia for their financial support which enabled me to continue my studies.
- Last, but certainly not least, I am indebted to my parents and my family, especially my wife, for her endurance and outstanding support which provided vital encouragement. Finally, I would like to mention my beloved children, Haela, Ali, Alia, and Abdullah who given me happiness during the difficult period of my postgraduate research.

Acronyms

AI	Artificial Intelligent	4.2
ASCII	American Standard Code for Information Interchange	6.5
CF	Conversion Function	6.4
DAML	DARPA Agent Markup Language	4.3
DAMLONT	DAML ONTology	4.3
DARPA	The Defence Advanced Research Projects Agency	4.3
DB	Data Base	3.4
DBMS	Data Base Management System	3.3
DL	Description Logic	4.3
DM	Document Mining	2.3
EBCDIC	Extended Binary Coded Decimal Interchange Code	6.5
ECF	Elementary Conversion Function	6.4
GALEN	Generalised Arch. for Languages, Encyclopaedias, and Nomenclatures in Medicine	4.2
GATE	General Architecture for Text Engineering	2.3
GBP	British Pound	6.4
HP	Horsepower	7.2
IE	Information Extraction	2.1
IR	Information Retrieval	1.3
IRS	Information Retrieval System	3.4
KB	Knowledge Base	5.3
KDD	Knowledge Discovery in Database	2.3
KDT	Knowledge Discovery in Textual documents	2.3
KIF	Knowledge Interchange Format	4.3

KR	Knowledge Representation	4.3
KSTEM	Krovetz Stemming	2.3
MCF	Multivalued Conversion Function	6.5
MIX	Metadata-based Integration model for data X-change	2.2
NLP	Natural Language Processing	3.1
OCML	Operational Conceptual Modelling	4.3
OCR	Optical Character Recognition	2.3
OIL	Ontology Interchange Language	4.3
ONTODM	ONTOlogy for the construction of Domain Models	4.2
RDBMS	Relational DBMS	3.4
RDF	Resource Description Framework	4.3
RDFS	RDF Schema	4.3
RT	Representational Type	6.4
SA	Semantic Attribute	5.6
SHOE	Simple HTML Ontology Extension	4.3
SNOMED	Systematised Nomenclature of Human Medicine	4.2
SQL	Structured Query Language	3.2
SRD	Semantic and Representation Detection framework	1.1
SU	Semantic Unit	5.6
TINTIN	Table INformation -based Text INquiry	2.2
TOVE	TOronto Virtual Enterprise	4.2
VAT	Value Add Tax	3.3
VITAL	Visualization and Implementation of Temporal Action Logics	4.3
W3C	World Wide Web Consortium	4.3
XML	eXtensible Markup Language	4.3
XOL	XML-Based Ontology	4.3

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

Documents are one of the most important ways of sharing knowledge between humans. They are constructed using some common assumptions about their structure. Authors intend to convey information in ways allowing readers accurate and effective interpretation of the contents. This is why understanding documents is a relatively easy task for an intelligent human reader. One of the ways that authors use to present information in documents is tables.

The number of tables used per page in scientific papers has increased quite steadily over time. It has grown from 9% of the pages in 1984 up to 32% in 1997 [Hur00]. Document tables have been created by humans to aid understanding of the information, therefore any attempts to reuse their content automatically needs effort to recognise and determine the table's structure and semantics.

Because of the large number of documents that have been published electronically, there is a real need to reuse the contents of these types of documents in investigations. This implies a need for automatic analysis of documents to aid human users. As an important part of a document, tables have received attention from researchers trying to locate, analyse, identify and transform them into reusable formats for further analysis by software systems [HD95, PC97, YTT01, RHI01]. Most researchers have concentrated on the table itself, identifying its physical and logical structure, without considering the relation between a table and the surrounding text in the document [Niy94].

The problem with isolating tabular data that appears in textual documents from the rest of the text, is that although the table data can be extracted and reused, it is not possible to fully understand its contents and reuse it effectively in other integration processes unless this data has been combined with parts of the text in the same document that are related to that table. This text describes the semantics of the information in the table.

This problem arises for two main reasons. Firstly, the author of the document tends to explain parts of the table in the text around it, and the information in the table does not make sense if the table is completely isolated (semantically) from its document. Even if there are no explanations about the table in the text, the table can not be completely isolated from its domain, especially if it is going to be integrated with other tables without losing some of its usefulness. Secondly, locating all data with related description of the semantics in the table structure makes the table difficult to understand as it extends the size of the table and affects the clarity of presentation. Therefore authors tend to leave parts of the meaning of data to be explained in the surrounding text.

The aim of our research is to develop techniques for detecting, extracting and representing table semantics that are buried in the text surrounding that table. It is essential that these techniques are as flexible as possible, and generic, in that they offer support for detecting semantics related to tables in documents from different domains, and are not confined to use in a single application area. Because we are dealing with table semantics which can be expressed in a variety of ways in natural language using synonymous expressions, a domain ontology will be needed to enrich the selection mechanism with alternative semantics related to the table's metadata. This enrichment will be used to improve the discovery of appropriate semantics in the document text.

When integrating tables stored in structured database systems, it is common to use the mapping and constraints for a table held in its metadata description, to help users see the data that can be linked when accessing tables to combine their contents. Also, database designers follow certain rules in creating tables with a DBMS to ensure the consistency of data within and between tables. Therefore, these constraints and rules help overcome some of the heterogeneity problems which occur when integrating tables derived from the same data source, but having different representations. However, in textual documents, this type of information is not usually available within the table itself. Although, it is often the case that other types of information, constraints, semantics, or rules which will help with this integration are buried in a variety of formats in the text surrounding a table in a document.

In this thesis, we present and investigate approaches which enable semantics related to a table, and buried within the document's text, to be detected, extracted and represented in a suitable form for use in understanding and using the table and its contents. Techniques presented are for documents held in ASCII format. To demonstrate and evaluate these approaches, a prototype system has been developed which we call SRD (Semantics and Representation Detection system). SRD is not intended as an end product, but as a test prototype software system, which can be used to demonstrate and test our ideas. We have tested our prototype system by applying it to 300 documents extracted from the web. These experiments were designed to determine the significance of the approaches used and to compare alternative techniques.

In the rest of this chapter we discuss the motivations, objectives and achievements of our research.

1.2 Research Motivations

The research presented in this thesis was motivated by two PhD theses. The first was by L. E. Hodge [Hod01]. He concentrated in his work on the development of approaches that enable tabular data appearing within semi-structured documents to be detected and reused in wider contexts. The main focus of his research was the development of techniques for detecting and reusing tabular information appearing in plain text. He developed a three-stage approach involving the location, analysis, and transformation of tables. At the end of his thesis in the future work section, he mentioned that metadata (semantics) relating to table content can be extracted from the text that accompanies tables, and he said that investigation into new techniques for detecting such semantics would be desirable if effective utilisation of metadata and table contents is to be achieved. This was a recognition that the metadata in the table was limited and needed enhancement by information given in the text.

The second thesis was by Mathew Hurst [Hur00]. He also concentrated on extracting tables from text, but with a different model. His model contained graphical, physical, structural, functional and semantic components (see Section 2.2.2.2 for more detail). He stated that working with table semantics in text will always be incomplete until the table held metadata is joined with the information in the text of the document. He also said " No work to date looks at the content of the document as a whole" [Hur00]. These issues motivated our investigation into approaches that would analyse the table's accompanying text to detect related semantics, and so allow tabular information to be used as a complete unit, instead of isolating the table itself, and looking at it out of its complete context.

1.3 Objectives

The amount of online structured, semi-structured and unstructured data is growing rapidly [LRO96]. The reasons for this are the growth of e-commerce, e-services, e-government and e-library. These are areas which publish a very large number of documents on the Internet, and many of these documents contain tabular data. Analysis and extraction of information from these types of semi-structured data has increased especially in the area of Information Retrieval (IR) [SL97]. When we consider semi-structured documents, in the majority of cases their content is not in an appropriate form for reuse in other contexts (e.g. descriptive text). This is not true of tabular data, and a number of research projects have analysed and transformed the content of such tabular data into reusable formats.

Currently, these systems always look at tabular data as an isolated unit from the document, and this overlooks an important part of the information related to a table which is held in the text. If these related semantics are not extracted, a semantic conflict or semantic heterogeneity and representational conflict might occur when integrating data from these tables, which invalidates the use of the data. In order to resolve problems caused by semantic conflict, an information system must be able to ensure the semantic interoperability by discovering and utilising this contextual information. The context of a piece of data in a table is the metadata relating to its meaning [GBMS99]. Context information can be quite varied in form, e.g. it includes information such as a unit's specification (e.g. currency, length). To extract useful context information, we need to know what kind of context information is needed for a specific domain to resolve conflicts, and how to use it effectively in this task. Therefore if a knowledge base or ontology is combined with the extraction system we will ensure the detection of related context within a document.

The hypothesis of this thesis is based on the assumption that when a table appears in a document, there is additional description information about the table in the document's text which defines and elaborates the meaning of the information held in the table. This contains more detailed information than the descriptions held in the table itself.

The hypothesis is that this extra information can be located, extracted and used to improve the use and understanding of the table's content when linking the table contents with other information.

This means, we can develop tools that can link tabular data with the text describing its content in the same document, possibly using an ontology to assist in this process. This will elaborate the meaning of the information held in the table.

Thus, the aim of our research is to investigate how the content of a table appearing within a document can be used to detect the hidden and related semantics about the table in the surrounding text. Our hypothesis splits into five objectives for our research.

1.3.1 Objective 1 : To demonstrate that semantics about the table exist in the surrounding text

The table semantics can be described as all the information related to that table. In the case of tabular data presented in a document, the table semantics include information, and descriptions about the tabular data in the document text. Any information describing or elaborating the table content and presented in the surrounding text is an important part of the table's semantics. We believe, most authors describe table's meaning within the text of that document. Therefore one of our objectives in this research is to investigate whether this type of information does exist in the text surrounding the table.

1.3.2 Objective 2 : To show that this information can enrich the description of the table

It is difficult, because of the structure of table, to completely describe the table metadata or any other information related to that table in the table itself. Also, we know that some of table metadata needs fuller explanations. Therefore information related to metadata in the surrounding text is used to describe and explain the meaning of such metadata. For example, a table about cars has a field called "price". This field has some important descriptions like "currency" and "VAT" which can not be described fully in the table, and can not be ignored if we are to fully understanding its role. Authors most likely are going to explain this type of data more fully in the surrounding text. By extracting information of this type, we can enrich the semantics of the table. Our objective is thus to detect and extract the beneficial semantics related to a table in its surrounding text.

1.3.3 Objective 3 : To show that this semantic information can be transformed into a usable representation for further processing

The semantics in the surrounding text might be related to the table metadata directly in the text and, in this case, a simple word searching mechanism can be used. In many cases, the buried semantics are related to a description of the table metadata, and not to the metadata directly. Therefore, approaches used have to be able to augment the table header with synonyms and related semantics. The extracted semantics then have to be represented in a format that allows the information to be reused in an efficient way without any loss of information. There are two ways of representing detected semantics. Either in a stand alone format or by combining it with the table itself.

1.3.4 Objective 4 : To show how these semantics can be used to link tabular data more effectively with other data

Heterogeneity between table representations that have been extracted from different documents is highly likely to occur because of the differences in the author's knowledge and perspective. Enriching a table with semantics from the surrounding text will help to overcome some of the problems that can be caused by this heterogeneity. The integration system must be able to detect heterogeneity in table semantics, and then resolve some of the differences where appropriate by applying suitable conversion functions to bring values into a common representation form. These functions convert the representation of values to other formats. Our objective here is to assure consistency between the semantics of tables that are going to be integrated so that it is possible to use this information when integrating tables. Although this stage was not fully implemented a suitable in this project, we implemented the semantic detection algorithm and have considered in chapter 7 how such conversion functions could be implemented.

1.3.5 Objective 5 : To show that a generic approach can be developed for locating this information

Any system that addresses these semantic problems should be generic and work with different document domains. A suitable domain ontology which describes the concepts of the current domain has to be provided for the system to achieve this. The generality will come from the ability to change the domain ontology to suit the current problem, without affecting the structure of the system. Also, generality should be covered from a point of view where the system is able to deal with semantics in different representations. Our objective here is to ensure the generality of the approaches used to detect and extract the related semantics from the document text.

1.3.6 Objective 6 : Critically Evaluate results

By developing a prototype system, creating an experiment to evaluate the system results and using data analysis techniques to show the significance of the approaches used, we will evaluate their applicability, strengths and weaknesses.

Based on the above background and objectives, our research is based on some assumptions, namely that it is possible to locate tables within documents, analyse their contents and boundaries, and transform them into reusable forms. These tasks are out of the scope of this research, and we assume that the structure of the table has already been identified. This work has been presented by several authors in earlier work, [HGF98, WH02].

1.4 Achievements of Research

Based on the above objectives, our research has focused on processing approaches that would enable us to use the text surrounding a table in a textual document to detect, extract and represent related semantics in the text in an effective manner. This work resulted in the development of a prototype system - SRD, and an experimental domain to enable us to evaluate the approaches used. This system has shown it is possible to:

1.4.1 Detect useful semantics related to tabular data in a document

Using our system - SRD, we have found that there is a significant number of semantics that are related to tables presented in the document. We have applied our system to 300 documents from different data sources, and we were able to detect a significant number of semantics in each document about the tables. This showed that semantics about tables do exist in the surrounding text (see Section 9.4.1).

1.4.2 Extract and represent detected semantics

We have shown how the detected semantics can be extracted from the text. We have used two approaches to detect and represent these semantics. The first approach uses the table header or table metadata to detect the hidden semantics in text. The other approach augments the table header with synonyms and related information to the table header. Each of these approaches produced extra semantics related to the table. After extracting these semantics, we were able to represent it as either semantic-units or enhanced metadata (see Section 6.6.1). These alternative representations are useful in different contexts.

1.4.3 Create conversion functions to convert semantics between different representations

To reduce the heterogeneity occurring between different metadata for improved integration, the potentially similar attributes that have been detected must be converted to match each other in representation. We have been able to define a number of conversion functions that convert the representation of detected semantics. (see Section 7.3).

1.4.4 Using approaches in different domains

We have used two approaches to detect the hidden semantics. We have found that a domain ontology approach has given us the highest number of semantics. We found that 80.7 % of the total number of the known semantics were detected using a domain ontology, and 19.3 % without ontology (see Section 9.4.1.1).

1.4.5 Distribution of related semantics in documents

•

We found that 75.05 % of the known semantics come from the paragraphs adjacent to the table. Also, we found that 71.8 % of the known semantics appears in the paragraphs under the table. It is common that the writer of the document will describe and elaborate the table after showing it to the reader. (see Section 9.4.2.1).

1.4.6 Identify different types of semantic location identifiers

•

By analysing the types of indicators in the texts, we found that there are a number of formats that writers use to indicate a description of a table. The first and the most commonly used is referring to the table by its number, for example (Table 2.1). The other types of indicators are 'the above table, the next table, the last table, the previous table, in the table below'. Also, 'figure' is often used instead of 'table'. We also found that paragraphs that have such indicators always have semantics in them and they are always next to the indicator. We also found that, the paragraphs that have indicators have most of the known semantics in the text. Therefore in large documents, it is useful to search for the indicators first and concentrate on the paragraphs that contain them, as this approach will locate the majority of the semantics in the text. (see Section 9.4.2.3).

1.5 Thesis Organisation

This section presents an overview of the thesis organisation. This first chapter has presented an introduction to the research undertaken. The background, motivations, hypothesis, and objectives were specified. Finally, the overall research achievements were presented. A short description of each chapter follows.

Chapter 2

This chapter introduces and summarises the fields of research relevant to this thesis. A number of table and document related research studies are presented and analysed. The relation of our research to different relevant research fields such as information extraction (IE), information retrieval (IR), and ontology are examined, as they also aim to provide the capability to identify, extract, retrieve, and integrate efficiently and effectively information from documents taken from different sources. This helps to set the scene for the next three chapters which analyse the methodologies employed in our research.

Chapter 3

In this chapter we address the problem of semantics and representational conflict. We discuss table semantics and the differences between semantics and representation. Also, we address semantic heterogeneity and how to overcome this problem by using the context of the table in the document.

Chapter 4

In this chapter we overview ontologies covering the different definitions that have been used to declare the meaning of ontology. Different types of ontologies are covered and their categorisation depends on the level of generality of their context and subject of conceptualisation. Ontology representation is another aspect of ontology that affects the usability of the ontology. Also, we talk about the different schemas that knowledge representation uses. Finally, a number of representation languages are described.

Chapter 5

In this chapter we discuss our system design and representation. We present the documents used in the system. Also, we discuss the different mechanisms that can be used in collecting a table's metadata and all the related words to the metadata in the domain ontology. Searching mechanisms and extraction process are also discussed in this chapter.

Chapter 6

In this chapter we discuss the architecture of our SDR system. We also talk about the detection process which is going to detect the hidden semantics, using different approaches. After detecting the hidden semantics, they will be extracted and represented in formats called semantic units and enhanced metadata. Suitable conversion functions are used where necessary, and then the integration process is used to integrate these semantics.

Chapter 7

In this chapter we discuss the similarity relation used to find the relation between different concepts. Also, we discuss the types of conversion functions, together with their properties.

Chapter 8

In this chapter we discuss the prototype system for our framework. We start with the architecture of the prototype system. The prototype consists of three parts, data inputs, system processes and data outputs. We discuss each part and its components. Also we describe the processes that the system performs such as, keywords gathering, document representation, semantics detection and semantic representation.

Chapter 9

In this chapter, we discuss the experimental design used to evaluate the prototype and the objectives of this experiment. We also discuss the different types of test, the experiment will perform. After using the system with the test data and gathering the results of these tests, we will analyse the result using statistics analysis methods such as, t-test and Mann-Whitney test. Also, in this chapter we show the significant results in our experiment and how they match the hypothesis.

Chapter 10

In this chapter, we draw conclusions and identify any future work that could be carried out based on this research.

CHAPTER 2

Introduction

2.1 Introduction

In this chapter, a number of table and document-related research studies are presented and analysed. These studies have a focus on identifying the logical content of documents and tables. In particular, the semantic relationship between a table and its containing document is an important interest. The relationships of our research with different relevant research fields, such as information extraction (IE), information retrieval (IR), and ontology, are examined, as they also aim to provide the capability to identify, extract, retrieve, and integrate efficiently and effectively information from documents taken from different sources.

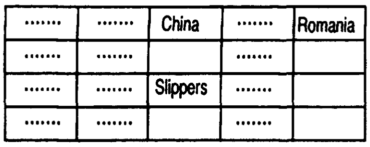
2.2 Table-related Research

As presented in Chapter 1, we concentrate in our research on tables that are presented in text documents. Tables are one of the important parts of the document. They contain a vast amount of data concentrated in a limited part of the document. Many information retrieval systems have focused on identifying and extracting information from text documents and especially tables [HROM00, SFJ02]. A number of researchers have concentrated on extracting data from documents to allow users to formulate queries about the tables on these documents [Hoc94]. Other researchers use IR techniques on documents to answer their queries. Also, there are number of researchers which concentrating on the reuse of all the data in tables by extracting and transforming the full table to a reusable format. As this research proceeded, the researchers have found that the reuse of tables from text documents would not be effective unless some type of descriptions of the meaning of table data is included in the extraction process. Therefore a number of researchers used an external resource (ontology) to facilitate the extraction of this types of table information.

2.2.1 Tables and information retrieval

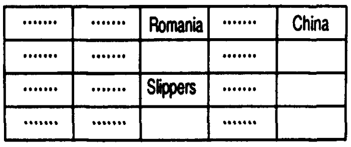
Possibly unique to the overlap of table research and IR is the TINTIN system [PC97]. The objective of this research system is to exploit the relationship between a structural phenomenon, a table, its contents, and the content of a query. There are two parts to this research, methods for identifying the tables in an unmarked document; and systems allowing a user to formulate queries which are sensitive to the particular model of the tables. The TINTIN model has two components based on the general syntactic elements of tables: captions - also known as the head of the table, and table-lines. A heuristic approach is used to recognise the tables in a document. Indexing information for the retrieval process is extracted from the caption and table line segments of a table and held as metadata.

Filah [FLdS01] extends this work with respect to the identification of the functional areas of the table and the requirements of the query processing system to identify terms desired in the data or terms desired in the index structure of the table. Significant as these enhancements would be, this proposal for the apparent functional analysis of the table relies on a template approach to the identification of the appropriate areas, which views the table as a series of uniformly labelled columns.



·······	·······	China	·······	Romania
·······	·······		·······	
·······	·······	Slippers	·······	
·······	·······		·······	

Figure 2.1: China and slippers



·······	·······	Romania	·······	China
·······	·······		·······	
·······	·······	Slippers	·······	
·······	·······		·······	

Figure 2.2: Romania and slippers

For example, one could slice a table into columns and treat each column as a separate document. The column header and content occurring together indicate more specificity and could be a source of multiple evidence for the corresponding table. For example, if the query is "Dose China exports slippers?" and we have a table with "China" and "slippers" occurring together in a column as in Figure2.1, this should receive more weight than the case in which "Romania" and "slippers" occur in one column and "China" occurs in another as in Figure2.2. Here the co-occurrence of values in a column indicates a higher relevance to the query.

2.2.1.1 Tables and their logical structure

L. E. Hodge [Hod01, HGF98] has concentrated in his work on the development of approaches that enable tabular data appearing within semi-structured documents to be located, reused and transformed into a relational format. The main focus of his research is the development of techniques for reusing tabular information appearing in plain text and linking it with database data. He has developed a three-stage approach, involving the location, analysis, and transformation of tables.

He introduced a tree model that enables the visualisation of a table's logical structure, and indicates groupings that exist within the table, and the access mechanism used to locate specific data within the table. This can help in the normalisation process which ensures better linkage with relational database.

Also, in his work, he tackles the problem of extracting of metadata which relate to the table from the accompanying text. He attempts to extract some of the metadata from the textual components of a document (e.g. table captions and column descriptions). He dealt with a very limited range of metadata, as he said, metadata extraction and semantic detection were beyond the scope of his research. He recognised its importance and did a limited investigation as a proof of concept which clearly showed the need for a fuller treatment.

Certainly, the full reuse of tabular data in textual documents will not be achieved until all related metadata for the tabular data in the documents have been detected, extracted and represented, and this is mainly our main goal.

2.2.2 Table semantics and ontology

Embley and others [ECJL98, EX01, EJN99] discuss an approach to extracting and structuring data from documents posted on the Web. Their data extraction method is based on conceptual modelling, and, this approach that also represents a direction for research in conceptual modelling itself. Their approach specifically focuses on unstructured documents that are data-rich, narrow in ontological breadth, and contain multiple records of information for the ontology.

Their data extraction method consists of the following five steps.

1. Develop an ontological model over an area of interest.

2. Parse this ontology to generate a database schema and rules for matching constants and keywords.

3. Obtain data from the Web, by invoking a record extractor that divides an un-

structured Web document into individual record-sized chunks, cleans them by removing markup tags, and presents them for further processing as individual unstructured record documents.

4. Invoke recognisers that use the matching rules generated by the parser to extract from the cleaned individual unstructured documents the objects expected to populate the model.

5. Populate the generated database schema by using heuristics to determine which constants populate which records in the schema. These heuristics correlate extracted keywords with extracted constants, and use relationship sets and cardinality constraints in the ontology to determine how to construct records and insert them into the database schema.

Yoshida and Torisawa [YTT01] describe a method to extract ontologies from tables on the Web. A table can be viewed as a device to describe related objects by attribute-value pairs. The attributes specify the information that is needed to identify and utilise the described objects. For example, they may identify a CD by its values for the attributes 'Title', 'Composer' and 'Price', and then use this information in further analysis.This use of attributes is the same as the representation of concepts in generic ontologies. More precisely, ontologies, or some part of them, can be described by attribute-value pairs, and these attributes express what is needed to be known for identification and utilisation of the described class of objects. So, by properly processing a wide range of tables, they construct an ontology for the tables domain. They propose a method that classifies a table according to the objects described it, and collects the attributes and their possible values from tables describing a class of objects.

2.2.2.1 Using domain ontology for identifying and extracting semi-structured data

In the MIX *Metadata based Integration model for data X-change* project, [BB99] [Bor99], a domain-specific ontology is used as a common interpretation base for integrating semi-structured data from the web. This work concentrates on a specific part of the data in the web page and is related to the travel industry. A semantic object is used, which may be understood to be a data item with additional information attached to support its correct interpretation. It consists of the data item (extracted from the web) together with its underlying semantics, which can be driven from either the web data itself or from the domain ontology.

This work distinguishes between simple and complex semantic objects. The concept of a simple semantic object represents atomic values, like simple number values or character strings, while complex semantic object can be understood to be a heterogeneous collection of simple semantic objects, each of which describes exactly one attribute. These sub-objects are grouped under one corresponding ontology concept.

This approach suffers a number of limitations. It is clear from the published work that there is no determination of how accurate the data detected and extracted is. Also, the isolation of a specific part of the data from the remaining document might lead to a loss of related data.

2.2.2.2 Tables and semantics of cells

Matthew Hurst's research [Hur00] is concerned with advancing a model of tables suitable for the information extraction task. The model he presents contains graphical, physical, structural, functional and semantic components, as follows:

1. Graphical: His work assumes some basic graphical representation of the table, e.g. a bitmap of a document image.

2. Physical: A description of the table is available in terms of the physical relationships between its basic elements when rendered on the page.

3. Functional: The purpose of areas of the table with respect to the use of the table by the reader is available.

4. Structural: The organisation of cells in the table is an indication of the relationships between them, representing the intent of the author within the restriction of the two-dimensional page.

5. Semantic: Description of meaning meta text in the cell, object text in the cell, the relationship between the interpretations of cell contents, the meaning of structure in the table, and the meaning of a reading of the table. An ontology is used to describe particular aspects of the table. These descriptions can be combined to deliver the desired semantic description.

Thus, if we concentrate only on the table and isolate it from the remainder of the document, then this is a limitation of the work. As mentioned at the beginning of this thesis, working with table semantics or relations will always be incomplete until this information is joined with the rest of the document.

The previous research in this area started using tables to formulate and answer queries and then progressed to reuse all the content of the table. The researchers found that it is important to find a description for the table data in order to be able to extract the data efficiently. Thus, some researchers introduced another approach for extracting data from tables by using an ontology as an external resource to facilitate that extraction.

Unfortunately, they have missed an important part of the document that is related to the table and gives us some of the description we are looking for. Instead of using only an ontology to discover the semantics of the table, we believe that there is a significant amount of semantics related to that table in the surrounding text of the same document. In our research, we concentrate on detecting, extracting and representing the semantics that are related to the table data and are buried in the text surrounding the table in the same document.

Searching documents for specific information takes us into the research field on documents and especially into the part related to information retrieval.

2.3 Document-related Research

Relevant document research occurs in two areas. The first is that of *document analysis systems* which deal with a document as an image needing to be analysed so that its content can be identified. Work in this area includes optical character recognition (OCR) [DR02b][DR02a] [SQ02], layout analysis [CCMM98] [Bre02] [AM02], handwriting recognition, indexing and retrieval [BR99] [SWS$^+$00] [Doe98, MC00] [Doe98] [DSK$^+$96], and document engineering [vO02]. These systems transform documents in an image format into a machine readable format. This is an important area as it makes use of the vast amount of documents that are presented in an image format and allow them to be used in Information Retrieval systems. The second relevant research area is that related to Information Extraction. This field concentrates on finding useful information in a document. Because our research concentrates on documents in an ASCII representation format, it is more related to document mining than to OCR systems or document engineering. Part of our research is how to search the document for the semantics related to a table appearing in the document text. There are a number of research techniques for information extraction from documents. A widely known method to extract information from Web documents is by generating a wrapper (see Section 2.3.1). *Document mining* (DM), also called *text mining*, is another approach used in identifying the content of documents and extracting targeted data (see Section 2.3.2).

2.3.1 Using wrappers to extract data from documents

One of the most common ways to extract information from Web documents is by generating a wrapper, which parses unstructured data and then maps it into a structured or semi-structured form. If the mapped form is structured, then standard query languages such as SQL are used to query the extracted information. While if the mapped form is semi-structured, then special semi-structured query languages are used [ACC$^+$97], [AQM$^+$97], [BDHS96]. Wrappers can be written by hand as they were in the TSIMMIS project [CGMH$^+$94](whose main thrust was information integration).

Wrappers can also be written semi-automatically. Approaches to semi-automatic wrapper generation include generators using (1) handcoded specialised grammars [ACC$^+$97], (2) formatting information [AK97],[Fla98], (3) page grammars [AMM97], and (4) concept definition frames [LS93]; these approaches are all similar. Wrappers have been written either fully manually [AM97],[Fla98], [FBY92], or with some degree of automation [Ade98], [AK97], [DEW96].

Hand generation and semi-automatic generation have two disadvantages: (1) the amount of work undertaken to create the initial wrapper is large, and (2) the amount of work required to update the wrapper when a source document changes is also large.

Another disadvantage is the limited semantic recognition in their work. Concentrating only on the structure of the document in identifying its content and ignoring the semantic relations between different parts of it leads to semantically poor wrappers.

2.3.2 Document mining

Significant experiments in document mining were performed at the University of Helsinki [AHKI97]. These involved the researchers applying data mining techniques to text-based resources, which become increasingly unstructured as the experimentation progressed. Knowledge-discovery-in-databases (KDD) techniques are used with some success in this area. Another report by the same group deals with investigations into the application of the raw techniques used in data mining to the results of preprocessed text information [AHKV97]. The report states that the pre-processing phase is a crucial one, as it effectively changes the nature of the data mining, which depends on how the text has been initially processed.

The knowledge discovery path is followed in a paper by Feldman and Dagan, *Knowledge Discovery in Textual Databases* (KDT) [FD98]. KDT is another term used for document mining; here, Feldman looks at using a simplistic form of information extraction to achieve knowledge discovery. Establishing a set of meaningful concepts for a text allows one to look at a hierarchical ordering of the concepts, and to "mine" for relationships between documents and between concepts. The main application of this work by Feldman and Dagan is in text categorisation. The system developed with [FKBY$^+$97], is called Document Explorer. It is one of the most advanced document mining systems currently available. Building on the KDT experience, Document Explorer constructs a database from a collection of documents, and applies data mining techniques to this database based on concept graphs. The system is generic enough to allow different modules to create databases from various types of text collections, including the World Wide Web.

A recent project at the University of Sheffield, called A General Architecture for Text Engineering (GATE), has produced several papers discussing approaches to the task of information extraction and result visualisation. An overview of this system, and its areas of applicability are found in [CGW96] which has an evaluation of the system. The GATE system was entered in the *MUC6* [1] competition. DM is closely related to IE and IR and, indeed, can be considered to be built from components that perform these tasks. An excellent view of a DM system is that it follows a sequence of steps, outlined below, which is similar to the DM process described in [FPSS96]. A similar process in extracting knowledge, although it combines the retrieval and extraction phase as a single pre-processing phase, is described in [AHKV97]. These steps are:

1. Information Retrieval: Locate and retrieve the documents considered relevant to the task at hand. Typically, users of a system can specify their own document set, but the system still needs to filter out irrelevant documents in this set so this stage must still happen.

2. Information Extraction: Extract information from the selected documents. This extraction is typically a process of filling out user-specified templates or keyword lists of expected information.

3. Information Mining: Once a template entry has been filled out for each document, one has a database which is compatible with standard DM techniques and with which pattern-discovery tasks can be be performed using normal DM tools.

4. Interpretation: Place an interpretation on the patterns retrieved from the mining phase. Ideally, the interpretation is given in natural language.

Our system is related to Document mining in respect to information retrieval and extraction. Part of our research is to locate a type of data in the text document. We have a list of words that have been extracted from the table, and we try to search the document for words that are related to our word list. There are a number of

[1] The Sixth Message Understanding Conference (MUC-6 1995) , one of a series of ARPA-sponsored conferences that has promoted research in free text IE.

techniques used in document mining for keyword searching, which should help us in searching the document text for these related words or phrases. Some of these are presented in the following sections.

2.3.2.1 Stemming

Stemming is a common form of language processing which is used in most information retrieval (IR) systems [Kro93]. It is similar to the morphological processing used in natural-language processing, but has somewhat different aims. In an IR system, stemming is used to reduce variant word forms to common roots, and thereby improve the ability of the system to match query and document vocabulary. The variety in word forms comes from both inflectional and derivational morphology, and stemmers are usually designed to handle both, although in some systems, stemming consists solely of handling simple plurals. Stemmers have also been used to group or conflate words that are synonyms (such as 'dog' and 'canines'), rather than variant word forms, but this is not a typical function. Although stemming has been studied mainly for English, there is evidence that it is useful for a number of languages, such as Slovene [PW92] and Dutch [KP96], but there are no stemming studies for the Arabic language which have a different word structure and the language syntax. Stemming is usually viewed as a recall-enhancing device in IR [KP96], since it expands the original query with related word forms.

Stemming in English is usually done during document indexing by removing word endings or suffixes, using tables of common endings and heuristics about when it is appropriate to remove them. One of the best-known stemmers used in experimental IR systems is the Porter stemmer [Por80], which iteratively removes endings from a word until termination conditions are met. The Porter stemmer has a number of problems that are found, to varying degrees, in other stemmers:

- It is difficult to understand and modify.

- It makes errors by sometimes being too aggressive in conflation (e.g., 'policy'/'police' and 'execute'/'executive' are conflated) and by missing others (e.g., 'European'/'Europe' and 'matrices'/'matrix' are not conflated).

- It produces stems that are not words and are often difficult for an end-user to interpret (e.g., "iteration" produces "iter"; "general" produces "gener").

Despite these problems, recall/precision evaluations of the Porter stemmer have shown that it performs at least as well as other stemmers (Lovins, inflectional, derivational, and removing s) [Hul96]. Krovetz [Kro93] developed a new approach to stemming, based on machine-readable dictionaries and well-defined rules for inflectional and derivational morphology. This stemmer (now called KSTEM) addresses many of the problems with the Porter stemmer, but does not produce consistently better recall/precision performance when this is evaluated. One of the reasons for this, is that KSTEM is heavily dependent on the entries in the dictionary being used and can be conservative in conflation. For example, because the words 'stocks' and 'bonds' are valid entries in a dictionary for general English, they are not conflated with 'stock' and 'bond' (which are separate entries). If the database being searched is *The Wall Street Journal*, this could be a real problem.

Evaluations of stemming techniques using test collections have produced mixed results [Har91], but more recent work has shown consistent (if rather small) improvements in retrieval effectiveness across a range of collections [Hul96, Kro93]. Giving that some stemming algorithms have 260 suffix patterns, it could be that stemming might mislead the searching process by adding a lot of unwanted words. As a result of that evaluation, we are not going to use stemming in our research to narrow the bandwidth of searching words. Stemming might be used in the future by researchers taking forward this work to measure the enhancement that stemming might give to this area of research.

2.3.2.2 Simple word searching

This method uses the exact words provided in the search process. This method relies on the fact that most users know what they are looking for, and are looking for a precise word. This method is simple, straightforward, and time and effort saving. Simple word searching is mostly beneficial when searching for words that have been collected automatically. In a typical system, the software produces a list of words that have been extracted from the user profile and searches for them in a number of documents. It is highly likely that the words such software is looking for are in the list. In our research, the word lists that are going to be used in the search has been created using words from the document's table, and the search will be in the same document, there may be a similarity between the words used in the table and the text. This is not surprising as the writer of the document is likely to use the same words if he/she is going to refer to a concept again.

2.3.2.3 Synonyms

Synonyms are words that have the same meaning, e.g. car, automobile and vehicle. A language is called a rich language if it has many synonyms in it. Knowing all possible synonyms gives us the ability to understand more fully the language that we are using. In text mining, synonyms play a great role in detecting the targeted words by broadening the bandwidth of the searching scope. Synonyms usually do not add a huge number of words to the search list and therefore do not cost much in time and effort. There is a number of good resources of synonyms on the Internet, and this technique is widely used by text mining in the Web [www.thesaurus.com].

One of the disadvantages of using synonyms in text mining is the difficulty of managing synonyms, whose meaning changes with the context (eg. 'marriage' between two companies is seen as a partnership and not as a sacrament). Therefore, using a domain ontology as a source of synonyms instead of a general synonyms database will address this problem as it gives context. In our research, we use a domain ontology to augment the table header words and enrich the searching list. We extract the concept which the searching word is related to in the domain ontology and all synonyms associated with that concept, together with the concept relations. This gives us an extended word list augmented by synonyms related to the concepts being used in the table, which is appropriate to our research. We investigate how useful such an ontology based approach is.

2.3.2.4 Linguistic analysis

More recently introduced technology uses linguistic analysis. It is based upon the structure of language, and improves the process of text searching greatly by more accurately recognising the context within which words are used. This technology uses a number of natural language processing components such as : automatic language and character encoding identification, document analysis which identifies paragraphs and sentences within text, word segmentation, stemming, and part-of-speech tagging [www.inxight.com] These types of software are costly to buy, difficult to build, and require high performance computing systems if they are to give good results. They are mainly used by search engines and while they may be useful in our domain of research, they were not investigated in this project.

2.4 Summary

This chapter reviewed the main areas of research concerned with locating text in documents and tables referring to particular concepts. Two approaches for document and table-related research ware identified as relevant to our work: the physical-layout approach which contains research areas like OCR, layout analysis, and table recognition; and the logical-content approach which comprises areas like document mining, document summarisation, table and information retrieval, and table semantics and ontology. Further discussion on table semantics is presented in Chapters 3 and 4, where ontologies and knowledge bases are described more fully.

CHAPTER 3

Semantics and Representation Detection

3.1 Introduction

A common problem in information retrieval and information extraction is that the terms employed by a user to refer to some concept may not be equivalent to the terms employed in a document to refer to the same concept. In a commercial context the user may be interested in 'Cars', but a document might refer only to 'Vehicles', such as 'Toyota', or to the associated model names such as 'Land Cruiser'. To overcome this problem, the semantics of a word must be combined in the IRS to ensure that all equivalent and related terminologies are detected. A method is required therefore to translate between equivalent terminologies to find related information. This can be addressed by the use of ontologies that encode semantic relationships between concepts, and hence facilitate the detection of associations between related terms. With the development of new NLP techniques in recent years [TSN00], considerable attention has been paid to exploring the potential of NLP in different areas of information retrieval. Currently there is considerable interest in the development of natural language text processing systems that develop semantic analysis that may be used in accessing information from text by locating text related to a concept [Ner96, RB98] . Semantic analysis in NLP deals with the meaning of the words and sentence and this is usually stored in a knowledge base format, i.e. ontology. The ontological information is used to derive meaning and to resolve ambiguities that cannot be resolved by considering only structural considerations [DN92].

In the case of tables presented in textual documents, the table metadata is not enough to resolve the semantic and representational conflicts that might occur when using this data. However, information that is related to the table metadata can often be found in the text surrounding the table. An ontology can be used in detecting semantics that are related to the table metadata and searching for such information in the combined text. By detecting these related semantics in the surrounding text, the table metadata can be enhanced by related semantics, and semantic conflicts can be detected and resolved using the enhanced metadata the resolution process.

In this chapter, we are discussing semantics, the meaning of concepts and the different types of semantics that can occur with their different representation. We concentrate on table semantics and the differences between semantics and representation. Considering semantics leads us to semantic heterogeneity and how to solve problems due to the heterogeneity by using the context of the table in the document. The chapter also covers semantic units and enhanced metadata and their potential use.

3.2 Semantics

Semantics is the study of meaning in language, with language taken in a quite general sense, as it includes natural languages, programming languages, graphical languages, technical drawing notations, etc. In the case of programming languages, it is important to be able to specify precisely their semantics. The two major methods for doing so, being *operational semantics* and *denotational semantics.*

The operational semantics for a programming language is specified in terms of an underlying model of computation. For example, the database query language SQL can be described using an operational semantics based on the relational database model which is based on relational algebra.

The denotational semantics for a programming language is specified in terms of mathematical functions. For example, relational algebra can provide denotational semantics describing the queries and tables of a relational database [DC02].

Natural semantics refers to the meaning of a concept in terms of the real world. For example, the natural semantics of a relational database is the relationship between its tables and fields and the real world entities that they represent. A *semantic network* is a mathematical object which can be used to specify semantics in terms of a network of concepts. The meaning of a concept in a semantic network is defined by "everything the concept is connected to" in this network [Qui68]. Consequently, the meaning of one concept is constrained by the other concepts to which it is connected. Thus, if concept A has a parent B to which it is connected by an 'IS-A link', then the meaning of A is constrained to be more specific than the meaning of B. Similarly, if concept A has a 'part-of' relationship to a concept C, then A is constrained to be a part of C, and often implies that A is physically connected to other parts of C. In other words, one does not give a definitive expression for the full semantics of each concept, but rather one describes the relationships among the concepts [GGPH03].

In the case of a document containing tabular data, there are a number of ways, of representing the semantics of that data and its values, the real-world meaning of the table's metadata, for example the real-world concepts related to the metadata, and all information related to that table in the document.

3.3 Table semantics

The table semantics can be described as all the information related to that table. This related information can be found in metadata of that table, relations to that table in the DBMS, the user knowledge about this table which most of the time are assumed to be obvious and common by understood and derivable from the context that the table appears in. Of course, not all information related to the table is beneficial information in that it is useful to a user of the table. However, we can say that information which describes or is related to the table metadata is beneficial data. This information can be categorised as:

1. Relationship between the table and other tables in a database or in the real world.

2. The structure of the table and the meaning of each element in the table, in other words, the provided metadata.

3. In the case of tabular data presented in a text document, its semantics also means all the information in that document which is describing the metadata of that table. This can be described as *enhanced metadata.*

3.3.1 Metadata

Metadata is information about the data. It can be used to develop a logical model of entities and the associations between those entities [HPM02] We distinguish here between structural and semantic metadata. Structural metadata represent information that describes the organisation and structure of the recorded data, e.g. information about the format, the data types used, and the syntactic relationships between them. In contrast, semantic metadata provide information about the meaning of the available data, i.e. data that describes the semantic content of the data values (unit of measurement and scaling), the semantic relationship between elements of the data (i.e. age can be calculated from date of birth and today's date), and data that provide additional information about its creation (calculation algorithm or derivation formula used) and quality (e.g. actuality and precision) [Mad95]. Some of these semantic metadata which are not presented in the table itself need to be described and represented so that users of the table can overcome any semantic conflicts by its use. We call semantics represented in this way enhanced metadata.

3.3.2 Enhanced Metadata

As discussed in [AG02][AG03a], when we are using tables of data in a text there may be information in the text which is related to the table metadata. This is usually found in the text around the table. This hidden information describes the meaning of the table as metadata and it also adds more semantics related to that metadata. This information is usually not represented in the structure of the table. However, these hidden semantics can be used to enhance table metadata with data that either describes the metadata itself or adds more meaning to it. The enhancement would come from enriching the semantics of the metadata and from the declaration of the table metadata. In Figure 3.1 for instance, the table metadata can be enhanced with information from the text combined with the table. We can add 'engine-size, litres' and 'VAT,excluded' which definitely will enhance the understanding of the table data. Detecting, extracting and representing these data is a major part of our research and we call these elements of a table's semantics its *enhanced metadata*, once it is represented in this format.

3.4 Semantics and Representational Conflicts

The representations of semantic content in a text are idiosyncratic, in the sense that representational similarities and differences cannot always be easily recognised. An example of this in two tables might be the number '25' appears as a temperature but in one it is degrees centigrade and in the other fahrenheit. This is called *heterogeneity*. Heterogeneity can be found between any data that is drawn from different data sources, or sometimes from the same source when it has been created by different programmers. One study found that the probability, that two database designers, even when they are domain experts, will choose the same element names for the same data attribute, is between .07 and .18 [FLGD87]. The author suspects that the probability is even lower for data on the Web. In general, heterogeneity is very likely to occur when data is used from different sources.

Furthermore, the meanings of names and values may change over time or when used in different places. For example, the representation of prices in France has changed from Francs to Euro, and the Soviet Union has changed its name to Russia with parts of it becoming separate countries. Also, the representation of weight 'pounds' is different between UK and USA and each country has its own value of gallon (the US gallon being smaller than the UK gallon). When trying to integrate data from different sources drawn from the same domain, *semantic conflicts* and *representational conflicts* will occur frequently. A semantic conflict is a subset of semantic heterogeneity - it is concerned with differences in the meaning of a table's metadata, i.e. attributes and names, resulting from such integration. Examples of conflicts are:

1. synonym, when different data names represent the same data item in the real world.

2. homonym, when the same name represents different data in different domains, e.g. plant can be a biological entity or a factory.

3. hidden semantic relationship which exists between two or more domain terminologies.

For example, the relationships between cost and price, profit and net-profit, or car, vehicle and lorry cannot be understood unless we use a domain ontology to relate these terms, and so overcome the synonym problem.

In Web documents, semantic heterogeneity is even worse than between databases, as each provider of data has complete autonomy. Thus, it is believed that documents created for the Web exhibit more heterogeneity among their data than other types of documents. This is however a belief, which is probably well founded since:

1. The amount of structured data in Web pages is generally less than in databases. This is because the Web pages are designed to show data to any user in such a way that a user can understand the Web page on its own as an isolated unit of information. while for structured data, for instance data managed by an RDBMS, the user needs to be a knowledgeable person to be able to work with it and understand the structure that it uses, and Internet users do not always have this type of knowledge. Therefore, the designers of Web pages try not to use complicated data structures in presenting their data on the Web pages. Thus, semantic heterogeneity will appear in unstructured or semi-structured data sources more than in structured data.

2. The amount of data in Web pages is small compared to the amount held in DB, therefore when trying to integrate data from web sites for a certain purpose, i.e. to answer a user query or for IRS, we need a larger number of Web pages to get the same amount of data. Therefore the chances of representation heterogeneity are higher because each data source can have its own representations.

3. Web pages are designed for different purposes for different people of different cultures. We mean by different purposes, each Web site has a purpose, for instance commercial, educational, or news. Each purpose has an associated way of presenting data in the Web page. Also, Web sites are designed for different users, that is, each Web site has its own targeted users and it concentrates on providing data suited to the users, i.e. scientist, students, buyers, or general

users. Thus, the culture of the designer and the user affect the content of the Web page in all sorts of ways, e.g. preferred colours, traditional clothes, and religion. All these differences in the design of the Web page lead to different representations, which lead to high heterogeneity between pages.

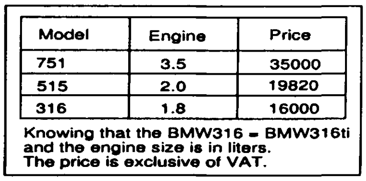
4. The Internet is free and open. Thus, Web designers are free to present data as they wish. Documents on the Web can be structured, semi-structured, or unstructured, while databases are very structured and usually designed with specified aims and under the control of staff like a DBA (database administrator).

5. In relational databases, data is found in a table, whereas in a Web page data can be in a table, in plain text, or in graphics. This leads to difficulties in extracting and identifying the content of the Web page which might lead to missing some of the data representation in the Web pages and so missing the data.

Representational heterogeneity, also called syntactic heterogeneity, refers to differences in the representation, i.e. the structure of semantically equivalent information. For example, SALARY information might be stored in pounds per hour or pounds per year; AGE information might be stored directly or be computed dynamically using DATE-OF-BIRTH and the correct date. See also [Hei95], for additional examples of heterogeneity.

3.4.1 Representational conflicts

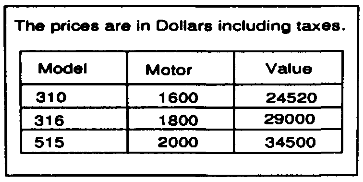
Representational conflicts occur due to the way data is represented and often the measurement unit used. Such conflicts are concerned with the value of an attribute.



Model	Engine	Price
751	3.5	35000
515	2.0	19820
316	1.8	16000

Knowing that the BMW316 = BMW316ti
and the engine size is in liters.
The price is exclusive of VAT.

Figure 3.1: An Example of Car Prices



The prices are in Dollars including taxes.

Model	Motor	Value
310	1600	24520
316	1800	29000
515	2000	34500

Figure 3.2: An Example of Car Evaluation

If we have two attributes that are semantically identical and presented with the same representation, it does not mean that there will not be any representational conflicts in their values. For example, if we have two fields which represent prices and they are represented in Pounds, there will be still a chance of representational conflicts. For instance does the value have VAT included or excluded. Also, rounding the values might lead to comparison problems. This is important when attributes are brought together; if the units are different, they must be converted to the same representation before comparisons are made. There are a number of reasons that can cause a representational conflict, for instance the documents that have those data have been created in different countries and each country has its own standards and variables. Another reason is that some representational values have the same names but different meaning, for example dollar might be US dollar or Australian dollar. Fortunately, a solution is not difficult as *representational metadata* can be used to provide information about the meaning of the value of an attribute, its representational relationships, and its units of representation.

3.4.2 Examples of Conflict Data

Figure3.1 and 3.2 show information about cars held in tabular form. These data sources describe equivalent information differently. They provide information about different aspects of cars, and represent the same real-world aspects using different structural constructs and semantic concepts. For example, in Figure 3.1 price is in pounds and exclusive of VAT and in Figure 3.2 value, which is the same conceptually as price, is in U.S. dollars and includes a sales tax. If the data in figure 3.1 and 3.2 is merged using only the table headings in integrating, the tables will result in semantic and representational conflicts. Thus, using the contextual information of such tables will facilitate the integration of these tables and solve some of the semantic and representation conflicts. The contextual information about such tables is usually presented in the accompanying text, as only a limited amount of metadata can be held in a table.

3.5 Addressing Semantic Heterogeneity

There are several ways to solve the problem of semantic heterogeneity in documents. Ontologies make explicit human intuitions about the meanings of domain names. They standardise the semantics of the vocabulary of the domain, so an entity type, or attribute name have agreed-on well defined meanings. The use of an ontology in a domain can give the same benefits of precision and mutual understanding enjoyed by mathematicians who can refer to a square root without the necessity for providing extensive context because the term has a well defined and understood meaning. If the standard is comprehensive, and has been adhered to, problems of semantic differences are greatly reduced. Even if the standard has not been adhered to, the documentation of semantic information can be limited to describing deviations from the standard. This simplifies the problem.

Therefore, using an ontology as a common interpreter between documents for a specific domain and its real-world terminologies helps to identify the hidden semantics in these documents. In the case of tables presented in documents, semantic heterogeneity is most likely to occur between different tables from different documents because there is not enough description of the semantics of the table. There are some hidden descriptions about the table metadata in the text surrounding the table; these hidden semantics can be used to overcome and resolve some of the heterogeneity between data held in separate tables. Thus using an ontology to detect related semantics in the text can be combined with the table results as an enrichment of table semantics and will help in eliminating the semantic heterogeneity between these tables.

3.6 Semantic Conflict Detection

The information needed to detect semantic conflicts when combining tables from different documents is often buried deep within the text associated with the table or in the Web site itself [SSR94]. For example, in Figures 3.1 and 3.2, the conflict between the fields names price and value will not be detected until the text metadata is combined with the table metadata which declare that in Figure 3.1 the price is exclusive of VAT, and in Figure 3.2 tax is included. In order to resolve problems caused by semantic conflict, an information system must be able to ensure valid semantic *interoperability* is occurring. This requires the discovery and utilisation of contextual information in the text which allows mapping the data to common representations. The context of a datum in a table is the information relating to its meaning [GMS96] [GBMS99].

Context information can be quite varied in form, e.g. it includes information like a unit's specification, such as currency or length. For example, in Figure 3.1 we can see that the context of the table gives us information, that engine size is in litres not in CC and price is exclusive from VAT. To extract useful context information, we need to know what kind of context information is needed to resolve or detect the conflicts for a specific domain and use it effectively. Using a domain ontology together with the tabular data in the document to illustrate the concepts and their attributes will resolve some of the heterogeneity in the semantics of that table. For instance, from the cars domain ontology, the price has a synonym value and it always either includes or excludes VAT or TAX. Thus, using this information from the domain ontology and the surrounding text, we can find that price in figure 3.1 is exclusive of VAT, but in Figure 3.2 it is included.

A number of approaches, such as text searching and augmenting the header and title using an ontology, can be used to help detect contextual information about a table presented in a document. Details of detection approaches are discussed in Chapter 6.

3.6.1 Semantic Units

A *semantic unit* comprises a datum together with its associated semantic context, consisting of a flexible set of meta-attributes describing the meaning of the datum. However, because we cannot describe all modelling assumptions, the semantic context always has to be recognised as being a partial representation. In addition, each semantic unit has a concept label associated with it that specifies the relationship between the unit and the real-world aspects it describes. These labels must be taken from a well known vocabulary or ontology for the domain. In this way, the concept labels,as well as the semantic context of a semantic unit help to describe the meaning of the data (see further discussion of this in Section 6.6.1).

3.6.2 Enhanced metadata representation

Table metadata is the information held in the table itself, e.g. the header describes the full table with its corresponding contexts. The table headings in the original table are recognised as part of this table metadata. This metadata is enhanced with any semantic and representation context found in the text. This table-enhanced metadata is suitable for storing as data in a format which allows it to be used when the table is linked with data from another database either for interoperation or integration. We represent the detected semantics for a table as follows:-

$$EnhancedMetadata =< C, SA >$$

Where C represents the knowledge concept derived from the domain ontology which relates to a corresponding data element in the table, and SA represents the semantic contexts that have been discovered in the text about this concept.

3.7 Summary

In this chapter we have discussed semantics, the meaning of semantics and the different types of semantics. We have concentrated on table semantics, the differences between semantics and representation, and the conflicts that might occur between data with different representations. Also we have described about semantic heterogeneity and how to solve this problem by using the context of the table in the document. We finished by describing the different methods of representing data along with its context and defining semantic units and their use to represent enhanced metadata.

CHAPTER 4

Knowledge Base

4.1 Introduction

From the discussion in Chapter 3 about the role of semantics in identifying and detecting the possible heterogeneity between tables drawn from different documents using the surrounding text, we identified that enriching the retrieval system with the semantics of a table's metadata will facilitate the detection of the description of the metadata in the surrounding text. We have decided that using an ontology as a common interpreter between documents for a specific domain and real-world terminologies helps in identifying the hidden semantics in these documents. Thus using an ontology to detect related semantics in the text which can be combined with the table metadata, results in enrichment of table semantics and will help in eliminating the semantic heterogeneity between tables.

In this chapter, we are discuss ontologies as there are different definitions that have been used to declare the meaning of ontology. Each definition looks at ontology from a different angle, concentrating on a specific aspect of an ontology that is related to the area of research. Different types of ontologies can be found and these are categorised depending on the level of generality and subject of conceptualisation. Ontology representation is an aspect of ontology that affects the usability of the ontology. Also, we are going to look at the different schemas that knowledge representation uses since we will have to link these representations to identify the different features. Finally, a number of representation languages will be described.

4.2 Ontology

In recent years, the use of an ontology has become increasingly widespread in the computer science community. While this term was mainly confined to the philosophical sphere in the past, it is now gaining a specific role in Computer Science, particularly in Artificial Intelligence [GN87], Computational Linguistics [Lan91], and Database Theory [Rei84]. In particular, its importance is being recognised in research fields as diverse as knowledge engineering [Gai97, Gru93, UG96], knowledge representation [Gua97b, Gua95, Sow99], qualitative modelling [BGM96, CV97, GG96], language engineering [Bat95, Lan91], database design [Bur97, RBD98], information modelling , information integration [MKSI98, Wie96], object-oriented analysis [Paz98, Wan89], information retrieval and extraction [Gua97a, SFDB99, McG98, Wel98], knowledge management and organization [Pol96], and agent-based systems design. The current application areas are disparate, as they include enterprise integration [GL02, SCH$^+$02], natural language translation [Mah96], medicine [GPS98], mechanical engineering [SM00], standardisation of product knowledge [BCWW97, GBM97], electronic commerce [Leh96], geographic information systems [CSV98], legal information systems, and biological information systems.

In philosophy, the term 'ontology' refers to "a particular theory about the nature of being or the kinds of existence." [HSW97]. This broad definition can be interpreted in a number of ways, depending on the metaphysical stance that one takes with respect to what 'existence' is. A number of researchers in knowledge engineering have therefore suggested more specific, AI-oriented definitions of ontology. In general, AI definitions avoid referring to reality, but rather use such terms such as representation and conceptualisation to describe the role of an ontology. An often cited definition is that of Gruber [Gru93]:

An ontology is an explicit specification of a conceptualization.

Thus, the term is borrowed from philosophy, where an ontology is a systematic account of Existence. For AI systems, what 'exists' is that which can be represented [Gua97b]. Although not explicitly stated in the wording, this definition suggests, by mentioning the conceptualisation, that an ontology is a meta-level description of a knowledge representation. Thus, the ontology is not part of the representation itself. This means, ontology is a description of concepts without being concerned about the real values of these concepts. Another important aspect of an ontology that can be found in a definition formulated by Wielinga and Schreiber as:

An (AI) ontology is a theory of what entities can exist in the mind of a knowledgeable agent [WS93].

This definition emphasises that we want to apply the notion of ontology to the concepts in the knowledge base of all knowledgeable agents, including humans. Since different knowledgeable agents will often have different symbol-level representations of their stored knowledge, it is convenient to formulate ontologies at the knowledge level. This aspect is important for knowledge-engineering.

A third definition of ontology which is knowledge engineering oriented is given by L. Alberts [Alb93]:

An ontology for a body of knowledge concerning a particular task or domain describes a taxonomy of concepts for that task or domain that define the semantic interpretation of the knowledge

The three definitions above are not contradictory, and capture a large proportion of the aspects of ontology that are relevant for the research work of this thesis. Combining the above definitions results in the following definition due to Genesereth and Nilsson [GN87]:

An ontology is an explicit knowledge-level specification of a conceptualization, i.e. the set of distinctions that are meaningful to an agent.

Conceptualisation is the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualisation is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualisation, explicitly or implicitly, and an ontology is an organised knowledge base holding these concepts.

4.2.1 Types of ontologies

There are three categories of ontology which summarise all the types of ontology used in the diverse research areas namely: top-level ontologies, domain ontologies and task ontologies, and application ontologies. This categorisation depends on the level of generality and subject of conceptualistion of the ontology. It is useful to our work, as we are concentrating on a subjective and general ontology.

4.2.1.1 Top-level ontologies

These type of ontology describes very general concepts like space, time, matter, object, event, action. (e.g. ONTODM [GdF03] and TOVE [FFG94]). They are independent of a particular problem or domain. They apply to large areas of knowledge and contain general concepts. Thus, it is reasonable, at least in theory, to have unified top-level ontologies for large communities of users, which cover diverse domains.

4.2.1.2 Domain ontologies and task ontologies

These type hold descriptions of the vocabulary related to a generic domain (like medicine e.g. GALEN and SNOMED-CT [BMM03], or automobiles) or a generic task or activity (like diagnosing a patient's illness or selling). These ontologies specialise the terms occurring in the top-level ontology. Current knowledge engineering methodologies make an explicit distinction between domain ontologies and domain knowledge. Whereas the domain knowledge describes factual situations in a certain domain, the domain ontology puts constraints on the structure and contents of this domain knowledge.

4.2.1.3 Application ontologies

This ontology contains all the definitions that are needed to model the knowledge required for a particular application. Typically, application ontologies are a mix of concepts that are taken from domain ontologies and from generic ontologies. Moreover, applications ontologies may contain method and task-specific extensions. Application ontologies are not reusable in other applications. They may be obtained by selecting theories from the ontology library, which are then fine tuned for the particular application (e.g. PROTEGE-II [TEG$^+$95]).

From the above definitions and elaborations, it is clear that an ontology can be used to assist in the interpretation of data. Thus, to detect the semantics which relate to certain data, an ontology can be involved. In our research, we use two types of ontologies, namely a domain ontology and a top-level ontology. These two types of ontology are nearly always needed in most systems that need an ontology, since within most domains there are some concepts that are common to many domains and others that are specific to the domain. In other words, there are general concepts, which should not be represented and repeated in the domain ontology (e.g. colour, time, measurement, etc.) but will be in the a top-level ontology, while the specific concepts will be in the domain ontology.

4.3 Ontology Representation

Knowledge representation is a central problem in artificial intelligence. The question is how to store and manipulate knowledge in an information system in a formal way, so that it may be used by mechanisms to accomplish a given task. There are a number of techniques or schemas of knowledge representation as follows:

1. Logical Representation Schemas. Such schemas employ the notions of constant, variable, function, predicate, logical connective and quantifier to represent facts as logical formulas in some logic.

2. Network Representation Schemas. Such schemas, often called semantic networks, attempt to describe the knowledge in terms of objects (nodes) and binary associations (labelled edge).

3. Procedural Representation Schemas. Such schemas view a knowledge base as a collection of procedures expressed in some language.

4. Frame-based Representation Schemas. Since 1975, when Minsky originally proposed it [Min74], the notion of frame has played a key role in KR research.A frame consists of slots which contain values; for instance, the frame for house might contain a colour slot, number of floors slot. [Myl80]

4.3.1 Knowledge representation languages

There are a variety of languages which can be used for representation of conceptual models, with varying characteristics in terms of their expressiveness, ease of use and computational complexity. The field of knowledge representation (KR) has, of course, long been a focal point of research in the AI community [RD88]. Here we simply outline some of the KR languages which have been used:

4.3.1.1 Traditional ontology specification languages

In this subsection, we analyse the languages which can be considered as standards for the ontology community, namely: Ontolingua, OCML, FLogic and LOOM.

4.3.1.1.1 Ontolingua Ontolingua [FFR96] is a language based on KIF [GF92] and on the Frame Ontology approach [Gru93]. It is the ontology -building language used by the Ontolingua Server [FFR96]. KIF (Knowledge Interchange Format) was developed to solve the problem of heterogeneity in languages for knowledge representation. It provides for the definition of objects, functions and relations. KIF has declarative semantics, and is based on first-order predicate calculus, with a prefix notation. However, the Frame Ontology [Gru93], built on top of KIF, allows an ontology to be specified following the paradigm of frames .

4.3.1.1.2 OCML OCML [Mot99] stands for Operational Conceptual Modelling Language. It was originally developed at the Knowledge Media Institute (UK) in the context of the VITAL project to provide operational modelling capabilities for the VITAL workbench. The current version of the language is version 6.3. It provides a mechanisms for expressing items such as relations, functions, rules (with backward and forward chaining), classes and instances. In order to make the execution of the language more efficient, it also adds some extra logical mechanisms for efficient reasoning, such as procedural attachments.

4.3.1.1.3 FLogic FLogic [KLW95] is an acronym for Frame Logic. FLogic is a language which integrates frame-based languages and first-order predicate calculus. It accounts in a clean and declarative fashion for most of the structural aspects of object-oriented and frame-based languages. These features include object identity, complex objects, inheritance, polymorphic types, query methods, encapsulation.

4.3.1.1.4 LOOM Loom [Mac91] is a high-level programming language and environment intended for use in constructing expert systems and other intelligent application programs. LOOM achieves a tight integration between rule-based and frame-based paradigms. It supports a 'description' language for modelling objects and relationships, and an 'assertion' language for specifying constraints on concepts and relations, and to assert facts about individuals.

4.3.1.2 Web languages for building ontologies

The recognition of the key role that ontologies are likely to play in the future of the Web has led to the extension of Web markup languages in order to facilitate content description and the development of Web-based representations of ontologies, e.g., XML Schema, RDF (Resource Description Framework), and RDF Schema [LD01],[http://www.w3.org/RDF]. RDF Schema (RDFS), in particular, is recognisable as an ontology/knowledge representation language. It describes classes and properties (binary relations), range and domain constraints (on properties), and subclass and subproperty (subsumption) relations between the concepts represented.

RDFS is, however, a very primitive language , and more expressive power would clearly be necessary/desirable in order to describe resources in sufficient detail. Moreover, such descriptions should be amenable to automated reasoning if they are to be used effectively by automated processes, e.g. to determine the semantic relationship between syntactically different terms. Thus, it is a language limited in its representational power. This section provides an analysis of the new languages created in the context of the Internet (XOL, SHOE and DAML +OIL). We describe these Web languages which are used for building ontologies.

4.3.1.2.1 XOL XOL [KCT99] stands for XML-Based Ontology Exchange Language. XOL was designed to provide a format for exchanging ontology definitions among a set of interested parties. Therefore, it is not intended to be used for the development of ontologies, but as an intermediate language for transferring ontologies among different database systems, ontology-development tools or application programs. XOL allows definition in an XML syntax.

4.3.1.2.2 SHOE SHOE [HHL99] stands for Simple HTML Ontology Extension. It is being developed at the University of Maryland. SHOE was first an extension of HTML, with the aim of incorporating machine-readable semantic knowledge in HTML or other World Wide Web documents. Recently, it has been adapted in order to be XML compliant. The intent of this language is to make it possible for agents to gather meaningful information about Web pages and documents, improving search mechanisms and knowledge gathering.

4.3.1.2.3 DAML and OIL In 1999, the DARPA Agent Markup Language (DAML) program was introduced with the aim of providing the foundations of a next generation semantic Web [MFHS02b]. As a first step, it was decided that the adoption of a common ontology language would facilitate semantic interoperability across the various projects making up the program. RDFS was seen as a good starting point, and was already a proposed World Wide Web Consortium (W3C) standard, but it was not expressive enough to meet DAML's requirements.

A new language called DAML-ONT [BLvHH00] was therefore developed that extended RDF with language constructors from object-oriented and frame-based knowledge representation languages. Like RDFS, DAML-ONT suffered from a rather weak semantic specification, and it was soon realised that this could lead to disagreements, both amongst humans and machines, as to the precise meaning of terms in a DAML-ONT ontology.

At around the same time, a group of (largely European) researchers with aims similar to those of the DAML researchers (i.e. to provide a foundation for the next generation Web and its languages) had designed another Web-oriented ontology language called OIL (the Ontology Inference Layer) [FHH+00, FHH+01]. Like DAML-ONT, OIL had an RDFS-based syntax (as well as an alternative XML syntax) and a set of language constructors based on frame-based languages. The developers of OIL, however, placed a stronger emphasis on formal rigour, and the language was explicitly designed so that its semantics could be specified via a mapping to a very expressive description logic, SHIQ [Hor02]. This would allow reasoning to be undertaken.

It became obvious to both groups that their objectives could best be served by combining their efforts, the result being the merging of DAML-ONT and OIL to produce DAML+OIL. The merged language DAML +OIL has a formal (model theoretic) semantics underpinning that provides machine and human understandability (as well as an axiomatisation [FM01]), and has a set of constructions formed by a reconciliation of the language constructors from the two languages (see Table 4.1).

Until recently, the development of DAML+OIL has been undertaken by a committee largely made up of members of the two language design teams (and rather grandly titled the Joint EU/US Committee on Agent Markup Languages). More recently, DAML+OIL has been submitted to W3C as a standard and it is to form the basis for the W3C's Web ontology language which the Web-Ontology Working Group has been mandated to deliver [1]. As already mentioned, beside the set of constructors supported, the other aspect of a language that determines its expressive power is the kinds of axiom supported. Table 4.2 summarises the axioms supported by DAML+OIL.

These axioms make it possible to assert subsumption or equivalence with respect

[1] W3C web ontology working group have delivered an ontology language called "OWL" (http://www.w3.org/2004/OWL) but it come too late for us to use in our work

Constructor	DL Syntax	Example
intersectionOf	$C1 \sqcap ... \sqcap Cn$	Human $\sqcap$ Male
unionOf	$C1 \sqcup ... \sqcup Cn$	Doctor $\sqcup$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x1, xn\}$	$\{john, mary\}$
toClass	$\forall P.C$	$\forall$hasChild.Doctor
hasClass	$\exists P.C$	$\exists$hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$citizenOf.$\{UK\}$
minCardinalityQ	$\geq nP.C$	$\geq$ 2hasChild.Lawyer
maxCardinalityQ	$\leq nP.C$	$\leq$ 1hasChild.Male
cardinalityQ	$= n$ P.C	= 1 hasParent.Female

Table 4.1: DAML+OIL class constructors

Axiom	DL Syntax	Example
subClassOf	$C1 \sqsubseteq C2$	$Human \sqsubseteq Animal \sqcap Biped$
sameClassAs	$C1 \equiv C2$	$Man \equiv Human \sqcap Male$
subPropertyOf	$P1 \sqsubseteq P2$	$hasDaughter \sqsubseteq hasChild$
samePropertyAs	$P1 \equiv P2$	$cost \equiv price$
disjointWith	$C1 \sqsubseteq \neg C2$	$Male \sqsubseteq \neg Female$
sameIndividualAs	$\{x1\} \equiv \{x2\}$	$\{KSA\} \equiv \{SaudiArabia\}$
differentIndividualFrom	$\{x1\} \sqsubseteq \neg\{x2\}$	$\{john\} \sqsubseteq \neg\{peter\}$
inverseOf	$P1 \equiv P-$	$hasChild \equiv hasParent-$
transitiveProperty	$P+ \sqsubseteq P$	$ancestor+ \sqsubseteq ancestor$
uniqueProperty	$\top \sqsubseteq\leq 1P$	$\top \sqsubseteq\leq 1hasMother$
unambiguousProperty	$\top \sqsubseteq\leq 1P-$	$\top \sqsubseteq\leq 1isMotherOf-$

Table 4.2: DAML+OIL Axioms

to classes or properties, the disjointness of classes, the equivalence or nonequivalence of individuals (resources), and various properties of properties. Thus, it seemed the best choice for our work as it gave us the functionality required (see next section). We have represented our domain ontology using the DAML +OIL language, and for the generic ontology, we have used a subset of an Engineering Mathematics ontology called Standard-Unit Ontology[GO94]

4.3.1.2.4 Reasons for choosing DAML+OIL We decided to use DAML+OIL in our system for a number of reasons (see [HPH02],[CHH⁺01],[MFHS02a]

- It is specifically designed to be user in a Web context.
- It is a clearly defined language.
- It supports reasoning.
- It has support tools which make it easier to use - such as DAML Builder, DAML Search, and DAML Viewer.
- It is open resource and readily available.
- It is reasonably stable.

Of course, some of the other ontology languages that have been mentioned have some of these advantages but not all of them as in DAML+OIL, hence our decision.

4.4 Summary

In this chapter, we have discussed the different definitions of ontology and its role in representing the semantics of data. Also, we have discussed the different types of ontologies. In our framework we are using DAML+OIL to build the domain ontology therefore DAML+OIL was overviewed. A Standard Unit ontology also a part of our framework and we gave in this chapter a brief description of this ontology.

CHAPTER 5

System Design and Representation

5.1 Requirement gathering

This project needs to analyse a set of documents which contain tabular data and so find descriptions about the table in the text surrounding the table as this is the main objective of the project. This objective requires a set of sub-objectives to be met, namely, gather the table metadata and related words to this metadata so they used to search in the text to locate the table description; analyse the content of documents and prepare them to be searched; detect and extract matching words from the text; represent the extracted words and integrate these words, bearing in mind that they might need conversion to a common representation. To fulfil these requirements, a number of techniques need to be involved in the development of this system, like word collection, searching process, extraction methods and conversion functions. These are incorporated into the system to give it the required functionality.

5.2 System components

This project's system is divided into three parts namely: input of documents, document analysis, and output of the results.

5.2.1 Input documents

The input documents are a set of documents that contain tabular data surrounded by text. These documents are in ASCII format and they are extracted from the Web. Each document consists of a textual part - a set of between 7 to 10 paragraphs. Each paragraph consists of a number of sentences. The second part of a document is its tabular data. A table consists of a header, which we call table metadata, and the data itself. The table can be in any part of the document and has text adjacent to it. We assume that the table has already been identified, extracted from the document, and represented in the system's required common format.

5.2.2 Document Analysis

The ability to analyse the documents to detect the semantics that are related to the table metadata in the text is a prime requirement of the system. This objective requires a number of techniques to be available within the system. This can be described as follows.

5.2.2.1 Word collection

This is the first phase in the system. It concentrates on collecting the set of words that are going to be searched for in the text. This phase starts with the table metadata as an initial set of words that need to be searched for. There are a number of techniques that can be used to extend the scope of the search by augmenting the word set. Instead of looking for only the table metadata, this set of words can be extended with other related words, so when searching the text the bandwidth of the search is widened. Stemming, as discussed in Section 2.3.2.1, can be used to reduce the variant forms of word's representation to a common root form. This improves the ability of the system to match the text vocabulary and find the parts of the text that might describe the table.

Unfortunately, stemming will also add words that are not related to the search or document domain. This can widen the searching so that more text is identified some of which is not relevant. Also table metadata are usually presented in an abstract format which does not require stemming. We are not going to use stemming in this project, although it might be used in future work to see the level of improvement that it might give to the system.

Another mechanism that can be used in enriching the word list is the metadata synonyms. This can be done using dictionaries, thesaurus and ontologies. Again the dictionaries and thesaurus will add more words than the relevant words. This additional words are related the metadata words, but the relationship is through a general sense and is done without taking into consideration the domain that the word or table is in at the moment. On the other hand, a domain ontology is more focussed and contains only the word that is related to the metadata semantically i.e. in the current context. In our research, we use a domain ontology to augment the table header and enrich the searching list. We extract the concept from the domain ontology which the searching word is related to and all synonyms associated with that concept, together with the concept relationships. This gives us a more focussed enhancement of the words from the table header than a dictionary or thesaurus would provide.

5.2.2.2 Searching mechanism

The project is going to analyse all the content of the document and the documents that we are using is relatively short. Therefore, there is no need to concentrate on a specific part of the document on which to constrain the search initially. There are many mechanisms that can be used in searching the documents, and they concentrate on saving either time (faster searching) or minimising the effort used. In our case, the time that the system takes to search the documents is not crucial as the documents are short, also as it is a research project, prof of concept is required not search speed. The improvement that a faster searching mechanism can give is measured by seconds or even milliseconds with our set of documents. Also, the effort involved in searching these types of document is not huge, knowing that documents used are relatively short. One of the searching mechanisms that can be used is a tree approach. This reorganises the content of the table into a tree structure which contains the document as the root of the tree, and the paragraphs, sentences and words as the levels in the tree. We use this approach because searching a tree is much simpler than conducting a search in a free text document. Also, knowing the exact location of the detected word and the ability to go back to this same location quickly is one of the advantages of the tree mechanism.

5.2.2.3 Extraction process

After detecting the required words in the text, we have to extract them and surrounding text. There are a number of words in the text that have a related word next to them (e.g. the sentence 'the currency is in pounds'). If we extract only the word, then we will miss an important related part of the context related to it. Therefore, the project has to consider extracting the detected word and words that are related to it in the same sentence and provide semantics for the concept.

5.2.2.4 Representation methods

Representing the extracted words is one of the most important parts of this project. The system has to consider representing the extracted words along with all related words from the table metadata and the related concepts from the domain ontology. we are using two methods for representing the detected semantics, namely semantics-units and enhanced metadata.

5.2.3 Output

The output of the system consists of two units. First, the representation of the detected semantics with the value it represents as an isolated unit from the table. The second way of representation is representing the detected semantics as the table itself.

5.3 Programming language

Because we are using documents from the Web and also the domain ontology that we are using is built using DAML+OIL, which has an ontology viewer, which is built using Java and can search and extract the concepts from the ontology, we therefore decided to use Java as the programming language for our system.

5.4 Evaluation

To evaluate the project, an experiment has to be carried out to ensure that the project and the hypotheses are correct. The experiment has to be performed twice, manually and automatically, using the system to evaluate the performance of the methods used in building the system. This experiment environment uses 300 documents and carries out a number of tests that evaluate the approaches used in searching and detecting the related semantics in the text, comparing their performance and identifying signs.

5.5 Summary

In this chapter we have discussed the the design of the system and its components. we have discussed also the content of the input documents. Also, a number of techniques involved in the analysis of the document components like word collection, searching process, extraction methods, and representational methods have been discussed. We have illustrate the reason of choosing Java programming language to implementing this system.

CHAPTER 6

Semantic and Representation Framework

6.1 Introduction

SRD (Semantic and Representation Detection framework) is to be a prototype system for discovering and interpreting the context information about tables present in the text of a document containing tabular data, and prepare them for interoperation with other data. Figure. 6.1 shows the proposed system architecture of the SRD system, which will extract and structure the context data about a table held within a textual document.

It consists of two main units and each unit has a number of processes and sub-units. The first is the **Detection Unit**. The main purpose of this unit is identifying, extracting and representing the information about the context of a table's elements given within the text. This information will be used to enhance the table's metadata and thus lead to better use of its contents. It operates on the documents, which are represented in ASCII characters and have tabular data in them. The unit consists of four sub-units, namely

1. **the detection process** which analyses the content of the document and detects potentially useful context information about tables present in a document;

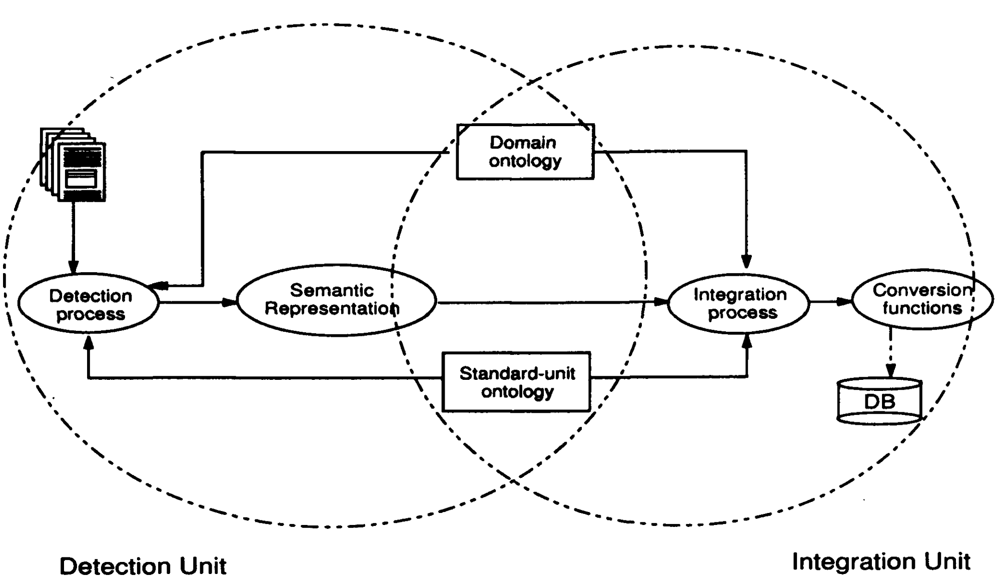


Figure 6.1: Semantic and representation detection framework architecture

2. **the domain ontology** which provides information about related terms, the representation and description of terms in the domain of the document, and provides details about how to convert between different representations used in the domain;

3. **standard-unit ontology** which is used to discover some of the value representations in the text. It contains details of how different concepts are related; and

4. **semantic representation** which represents the extracted semantics and representation information as metadata, which are stored for further processing and will enhance the table metadata.

The second unit is the **Integration Unit**. This is concerned with integrating the detected semantics with other semantics to create a description of a table and its representation. Documents are created by different authors who represent the data using their knowledge of the domain. Therefore the detected semantics might need to be converted into a common representation and terminology before data from this table can be integrated with data from another. These unit consists of five sub-units namely:

1. **domain ontology**. It defines a conceptual model of the underlying functional domain and provides a basis (shared vocabulary, relations and standards) with which the meaning of data in different sources can be described [GMS96]. We use this domain specific knowledge base for resolving and extraction of semantic conflicts. Thus, a domain knowledge base serves as a common basis for the representation of data and metadata;

2. **standard-unit ontology** this defines a series of SI units of measurement and other commonly used units that do not belong to SI units. It includes a Standard-Dimensions Ontology, which defines a series of physical dimensions (e.g. mass, time, length, temperature and electrical current) for different quantities [Per99]. This ontology is going to be used in discovering some of the value representations, which are buried in the context of document;

3. **conversion functions** these are functions which can convert an attribute of a semantic unit from one representation to another so that the attributes from different sources are in a common representation before integration of data;

4. **integration process** which integrates semantic units and table metadata with the corresponding data from a database; and

5. **the database (DB)** which is the database that is going to be used either to store the extracted semantics or to integrate these semantics with data from previous searches stored in the database.

6.2 Documents

The documents are the input to the system. They have been extracted from a data source in a subject domains. These documents, which are in ASCII format, contain tabular data surrounded by text and the aim is to analyse the document to identify the metadata for the table where the metadata is a combination of information extracted from the table and the text.

6.3 Detection Unit

This unit is the main unit of the framework; it uses the domain ontology, the standard-unit ontology, and the tabular data in the document to detect the hidden semantics in the document text which relate to the table. After detection; these semantics need to be extracted from the text and then represented in a suitable form for further processing and linking with other metadata.

6.3.1 Detection Process

This process analyses the content of a document and detects the useful context information about tables present in a document containing tabular data. The discovery of semantic context corresponds to the task of finding useful knowledge about a table in a document. Our framework contains a complex process for detecting semantics in textual documents. The proposed semantic detection process is depicted in Figure. 6.2. It consists of a word collection process, a searching process, and an extraction process.

6.3.1.1 Word collection process

The first step in the detection process is to collect the words that are going to be searched for in the surrounding text. This process is one of the most important

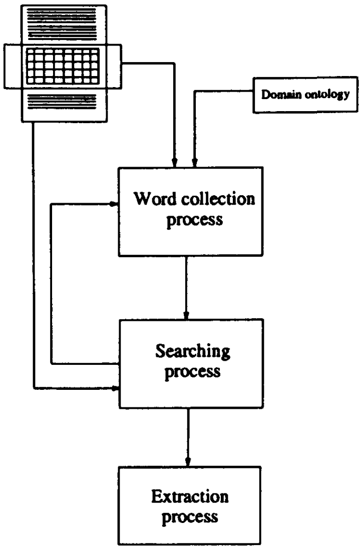
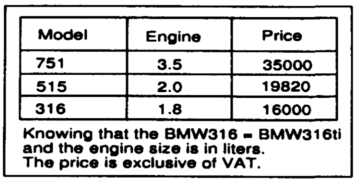


Figure 6.2: Semantic detection process

processes in the detection. If we fail to gather the right words, then some of the information related to the table will be missed or overlooked. There are a number of techniques that can be used in this task, as discussed in Section 2.3.2. We use in our system two methods, namely table data and knowledge-related words.

6.3.1.1.1 Table data This covers the use of the table headings as keywords to search for related semantics in the paragraphs adjacent to the table. We know that the table's column and row headers are metadata indicating the main concepts that the table represents. Therefore, by concentrating on this table metadata, we are hoping to ensure the correctness and accuracy of the detected contexts to the table.



Model	Engine	Price
751	3.5	35000
515	2.0	19820
316	1.8	16000

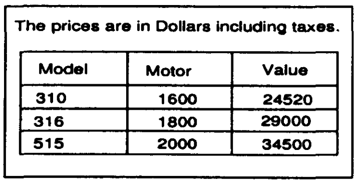
Knowing that the BMW316 = BMW316ti
and the engine size is in liters.
The price is exclusive of VAT.

Figure 6.3: Example of Car Prices

Taking the field 'engine' in Figure 6.3 as an example of this metadata, searching the adjacent paragraphs could result in detecting the context 'Engine size is in litres' in the text. This gives us information about the representation units used in the column of the table.

Another approach to locating metadata in the text is to use the table title (e.g. 'in Figure 6.4 : Example of Car Prices') and search for parts of the title in the text eg 'Figure 6.4'. We can also use the rest of the table title text which gives us the information that this table is about car prices. This leads to an understanding that the field name 'value' is equivalent to 'price' information provided by a domain ontology. We can then search for both value and price in the adjacent document text. This could result in finding the context text 'All prices are in dollars including taxes, ' which can be analysed at a later stage in our system to give two pieces of additional information, namely the price in dollars and that the price includes tax.

6.3.1.1.2 Knowledge base (ontology) related words In many situations, the table header alone is not enough to describe or find the semantics of that table which is in the text. Another approach is then used. The table header is used to extract corresponding concepts from the domain ontology and a search is made for the header and all its related synonyms in the text is performed.



The prices are in Dollars including taxes.

Model	Motor	Value
310	1600	24520
316	1800	29000
515	2000	34500

Figure 6.4: Example of Car Prices

For instance in Figure 6.4, the corresponding concept in the domain ontology for the field 'value' is price. If we search the adjacent paragraphs to the table using the augmented header it might results in the context 'prices are in US dollars'. In the domain ontology, the concept price has an associated concept currency, and currency is represented in monetary units, eg. US dollars, British pounds, or any other currency. We can say that the value 'price-currency=US dollars' is a useful semantic, in that it identifies the representation units being used in the table. Therefore, we can use an ontology to help in augmenting the table metadata and to enrich the search concept by adding the concepts which are related to the table data, as being synonyms and related for that concept.

6.3.1.2 Related word representation

The related words, which have been collected from the table and the knowledge base, will have a representation that helps in finding, extracting and representing semantics in the document text. The word and its related concept will be joined together and represented as a node with the format (C,W), where C represents the knowledge Concept, and W represents the related Word. This node will be used for extracting and representing the semantics of a document.

6.3.1.3 Searching algorithm

We represent each document as a four level tree (see Figure. 6.5). Starting from the bottom level (words), we compare a word with all the nodes identified by the word collection process described in section 6.3.1.1. If a word from the tree matches one of the nodes, then we perform the extraction process. Often, a single relevant

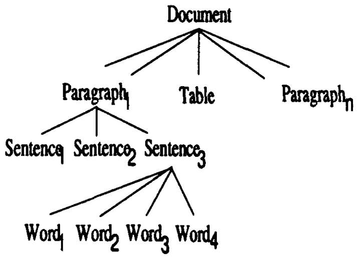


Figure 6.5: Four Levels Tree

semantic is represented in one sentence [RL94]. If the word is found in that sentence, move to the next sentence.

If no matched word found then move to the next sentence until end of tree is reached. This algorithm assumes only one relevant piece of semantics is presented in a sentence. This is usually the case and the algorithm can be adjusted if it is not.

6.3.1.4 Word extraction algorithm

There are two types of semantics which might need to be extracted from a document

1. Direct semantics, this is information which is beneficial by itself (i.e. Car, pounds) and thus its extraction is meaningful and useful.

2. Representational semantics, this type of semantic is not beneficial by itself as it needs a representational unit (value, unit of measurement, or type) so that its semantics can be interpreted. For example, the word 'colour' is not enough, unless it is combined with its value 'red'. The domain ontology coupled with the standard-unit ontology give us the expected representational values for semantics. Therefore in order for representational semantics to be useful, we need to extract them with their value.

6.4 Domain ontology

In our framework, we have built a cars' domain ontology using DAML+OIL (see Section 4.3.1.2.3. The ontology that we built consists of 15 classes which contain all objects in that domain (e.g. car, engine, and price). Each class has a number of properties (see Appendix A), In the second area of interest we used an already existing chemistry ontology obtained from the DMAL+OIL Web page (www.DAML.org). We used these two domain ontologies for evaluation purposes in our work, otherwise one is enough in application.

6.5 Standard Unit ontology

The other type of ontology we use in the system, is a Standard-Unit Ontology [GO94], which defines a series of SI units of measurement, and other commonly used units that do not belong to SI units. It includes a Standard-Dimensions Ontology, which defines a series of physical dimensions (e.g. mass, time, length, temperature and electrical current) for different quantities [Per99]. A unit of measure is an absolute amount of something that can be used as a standard reference quantity.

Like all quantities, units have dimensions, and units can be defined in the same way as any other scalar quantity. For example, the kilogram is a unit of measure for the mass dimension. The unit called 'pound' can be defined as a mass quantity equal to the kilogram times some constant, just as the quantity 50kg is equal to the product of the unit called 'kilogram' and the real number 50. What makes the pound special, compared with quantities like 50kg, is a matter of convention.

This ontology is going to be used in discovering some of the value representations, which are buried in the context of document. It also will be used in integrating these documents and eliminating any representational conflicts that might occur.

6.6 Integration Unit

This unit is the second unit in the framework; it uses the extracted semantics along with their corresponding concept from the domain ontology to create a semantic unit for each concept. Before integrating the targeted semantic unit or the enhanced metadata, a conversion function might need to be called before integration so the heterogeneity between semantic units can be resolved. This unit contains two processes: integration process and conversion functions.

6.6.1 Semantic Representation

After detecting and extracting the corresponding contexts, our internal model represents this information in two ways: as semantic units or as table-enhanced metadata. This can then be used to link corresponding information in different tables together.

6.6.1.1 Semantic Units

As mentioned in Section 3.6.1, a *semantic unit* comprises a datum together with its associated semantic context, consisting of a flexible set of meta-attributes describing the meaning of the datum. The semantic unit representing a value v can be given as a triple $(C,\ V,\ ST)$, where C represents the knowledge concept derived from the domain ontology representing the corresponding value, V represents the value, and ST represents the semantic contexts that have been discovered in the text. Taking the examples in Figures 6.1 and 6.2, the price for the car model 316 can be represented as

(*price*=16000, *currency*=pounds, *VAT*=exclusive)

and

(*price*=29000, *currency*=dollars, *TAX*=included)

If an appropriate domain ontology is used, one can find that *value* = *price* and that each *price* must have a *currency* and a *VAT* component.

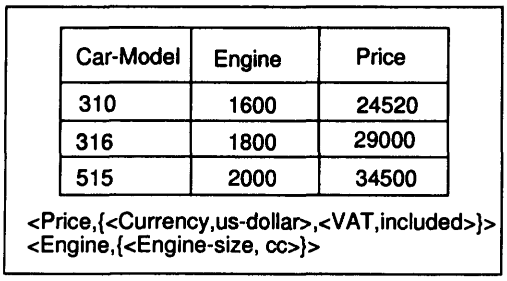
These Semantic Units are suitable for integration and comparison with data held in another database but might require definitions of some of its attributes, so that we can ensure they are referring to concepts with the same meaning and representation. For example, a definition of *VAT*, which is Value Added Tax gives an assurance that VAT and TAX can refer to the same concept *VAT*. We might need a conversion function to convert the value between different representations, even when the concepts have the same meaning.

6.6.1.2 Enhanced Metadata

Table metadata, e.g. header describing the full table with its corresponding contexts. The table headings in the original table are held as part of this table metadata. These metadata are enhanced with any semantic and representation context found in the text. In addition, all the table fields have a concept label associated with them that specifies the relationship between the table field and the real world aspects it describes. This table enhanced metadata is suitable for storing the data in the database and for integrating it with data from another database. We represent the detected semantics for that table as follows:

$$EnhancedMetadata =< C, SA >$$

where C represents the knowledge concept derived from the domain ontology which relates to a corresponding data element in the table, and SA represents the semantic contexts that have been discovered in the text (see bottom lines of Figure 6.6).



Car-Model	Engine	Price
310	1600	24520
316	1800	29000
515	2000	34500

<Price,{<Currency,us-dollar>,<VAT,included>}>
<Engine,{<Engine-size, cc>}>

Figure 6.6: An example of enhanced table metadata

6.6.2 Integration Process

In this process, we are trying to integrate extracted semantics to represent the search results to the user. In the case of semantic units, the integration process contains four steps: are the semantic units concepts the same? If yes, then are they have a similar attributes? If yes, then are these attributes represented in different ways? If yes then, these semantic units need a conversion function. (See next chapter for more details).

6.6.3 Semantic Conversion

An important part of our framework is its conversion functions. This ensures that there is no heterogeneity between integrated semantic units. Our conversion function consist of two main types: elementary conversion functions and conversion functions for multivalued semantic units. (See section 7.3 for more details).

6.7 Summary

In this chapter, we have discussed the architecture of our SDR system. We have also presented the detection process which is going to detect the hidden semantics, using different approaches. After detecting the hidden semantics, they will be extracted and represented in a format called semantic units and enhanced metadata. Suitable conversion functions will be used if necessary, and then the integration process will be used to integrate the semantics.

CHAPTER 7

Utilising Semantic Conversion Functions to Link Tabular Data

7.1 Introduction

As discussed in the previous chapter, our framework has two main units - the Detection unit and Integration unit. The detection unit has been discussed in chapter 6, along with the sub-processes in it. After detecting of the hidden semantics in the text, these semantics have to be represented as semantic-units. These semantic-units, then need to be integrated to make a coherent database. The process of this integration of semantics-units is the focus of this chapter. Before integrating the recovered semantic-units, they have to be semantically equivalent. If these semantic-units are not equivalent, then a conversion function is needed to bring them to a common form. In this chapter, we give a description of the integration unit of our SDR system. It contains two main parts- integration process and conversion functions.

7.2 Integration Process

The goal of this process is to integrate appropriate semantic units together. The initial stage of this is to check that there is no heterogeneity problems between the semantic units. To do this, we use two steps: concept matching, and attribute and representation checking.

7.2.1 Concept matching

Integrating data from different sources is based on finding similarities or differences between data items. To do so , we establish similarity relations between concepts. Detection of a similarity relation is based on conceptual relations [Gua97a] [GW00][HG01]. Conceptual relations (represented by β in the following levels of similarity) are definitions of concepts by logical axioms, and these conceptual relations are defined in the domain ontology. For example, 'driver' is a conceptual relation and its semantic relation is: $[driver] \sqsubseteq person(x) \cap Car(y)$ ie. a driver is a person with a car.

The levels of similarity between two concepts can be identified as:

1. Disjoint concepts: This level has the lowest degree of similarity. Two concepts are disjoint if the conjunction between them implies false. e.g. sister and father.

$$\beta Ci \cap \beta Cj = false \Longrightarrow Ci \neq Cj$$

2. Equivalent concepts: This level has the highest degree of similarity. If the semantic definition of the two concepts is equivalent, then the defined concepts are equivalent. For instance 'vehicle' and 'car' are equivalent, because they have the same semantic definition.

$$\beta Ci = \beta Cj = True \Longrightarrow Ci = Cj$$

3. Sub-concept: If the semantic definition of a concept i is an implication of the semantic definition of concept j , then Ci is a sub-concept of Cj. For example, 'pickup' is a sub-concept of 'Car'.

$$\beta Ci \cap \beta Cj = Ci \Longrightarrow Ci <= Cj$$

7.2.2 Attributes and representation unit checking

Even if two concepts are equivalent conceptually, that does not mean they can be integrated directly. When trying to integrate or merge concepts from different knowledge bases to build a global schema, concept matching is enough to identify equivalent concepts, but in our case we are trying to integrate concepts from different sources with different attributes and different representational units. Therefore, checking these attributes is essential as well if a meaningful integration is to occur. Before we integrate equivalent concepts, we have to check the differences and similarities between their attributes. We can do this by referring to the domain ontology and comparing the attributes for the concepts bearing in mind synonyms and homonyms. For instance, when integrating the concept 'car- engine' from different sources they might have attributes that affect the type of that concept. For example, the engine size, shape, and power might not be equal, and this can affect the integration of these concepts. After identifying the differences in attributes for the concepts to be integrated, a conversion function must be applied when appropriate to eliminate these differences.

7.3 Semantic Conversion Function

To reduce the heterogeneity occurring between different metadata description, before integration, the potentially similar attribute values that have been detected must be converted to match each other in representation. We treat the Semantic-Units and enhanced metadata in the same way, except that when converting a value V in a table, we convert all values for that data element in the table. We have two types of conversion function, known as elementary and multivalued [AG03b].

7.3.1 Elementary conversion function

We call a conversion function an *elementary conversion function* if and only if one attribute from the semantic unit is going to be used in the conversion process. For example, suppose we have the semantic units

$$Semantic - Unit(price1) =< price, 215, \{currency, 'us - dollar'\} >$$
$$Semantic - Unit(price2) =< price, 150, \{currency, 'GBP'\} >$$

Even though they represent the same concept, they have attributes with different representations. In order to integrate these semantic units we have to convert the attribute given in ' us-dollar ' to match the other semantic units attribute by converting it to 'GBP', or vice versa. Therefore the conversion function

$$(\{currency, 'GBP'\}, < price, 215, \{currency, 'us - dollar'\} >)$$

is an elementary conversion function in this case because it has one semantic attribute to be converted. This is a simple semantic unit $SU(C) = < C, V, SA1 >$ with a semantic attribute $< C1, RepType(c1) > \in SA1$. An elementary conversion function is a function that converts the value $V \in Domain(RepType(c1))$, represented by $SA = < cl, vl >$ to the value $V2 \in Domain(RepType(cl))$ represented by $SA = < cl, v2 >$ i.e.

$$CF(\{< cl, vl >\}, < C, V1, \{< cl, v2 >\} >) = < C, V2, \{< cl, vl >\} >$$

For example if we have the SU(pricel) and SU(price2) associated respectively with the semantics attributes

$SA(currency,' us - dollar')$ and $SA(currency,' GBP') \in RepType(currency)$ we can write the corresponding conversion function as

$CF(pricel) = (\{< currency,' GBP' >\}, < price, 215, \{currency,' us - dollar'\} >)$
$= < price, 135, \{< currency,' GBP' >\} >$

with ' 1 us-dollar = 0.624 GBP ' as a mapping rule which is obtained from either the domain ontology or the Standard-Unit ontology.

7.3.1.1 Types of Elementary Conversion Functions (ECF)

Elementary conversion functions can be categorised into four types as follows:

7.3.1.1.1 General Conversion Function An elementary conversion function is a General conversion function if for every quantity among different units of measurements (RepType) we can give a semantically meaningful mapping rule. Having a simple semantic-unit(C)= $< C, V, SA1 >$ and a semantic attribute(SA) $=< cl, vl >$ where $vl \in Domain(RepType(cl))$ an Elementary conversion function is a Total ECF if it converts the value $V \in Domain(RepType(cl))$, represented by $SA = < cl, vl >$ for all $vl, v2, v3, ..., vn \in Domain(RepType(cl))$, within a certain mapping rule. For example, the CF(price) is a General conversion function, because we are able to give a specific mapping rule between a CF and another different currency as shown in Section 7.3.1.

7.3.1.1.2 Partial Conversion Function As opposed to a General conversion function, the ECF is called a Partial CF if we cannot give a clear and general mapping between one representation and another, i.e. the conversion of locations, like from city to country, and any values that have generality or specificity. For example, we have a semantic concept 'producer' with Domain(Producer) = {Country, Company, ...}. A conversion function which converts the maker of a car from one type of producer to another is a partial conversion function, i.e.

$CF(\{< producer,'country' >\}, < Maker, Rover, \{< producer,'company' > \} >) = < Maker, UK, \{producer,'country'\} >$

As we can see in this example, we can not give one specific mapping rule to convert all different semantic values. The 'country' of the 'company' 'Rover' is 'UK', but we can not use this mapping rule to all other values of the semantic 'Maker'.

7.3.1.1.3 Order-Preserving conversion function An elementary conversion function is called an order-preserving CF if the value of the concept and the converted value for all RepTypes have the same order concerning $' <' or' >'$. Having the conversion function,

$$CF(\{< \mathrm{cl}, val >\}, < C, V1, \{< \mathrm{cl}, v2 >\} >) = < C, V2, \{< \mathrm{cl}, va2 >\} >$$

and

$$CF(\{< c1, vb2 >\}, < C, V1, \{< c1, v2 >\} >) = < C, V3, \{< c1, vb2 >\} >$$

We call this type of conversion function an Order-preserving CF if all values

$va1, vb1, vc1, ..., vn1$ have the same order when the conversion result

$va2, vb2, vc2, ..., vn2$ is ordered, with respect to $' <'$ or $' >'$

In [SSR94] Sciore, Siegel, and Rosenthal, a mapping function concerning the semantic aspect CodeType specifying the underlying character code, e.g. ASCII or EBCDIC, of a given character value is given as an example of an elementary conversion function that is not order-preserving with regard to$' <' \; or' >'$. The following semantic objects:

$< CharCode, 48, \{< CodeType,' ASCII' >\} >$ and
$< CharCode, 240, \{< CodeType,' EBCDIC' >\} >$

are representations of character '0', and the semantic objects:

$< CharCode, 65, \{< CodeType,' ASCII' >\} >$ and
$< CharCode, 193, \{< CodeType,' EBCDIC' >\} >$

are representations of the character 'A'. Thus, the corresponding conversion function CodeType is not order-preserving with regard to $' <'$ or $' >'$, because by using ASCII as the underlying character code, we have '0' represented by a smaller value than character 'A', whereas when using code type EBCDIC, the inverse relationship holds.

7.3.1.1.4 Lossless Conversion Function We call an elementary conversion function a lossless CF if the conversions between the different measurement types do not lead to any loss of information. For example, most of the conversions between measurement types like km, mile, yard, kg, pounds, etc. involve converting values from or to any one of these types with any number of conversions, and this conversion function does not affect the value of the last measurement type. For example, we can convert a value from kg to GP pounds and then to US pounds and we will have the same value if we convert it from kg to US pounds directly. On the other hand, converting currency will affect the last amount of money converted. For example, converting a currency from 'us-dollar' to 'GBP' has a different value from converting the same amount from 'us-dollar' to 'EURO' and then to 'GBP', and we call this conversion function a lossy CF.

7.3.2 Conversion Function for Multivalued Semantic Units

When more than one semantic context is going to be used for the conversion, then we call this CF a multivalued CF For example, the semantic units

$Semantic{-}Unit(price1) = < price, 215, \{< currency,' us{-}dollar' >, < Scale, 1000 > \} >$

and $Semantic{-}Unit(price2) = < price, 150000, \{< currency,' GBP' >, < Scale, 1\} >$

To be able to integrate these semantic units, we have to convert the attribute ' us-dollar ' to match the other semantic unit 'GBP', having in mind the differences in Scale between '1000' and '1' knowing that the price in the Semantic-Unit(price1) is actually '215000' not '215' where the price in Semantic-Unit(price2) is exactly '150000'. Therefore the conversion function

$(\{< currency,' GBP' >, < Scale, 1\}, < price, 215, \{< currency,' us - dollar' >$ $, < Scale, 100 >\} >)$

is a multivalued conversion function in this case, because it has two semantic attributes to be converted.

Having a multivalued semantic unit $su(C) = < C, V, SA1 >$ and a semantic attribute $\{< C1, RepType(c1) >, < C2, RepType(c2) >\} \in SA1$.

A multivalued conversion function is a function that converts the value $V \in Domain(RepType(c1)$ and $RepType(c2))$ represented by $SA = \{< c1, v1 >, < c2, v1 >\}$ to the value $V2 \in Domain(RepType(c1)$ and $RepType(c2))$ represented by $SA = \{< c1, v1 >, < c2, v1 >\}$ i.e.

$$CF(\{< c1, v1 >, < c2, v1 >\}, < C, V1, \{< c1, v2 >, < c2, v1 >\} >) = < C, V2, \{< c1, v2 >, < c2, v2\} >$$

For example, if we have the SU(price1) and SU(price2) associated respectively with the semantics attributes
$SA\{< currency,' us - dollar' >, < Scale, 100 >\}$ and $SA\{< currency,' GBP' >, < Scale, 1 >\} \in RepType(currency)$ and $RepType(Scale)$
Then we can write the corresponding conversion function as

$CF(price1) = (\{< currency,' GBP' >, < Scale, 1 >\}, < price, 215, \{currency,' us - dollar', < Scale, 100 >\} >) = < price, 13416, \{< currency,' GBP' >, < Scale, 1 >\} >$ with ' 1 us-dollar = 0.624 GBP ' as a mapping rule .

7.3.2.1 Properties of Multivalued CF

1. A MCF will have the same properties as ECF.

2. An unrestricted CF is a MCF where the order between the converted values is not important. The example above (Section 7.3.2) illustrates that the order between the conversion of currency and scale is not important.

7.3.3 Conversion function algorithms

The algorithm that we use for the conversion functions is divided into two classes:

1. Simple Transformation: When we have a single value that is going to be converted, we use a simple algorithm for their conversion as follows:

   - Transform (X) (input unit, output unit); Where X is the value that is to be converted, input unit is the current representation, and output unit is the targeted representation For example, converting from mile to kilometer $Transform(20)(Mile, Kilometer)$
   - Look up the targeted representation in the ontology: We found $(define{-}instanceKILOMETER(unit{-}of{-}measure) := (/{*}MILE1.609):$ $axiom - def(= (quantity.dimensionkilometer)length - dimension))$
   - Find formula (mapping role) in the ontology := $(/ * MILE1.609)$
   - apply formula to (X) to get the results
     Kilometer := Miles * 1.609
     = 20* 1.609 = 32.18

2. Complex transformation In some cases, the input unit consists of more than one representation (e.g. mile/gallon). In this case, we apply a complex transformation function to convert this value as follows

- C.transform (X) (input unit, output unit); Where X is the value to be converted, input unit is a complex representation which consists of two variables, and output unit is the targeted complex representation which consists of two variables for example, converting from mile/gallon to kilometer/ liter.

  For example

  $C.transform(250)(mile/gallon, kilometer/liter);$

- Look up the targeted representation in ontology as in simple transformation.

- Because the complex transformation consists of two variables , one is multiplied and the other is divided, therefore we divide the complex transformation into two parts

  (a) multiply.transform (X, mile, kilometer)

  = 250*1.609 = 402

  (b) divide.transform (Y, gallon, liter)

  = 402/3.8 =105.9 kilometer/liter

7.4 Summary

In this chapter we briefly introduced the semantic-unit, which represents the extracted data together with its related semantics. Also, we have described the similarity relation used to find the relation between different concepts, and we have illustrated the four levels of similarity between the integrated concepts and the similarities between their attributes and representational types. After stating the similarities and differences between concepts, we discuss the types of conversion functions, together with their properties. We have two type of conversion functions, Elementary and Multivalued. This work can be used to link tables in semi-structured documents with information held in other such documents or databases to form a federated database by detecting SUs with common conceptual meanings and converting them to a uniform representation using these functions.

CHAPTER 8

Prototype System

8.1 Introduction

We have discussed the semantic and representation detection framework in Chapter 6. In this chapter we are describe the architecture of the prototype system that we have built to represent the framework and to analyse the content of documents from different data sources. These documents need to be analysed to detect the hidden semantics related to the data held in tables in the same document. The results of this prototype system will be used in the next chapter as input data for our experimental analysis of the approaches. Therefore, the goal of this chapter is to show how we implement the SRD framework.

8.2 Software architecture

Most of the software has a common architecture which contains three main parts, namely: Data input, System processes and Data outputs as shown in Figure 8.1.

8.2.1 Data Inputs

All data sources and external ontologies can be treated as inputs to the system. In our system, we have two types of input.

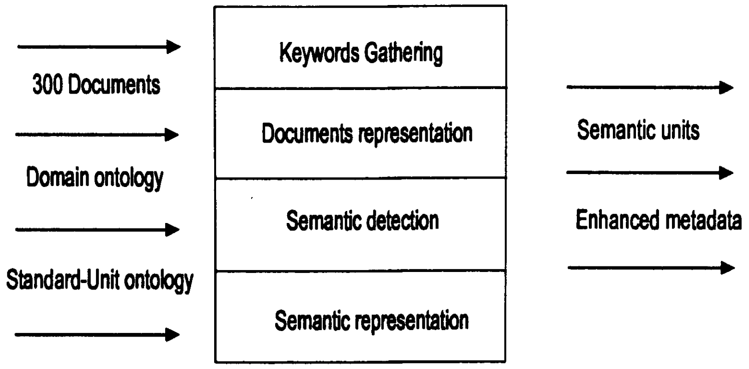


Figure 8.1: Software architecture

1. Documents. This is the main type of input to the system. We have 300 documents extracted from four data sources which we use in our experiments. These documents are going to be analysed to detect and extract the hidden semantics in them. Each document contains tabular data and text surrounding that table. We are concentrating on the tabular data and trying to find related semantics to the table in the surrounding text.

2. Ontologies: We have two types of ontology as input to the system. The first is a cars domain ontology. This ontology is used to enrich the searching keywords which are related to the tabular data in input documents about cars. The second ontology is the standard unit ontology which is used to add different value representations to the searchable keywords. The standard unit ontology is also used in the conversion functions between different value representations of the detected semantics. See Section 4.2 for more information on this aspect.

8.2.2 Data outputs

Our system has two types of output as follows:

1. Semantics units. All detected semantics are represented in semantic units which contain related domain ontology concepts, detected semantics, and semantic values. We have discussed semantic units in detail in Section 6.6.1.1

2. Enhanced metadata: the semantics which are related to table data are represented within the table metadata as enhancements to that table, so that all the discovered semantics within the text about the table are used in this enhancement process.

8.2.3 System process

We have divided our system into four phases, as described in section 8.2.3.1 to section 8.2.3.4

8.2.3.1 Gathering searchable keywords

We will not be able to detect the appropriate semantics in the documents until we have identified the keywords to search for. As stated in section 6.3.1.1 , we are using a number of approaches to gather the words that are going to be used in the searching process of the documents to detect hidden semantics. The first approach is using table metadata, where we extract the table header, footer, and its data, and then add them to the searching table as the first set of searchable keywords.

We assume that the structure of tables is not an issue, because it has been tackled in the previous work at Cardiff University by Hodge [HGF98, Hod01]; therefore we assume that the table metadata is going to be extracted directly from the table. The second approach is to use the domain ontology to add extra keywords to the searching table. Using the keywords from the first approach, we search the domain ontology for related concepts and relations to these keywords. We built our cars domain ontology using the DAML+OIL language. This system has a very good tool to browse and search ontologies prepared using DAML+OIL called DAML viewer.

DAML viewer gives us the ability to search the ontology and find all related

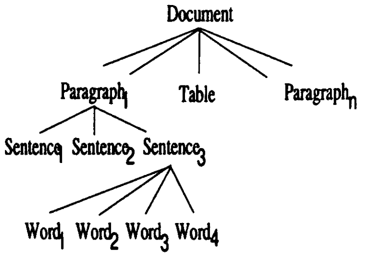


Figure 8.2: Four Levels Tree

concepts, relations and synonyms for a searched word. The result of this search will be a set of related words which are added to the searching table. These data will be stored as a combination of the table-related word, the detected word from the domain ontology, and the related concept from that ontology. For example, horsepower can be found in the ontology, and it has as a synonym, HP; therefore this word can be stored as 'horsepower', 'PH', 'engine size'). After representing all the related keyword in the searching table we move to the second phase which is representing the documents in a tree format.

8.2.3.2 Document representation

We represent each document in a tree format as it makes it easier to search the content of that document. As shown in (Figure 8.2) we start the tree root with the document number. This root has a number of branches which represent paragraphs in that document. All paragraphs presented before a table will be represented in the left branches of the tree and the paragraphs after the table in the right branches. This is for our analysis of where we detect semantics about a table in the document. Each branch has a number of branches which represent the sentences of that paragraph. We distinguish between sentences by the full stop at the end of each sentence. Each node representing a sentence has a number of branches which represent words in that sentence. For each document, we are going to have two items at the end of this phase, a search table and a document tree.

8.2.3.3 Semantics detection

In this third phase, we search the document tree for words equivalent to the words held in the searching table. We start this phase with the tree leaves (words) in the first sentence and first paragraph, and we compare it with the searching table. If there is a matching word, the whole sentence will be extracted and stored separately for further processing. This stored sentence will be stored with the paragraph number, the sentence number, the matching word from the searching table, and the document number. Tree searching will continue to the next sentence. If there is no match, we go to the next word in the same sentence. Repeat this unit every sentence in the document has been processed. There are limitations in our system that should be remembered. This is related to the sentence structure and keyword detection. For example, a negative word might come before the detected semantic which will totaley change the meaning of that sentence i.e. " No VAT is included " is very different from "VAT is included".

8.2.3.4 Semantic representation

At the end of the searching phase for the first document we are going to have a number of sentences that contain hidden semantics about the words in the table header. We start with the first sentence and locate the matching word. We then begin searching the sentence for mathematical symbols (=, <, >, etc.) or (is, are, equal, etc.). If there is such a symbol located next to the matching word, then we extract the word with the next word or value after the mathematical symbol and store it as:

- The original word from the search table with the related concept from the domain ontology.
- The matching word in the sentence.

- The word or value after the mathematical symbol, along with the equation.
- The sentence location as it is needed later.

8.2.3.4.1 User role in the system A knowledgeable user is required to assist the system in identifying certain types of detected semantics. In some cases, the system needs the user to direct the recognition of semantics in some sentences. During the extraction process, the system searches the targeted sentence for any mathematical symbols. If no symbols found, then the system asks the user for his assistance. The user then reads the sentence to identify if there is any useful semantics present. The user can then add the new mathematical symbols found in that sentence.

8.2.3.4.2 From the data, we gather all matching words that relate to the same concept and create the semantic unit for that concept. If there is no mathematical equation detected, then this sentence will be sent to the user for manual inspection. If the user accepts it as a related semantic or a new mathematical equation, then this information will be added to the searching list.

8.3 Summary

In this chapter we have describes the prototype system for our framework. We start by describing the architecture of the prototype system. The prototype consisted of three parts, data inputs, system processes and data outputs. We have talked about each part and its components. We have two types of input to the prototype system, documents and domain ontologies. Also we have talked about the processes that the system performs, keyword gathering, document representation, semantic detection and semantic representation. The prototype system produces two outputs, semantic units and enhanced metadata.

CHAPTER 9

Experimental Design, Analysis, and Results

9.1 Introduction

As presented in Chapter 8, we have implemented a prototype software system that represents the SRD system. This software detects and represents the hidden semantics related to the table in the document. The results of analysing the test documents needs to be statistically analysed to find the significance of the detection process. The aim of this chapter is to perform an experiment using the prototype system to evaluate our framework and our objectives. We evaluate the different approaches used in detecting the semantics, and we apply statistical analysis techniques to do this.

9.2 Experimental Design

In our experiment, we use documents from different sources available on the Internet. The documents were collected randomly, and had to contain tabular data to be valid for the experiment. We have used documents from four primary data sources, as this allows us to detect any differences due to a data source as well. We will investigate the differences between commercial data and non-commercial data sources, and between scientific domains and non-scientific domains with the four types of data source.

In the experiments we have a number of input variables which describe the experiment. These are:

1. The number of domains used. In our experiment we use two types of domain- a scientific (chemistry) and a non-scientific (cars). By using different domains, we will see if there are differences in the number of detected semantics in documents representing different domains.

2. The number of data sources for the domain. For each domain, we used two sources of data. In the cars domain, we used data from the Imotors and Which web sites.

- Which web site: This is a consumer magazine Web site based in the UK. It gives independent, unbiased advice and evaluations on different products and services. This Web site issues a monthly report on different types of car. It reviews many aspects about the cars (e.g. performance, security, and price). Most of the reports contain tables which present the results of an evaluation. With each table, there is text data around it, which explains the table data and important issues about the car. We have extracted 50 pages from this web site. These were selected at random and are different evaluations published on different dates.(See www.Which.co.uk).

- Imotors web site: This is a commercial website. It is one of the fastest, easiest ways to purchase a car online. It has a USA national network of car dealers who provide competitive quotes, and an inventory of thousands of used vehicles. Imotors is now a car-buying service focused on effectively matching consumers with the vehicles they want. Imotors web pages normally consist of a table, which lists the cars that meet the user requirements and a paragraph or two which describe some of the values in the table. We have extracted 100 pages from this web site, which represent different search results.(See www.Imotors.com).

In the scientific domain (chemistry), we used two web sites - the Thermoset web site and the Eastman web site.

- Thermoset is part of The Lord Corporation and traces its roots to 1919. Lord's ideas produced inventions, and led to chemical formulations, bonding processes, elastomers, adhesives, coatings, bonded elastomer assemblies and many more discoveries. They provide on their web site, descriptions and specifications of their products. We have extracted 100 pages from this web site. This website is a commercial website. (See www.thermoset.com).
- Eastman Chemical Company (NYSE:EMN) is a global company which is one of the world's largest suppliers of polyester plastics for packaging. Headquartered in Kingsport, Tennessee, USA, Eastman manufactures and markets more than 1,200 chemicals, fibres and plastic products. On their web site, they evaluate a number of their chemical products so that customers can determine whether a product meets their requirements. We have extracted 50 pages from this web site(See www.Eastman.com).

It is probable desirably to use other data sources to determine whether they show the same behaviour. However there was insufficient project time to identify another source domain with enough documents meeting our needs for a specific domain. Thus, for the two domains chosen we have two different types of sources- a commercial site (Imotors and Thermoset) and an evaluation source (Which and Eastman). Further work is needed to confirm the results hold in other domains.

3. The number of documents from each data source for the commercial data sources is 100 documents and 50 for the evaluation data sources, which makes a total of 300 documents. We have tried to increase the number of documents but

consistent sources of these types of documents (documents containing tabular data) are hard to find. The number of documents within each source type is sufficient for the experiment, because as Admantios mentions in [DS97], 30 items can show the true behaviour of an item with respect to its characteristions being evaluated.

9.2.1 Experiment objectives

For our experiments, we have a number of objectives we wish to evaluate. These are:

1. To show that there are hidden semantics in documents that are related to a table in that document and they are significant to its interpretation.
2. To show the usefulness of a domain ontology in detecting and using these semantics.
3. To determine the difference between alternative approaches that can be used to detect such semantics and to identify the best approach.
4. To determine whether there is a relationship between the number of semantics in a paragraph and the distance between the paragraph which contains the semantics and the table itself.
5. To identify whether there is a significant difference in the number of detected semantics in different domains.
6. To evaluate the system.

To achieve these objectives, our experiments will analyse the documents and calculate a number of values. These analyses and calculations will be performed twice. The first time, we will perform them manually (fully human, no use of any other system). The second time we will use our system SRD without any human interference. Doing the analysis twice will allow us to compare the two methods and discover if there is a significant difference between a manual approach and the SRD approach. We then do comparisons with the SRD system to determine the effect of using an ontology in the determination of semantics and to identify where the semantics occur in the text in relationship to the table.

The values that are going to be calculated in this comparison are:

1. The number of semantics detected using the table header augmented by the appropriate terms from the domain ontology as the search key (enhanced keywords).

2. The number of semantics detected using the table information (data, metadata, footer).

3. The number of semantics detected in the paragraphs before the table.

4. The number of semantics detected in the paragraphs after the table.

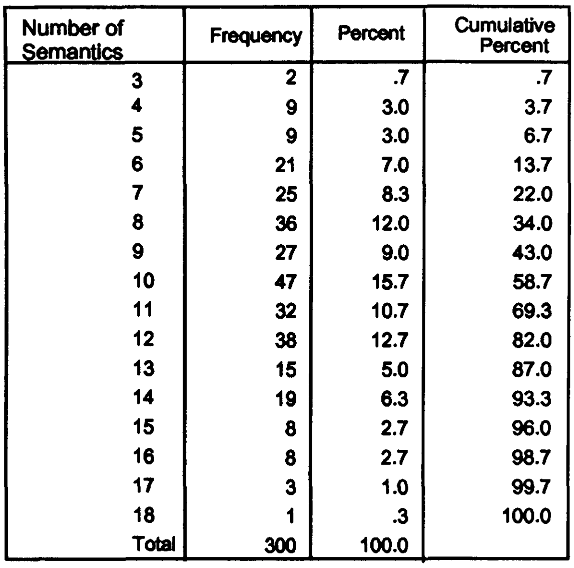
9.2.2 Experiment tests

In the first experiment we did a normality test on the manually detected semantics. This checks if the detected semantics are distributed normally in the sample document set. To do this, we performed three tests.

Figure 9.1 has four columns showing the results of this test for the 300 documents in the experiment. Its columns are:

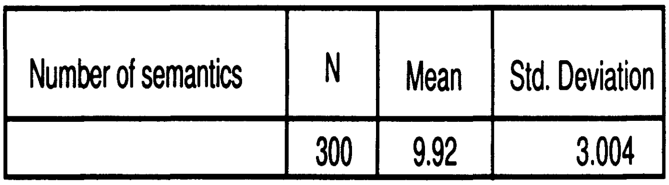
1. The number of semantics in a document (semantic interval), for example in the first row, the value 3 in this column shows the row refers to documents which have three semantics in their text.

2. Frequency: This shows how many times (number of documents) this semantic interval occurred in the 300 documents. For example, semantic interval '3'



Number of Semantics	Frequency	Percent	Cumulative Percent
3	2	.7	.7
4	9	3.0	3.7
5	9	3.0	6.7
6	21	7.0	13.7
7	25	8.3	22.0
8	36	12.0	34.0
9	27	9.0	43.0
10	47	15.7	58.7
11	32	10.7	69.3
12	38	12.7	82.0
13	15	5.0	87.0
14	19	6.3	93.3
15	8	2.7	96.0
16	8	2.7	98.7
17	3	1.0	99.7
18	1	.3	100.0
Total	300	100.0	

Figure 9.1: *Frequency of occurrence of number of semantics in the document set*



Number of semantics	N	Mean	Std. Deviation
	300	9.92	3.004

Figure 9.2: *Mean of the number of semantics detected in documents*

appeared in two out of the three hundred documents.

3. Percent: This compares the frequency of each semantic interval in the experiment's document set with the total number of documents (300). It is the percentage of the total documents with this semantic interval value.

4. Cumulative Percentage of documents in the set.

Figure 9.2, shows that the mean of the data presented in column 1 of Figure 9.1 is 9.92, and its standard deviation, which calculates the average amount of deviation from the mean, is 3. In general, the standard deviation 's' is defined as

$$s = \sqrt{s^2} = \sqrt{\frac{\Sigma(X_i - \bar{x})^2}{n-1}} \tag{9.1}$$

where X_i is an individual value of a semantic interval, $\bar{x}$ is the mean (9.92), and n is the number of intervals (16). Using this mean and standard deviation we can see that our data is normally distributed. Data is normally distributed when:

- 68.26 per cent of cases are within one std. dev. of the mean (69.34 in our experiment)
- 95.44 per cent of cases are within two std. dev. of the mean (98.95 in our experiment)
- 99.7 per cent of cases are within three std. dev. of the mean (100 in our experiment)

The second test is the Skewness test [BC98], which gave us a value of 0.101 . In general, a skewness value greater than 1.0 indicates a distribution that differs significantly from a normal, symmetric distribution, and the value we have is less than 1.0. Thus, the test is satisfied.

The third test uses graphs to check the distribution of our sample.

Figure 9.4 shows that the observed values (number of semantics in each document) are clustered around a straight line, which means that the sample is from a normal distribution.

The box plot graph in Figure 9.5 provides information about the shape and dispersion of the distribution of values. It shows that most of the values we have observed tend to be in the middle. In this case, the bulk of the observations are in the middle of the distribution.

The graph in Figure 9.3 shows that our data are normally distributed. This tells us that our data did not happen by chance, and the mean of 9.92 shows that the documents have a reasonable number of hidden semantics in the text. Therefore,

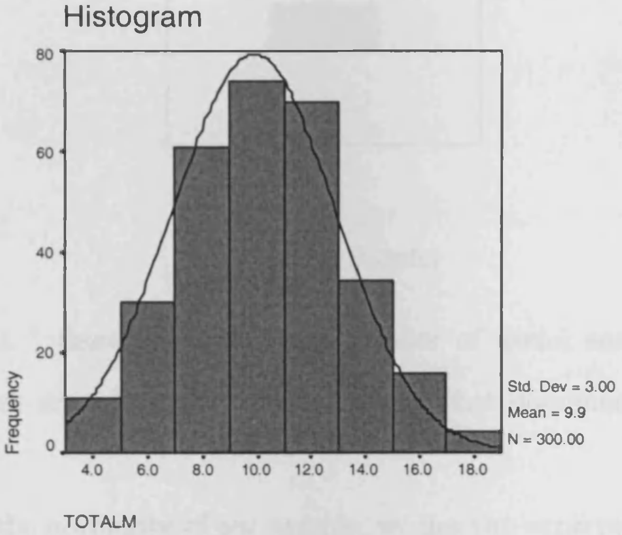


Figure 9.3: *Normality Graph*

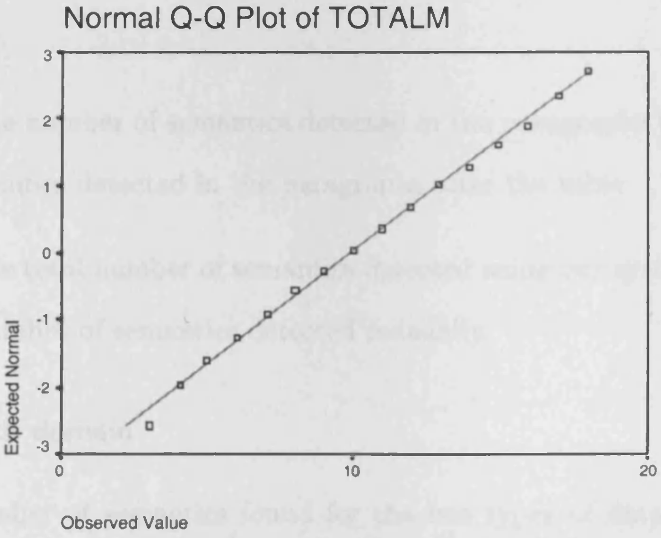


Figure 9.4: *Normal Q-Q Plot of semantics in documents*

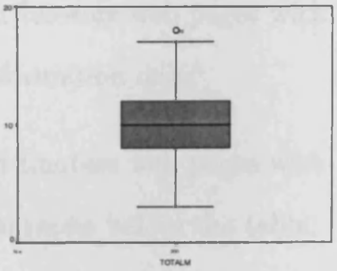


Figure 9.5: *Boxplot*

our assertion that " there is a significant number of useful semantics, hidden in a document, which are related to tabular data in that document" is shown to be sound.

After testing the normality of our sample, we use the experimental data to perform the tests described in the next sections.

9.2.2.1 For each data source

1. Compare the number of semantics detected using the domain ontology to enhance the search keys with the number detected using only keys from the table information.
2. Compare the number of semantics detected in the paragraphs before the table with the number detected in the paragraphs after the table.
3. Compare the total number of semantics detected using our system (SRD) with the total number of semantics detected manually.

9.2.2.2 For each domain

Compare the number of semantics found for the two types of data sources in the domain.

1. Number of semantics in Imotors web pages with number in Which web pages, detected using the table keys enhanced by terms from the domain ontology.

2. Number of semantics in Imotors web pages with number in Which web pages, detected using table information only.

3. Number of semantics in Imotors web pages with number in Which web pages, detected using the paragraphs before the table.

4. Number of semantics in Imotors web pages with number in Which web pages, detected using the paragraphs after the table.

5. Total number of semantics in Imotors web pages with total number in Which web pages.

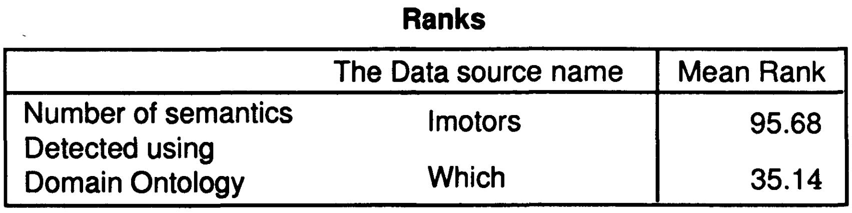
These tests will also be performed using the chemistry domain web sites when Thermoset web pages will be compared with the Eastman web pages. These tests will also be performed to compare the two sets of commercial web pages, and the two sets of scientific web pages, i.e. a comparison of the Imotors web pages with Thermoset web pages and the Which website with the Eastman website.

9.3 Experimental Analysis

In this section, we present the analysis of the results of the experiments, and interpret these results to explain how they prove our hypothesis.

9.3.1 Tests between the data sources for each domain

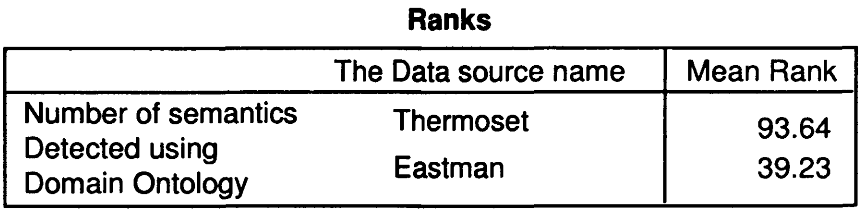
In this section, we test the relationship between the number of semantics detected in documents for a commercial website with the number of semantics detected for an evaluation website for each domain. We have performed five tests as follows:



Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Domain Ontology	Imotors	95.68
	Which	35.14

Figure 9.6: *Mean Ranks from Mann-Whitney test for the number of semantics detected using the domain ontology approach in the cars domain*



Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Domain Ontology	Thermoset	93.64
	Eastman	39.23

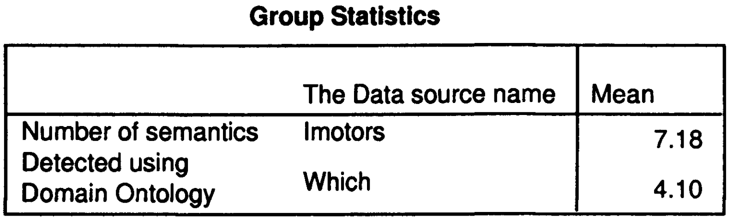
Figure 9.7: *Mean Ranks from Mann-Whitney test for the number of semantics detected using the domain ontology approach in chemistry domain*

9.3.1.1 Testing domain ontology approach

In this test we compare the number of semantics detected using the domain ontology approach in both data sources for each domain. We have performed two types of tests a Mann-Whitney test and t-test [BC98]. Both tests have given us positive results in that they show that there is a significant difference between the commercial data sources and the evaluation data sources for these results.

In the first test, see Figures 9.6 and 9.7, the mean rank of both commercial web sites, Imotors (95.68) and Thermoset (93.64), is higher than that of the evaluation web sites, Which (35.14) and Eastman (39.23), which shows that more semantics are detected for both domains (cars and chemistry) in the commercial documents.

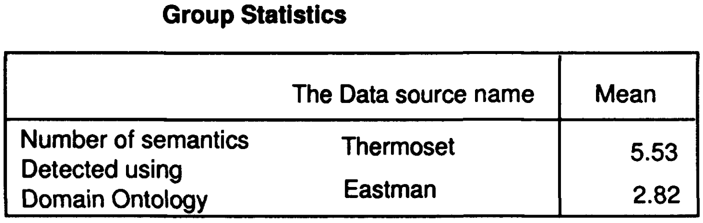
Thus, the Mann-Whitney test shows us that there is a significant difference between the commercial and the evaluation web sites, with respect to the number of semantics detected using our domain ontology approach.



Group Statistics

	The Data source name	Mean
Number of semantics Detected using Domain Ontology	Imotors	7.18
	Which	4.10

Figure 9.8: *Means from T-Test for the number of semantics detected using the domain ontology approach in cars domain*



Group Statistics

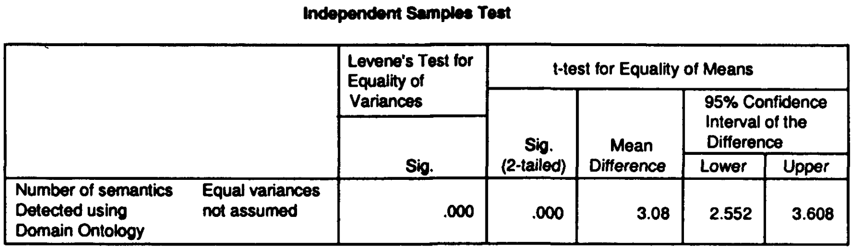
	The Data source name	Mean
Number of semantics Detected using Domain Ontology	Thermoset	5.53
	Eastman	2.82

Figure 9.9: *Means from T-Test for the number of semantics detected using the domain ontology approach in chemistry domain*

The second test, which is a t-test, has also shown that there is a significant difference between documents from commercial and evaluation websites. In Figures 9.8 and 9.9, the means of the commercial websites (7.18 and 5.53) are higher than the means of the evaluation websites (4.10 and 2.82) in the same domain.

In Figures 9.10 and 9.11, the 2-tailed t-test analysis shows us that there is a significant difference at the 5% level between the commercial websites and evaluation websites. Also the lower and upper values are at a '95 % confidence interval of the difference' not being zero, which also indicates that the difference is significant.

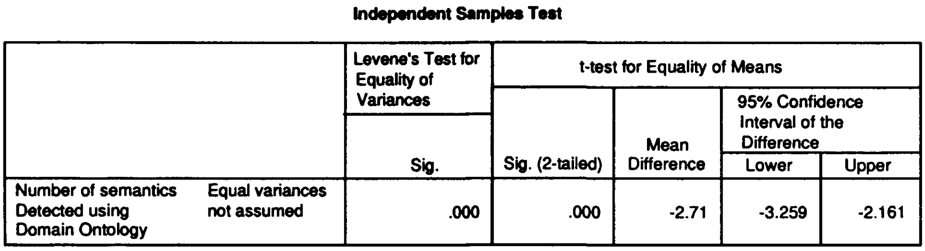
These tests have shown that the commercial web sites use, in building the text of their web sites, more semantic information related to table than the evaluation



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
		Sig.	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Number of semantics Detected using Domain Ontology	Equal variances not assumed	.000	.000	3.08	2.552	3.608

Figure 9.10: *Significance from T-Test for the number of semantics detected using the domain ontology approach between car domain data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
		Sig.	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Number of semantics Detected using Domain Ontology	Equal variances not assumed	.000	.000	-2.71	-3.259	-2.161

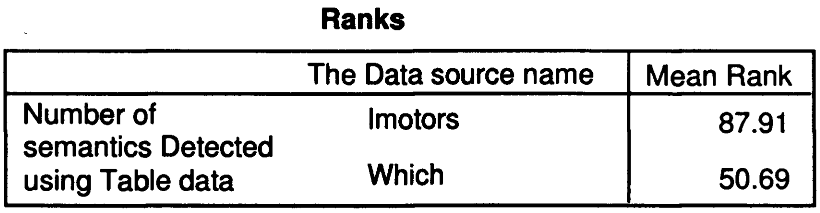
Figure 9.11: *Significance from T-Test for the number of semantics detected using the domain ontology approach between chemistry domain data sources*

web sites, and therefore using the domain ontology in detecting hidden semantics in the commercial web sites is more beneficial to the process of detecting hidden semantics.

9.3.1.2 Testing the table data approach

In this test, we are comparing the number of semantics detected using the table data approach in documents from commercial data sources with documents from evaluation data sources for the same domain. We would like to see if the results are significantly different for the two approaches.

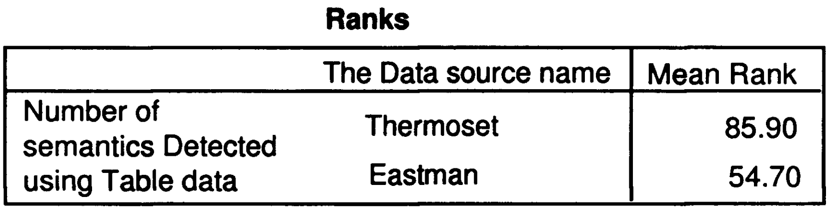
Figures 9.12 and 9.13 show that both of the commercial web sites have a higher mean rank (Mean rank lists the average of the ranks for each group), Imotors (87.91) and Thermoset (85.90) - than the evaluation web sites - Which (50.69) and Eastman



Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Table data	Imotors	87.91
	Which	50.69

Figure 9.12: *Mean Rank from Mann-Whitney test for the number of semantics detected using the table data approach in car domain*



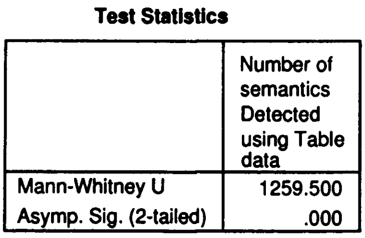
Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Table data	Thermoset	85.90
	Eastman	54.70

Figure 9.13: *Mean Rank from Mann-Whitney test for the number of semantics detected using the table data approach in chemistry domain*

(54.70). This means that the table data approach is giving us more semantics for the commercial data sources than the evaluation data sources.

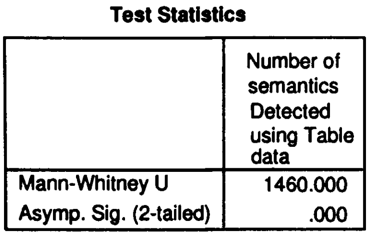
Using the Mann-Whitney test, Figures 9.14 and 9.15 show us that there is a significant difference between commercial and evaluation data sources with the values



Test Statistics

	Number of semantics Detected using Table data
Mann-Whitney U	1259.500
Asymp. Sig. (2-tailed)	.000

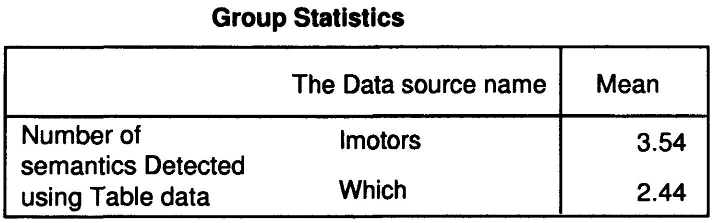
Figure 9.14: *Significance from Mann-Whitney test for the number of semantics detected using the table data approach between car domain data sources*



Test Statistics

	Number of semantics Detected using Table data
Mann-Whitney U	1460.000
Asymp. Sig. (2-tailed)	.000

Figure 9.15: *Significance from Mann-Whitney test for the number of semantics detected using the table data approach between chemistry domain data sources*



Group Statistics

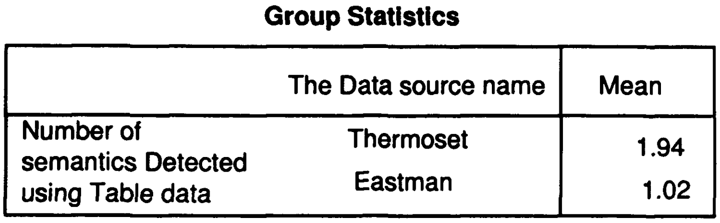
	The Data source name	Mean
Number of semantics Detected using Table data	Imotors	3.54
	Which	2.44

Figure 9.16: *Means from T-Test for the number of semantics detected using the table data approach in car domain*

of Asymp. Sig.(2-tailed) in the tables being < 0.05 which means that there is a high significance attached to the difference between these types of data sources. Using the independent two-valued t-test also shows that there is a significant difference between the means of the two types of data sources for each domain, Imotors web site has 3.54 and Which web site has 2.44. This means Imotors has 45 % more semantics than the Which web site. Also, for the chemistry domain, the commercial data source Thermoset has 1.94 semantics whereas the evaluation data source Eastman has 1.02 semantics (see Figures 9.16 and 9.17).

The independent samples t-tests in (Figures 9.18 and 9.19) show a significant difference between the commercial and evaluation data sources. This shows the difference is significant and gives us confidence in our results.

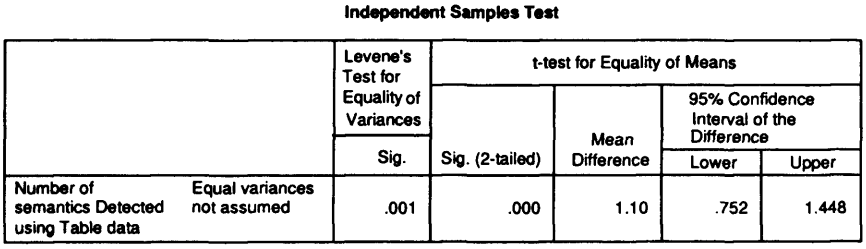
We can conclude from this test that the commercial data sources have more



Group Statistics

	The Data source name	Mean
Number of semantics Detected using Table data	Thermoset	1.94
	Eastman	1.02

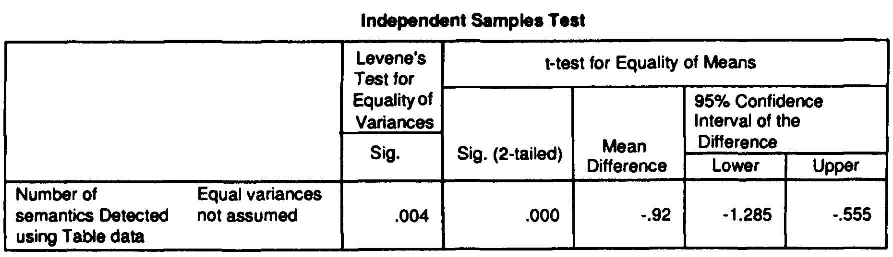
Figure 9.17: *Means from T-Test for the number of semantics detected using the table data approach in chemistry domain*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics Detected using Table data	Equal variances not assumed	.001	.000	1.10	.752	1.448

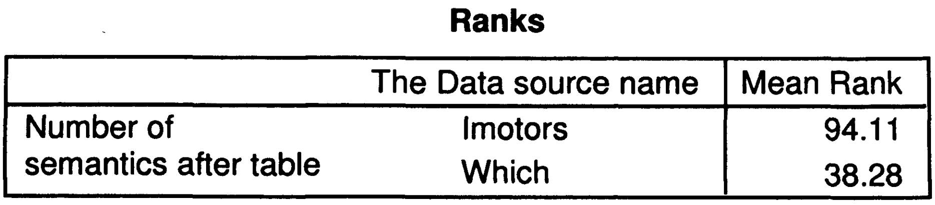
Figure 9.18: *Significance from T-Test for the number of semantics detected using the table data approach between car domain data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics Detected using Table data	Equal variances not assumed	.004	.000	-.92	-1.285	-.555

Figure 9.19: *Significance from T-Test for the number of semantics detected using the table data approach between chemistry domain data sources*



Ranks

	The Data source name	Mean Rank
Number of semantics after table	Imotors	94.11
	Which	38.28

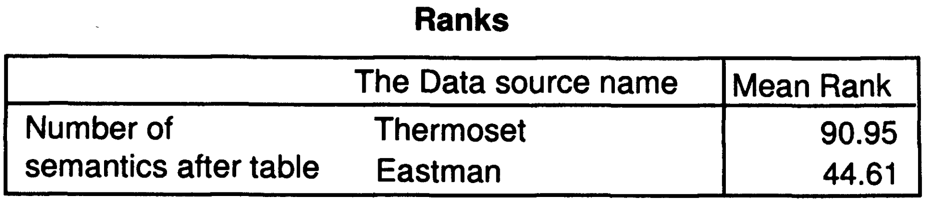
Figure 9.20: *Mean Ranks from Mann-Whitney test for the number of semantics detected after the table in the car domain*

semantics related to the table data in the text than the evaluation data sources, certainly in the domains we have investigated. We think that the reason for this is because the commercial web site designers tend to explain the content of their tables carefully and more fully to the consumers, whereas in evaluation data sources they only try to show the results of their evaluation and then explain these results in detail. Further investigation is needed to see if this difference is true in other domains.

9.3.1.3 Comparing semantics detected after the table

We are investigating here the occurrence of semantics in data sources in the paragraphs after the table. The results of this test show that the commercial data sources have a lot more semantics in these paragraphs than the evaluation data sources. In the cars domain, the Imotors web site has 70 % more semantic items in the paragraphs after the table than the Which web site, and similarly in the chemistry domain, the Thermoset web site documents have 92 % more semantic items than the Eastman web site in the paragraphs after the table.

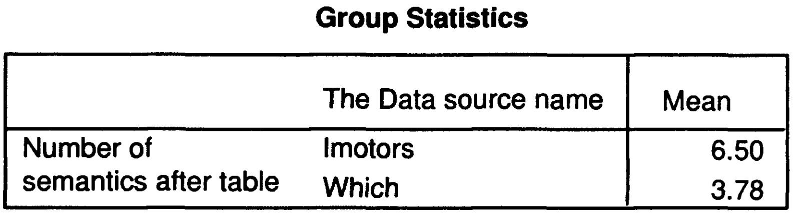
Figures 9.20 and 9.21 show the mean rank for semantics detected in paragraphs after the table of the commercial data sources is always higher than in evaluation data sources. Also, the mean of Imotors and Thermoset web sites is higher than the



Ranks

	The Data source name	Mean Rank
Number of semantics after table	Thermoset	90.95
	Eastman	44.61

Figure 9.21: *Mean Ranks from Mann-Whitney test for the number of semantics detected after the table in the chemistry domain*



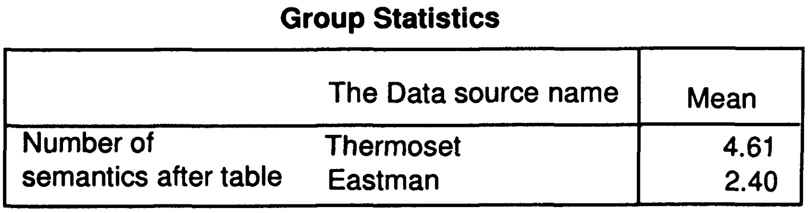
Group Statistics

	The Data source name	Mean
Number of semantics after table	Imotors	6.50
	Which	3.78

Figure 9.22: *Means from T-Test for the number of semantics detected after the table in the car domain*

mean of Which and Eastman web sites (see Figures 9.22 and 9.23) .

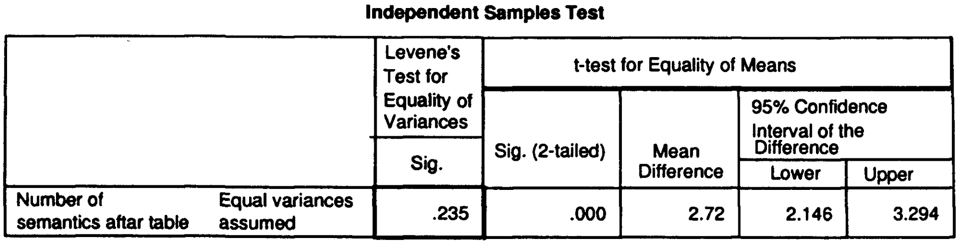
The T-test results (Figures 9.24 and 9.25) show that there is a significant difference between the numbers of semantics detected after the table in commercial and evaluation data sources. This means that commercial data sources concentrate on



Group Statistics

	The Data source name	Mean
Number of semantics after table	Thermoset	4.61
	Eastman	2.40

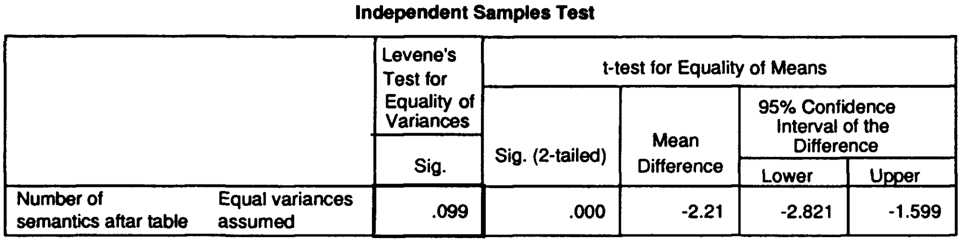
Figure 9.23: *Means from T-Test for the number of semantics detected after the table in the chemistry domain*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
		Sig.	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Number of semantics aftar table	Equal variances assumed	.235	.000	2.72	2.146	3.294

Figure 9.24: *Significance from T-Test for the number of semantics detected after the table between the car domain data sources*



Independent Samples Test

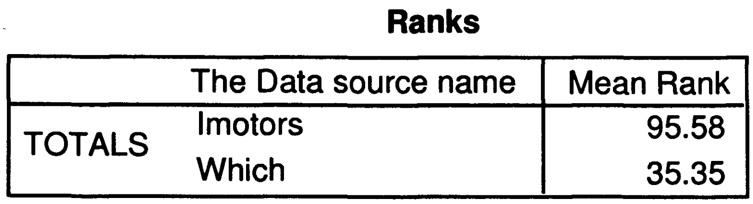
		Levene's Test for Equality of Variances	t-test for Equality of Means			
		Sig.	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Number of semantics aftar table	Equal variances assumed	.099	.000	-2.21	-2.821	-1.599

Figure 9.25: *Significance from T-Test for the number of semantics detected after the table between the chemistry domain data sources*

putting semantic information about a table in the paragraphs after the table in the documents to a greater extent than the evaluation data sources.

9.3.1.4 The total semantics detected

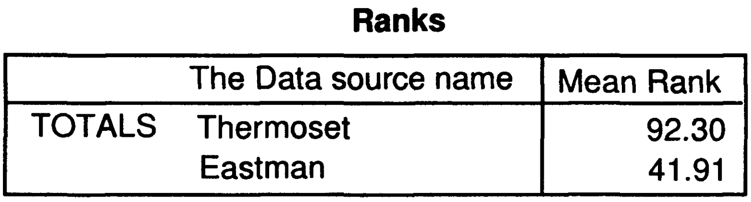
Here, we investigate whether the total number of semantics detected in each set of documents will have the same results as in the previous tests. The obvious answer is yes, and this is what the Mann-Whitney and t-test indicate. If we look at the mean ranks in Figure 9.26, Imotors has 95.58 which is higher than the mean rank for Which website (35.35). Also in Figure 9.27, Thermoset website has higher mean rank (92.30) than Eastman website (41.91). Figures 9.28 and 9.29 show that the means of both commercial data sources are higher than the means for evaluation data sources. The mean of Imotors web site is 9.59 whereas for the Which web site



Ranks

	The Data source name	Mean Rank
TOTALS	Imotors	95.58
	Which	35.35

Figure 9.26: *Mean Ranks from Mann-Whitney test for the total number of semantics detected in the document for the car domain*



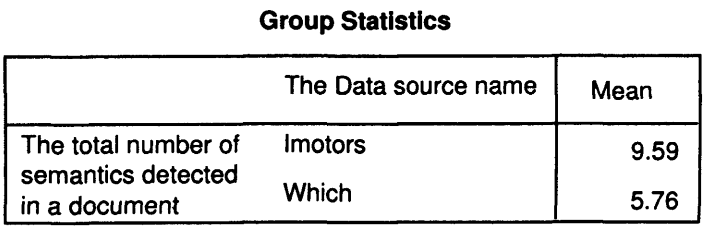
Ranks

	The Data source name	Mean Rank
TOTALS	Thermoset	92.30
	Eastman	41.91

Figure 9.27: *Mean Ranks from Mann-Whitney test for the total number of semantics detected in the document for the chemistry domain*

it is 5.76 and the mean of Thermoset web site is 6.29 whereas it is only 3.16 in Eastman.

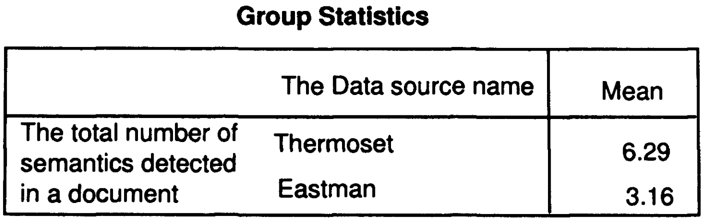
Both the tests show that the commercial data sources have more semantics than the evaluation data sources and the difference is significant (see Figures 9.30 and



Group Statistics

	The Data source name	Mean
The total number of semantics detected in a document	Imotors	9.59
	Which	5.76

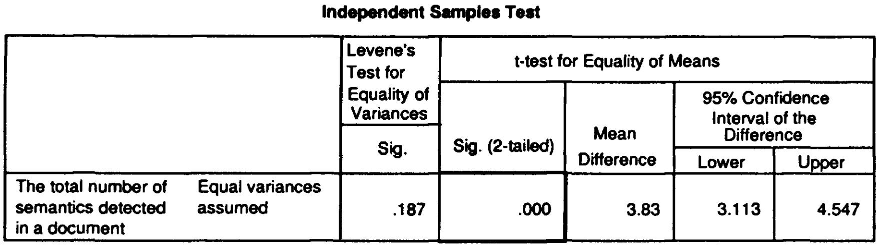
Figure 9.28: *Means from T-Test for the total number of semantics detected in the document for the car domain*



Group Statistics

	The Data source name	Mean
The total number of semantics detected in a document	Thermoset	6.29
	Eastman	3.16

Figure 9.29: *Means from T-Test for the total number of semantics detected in the document for the chemistry domain*



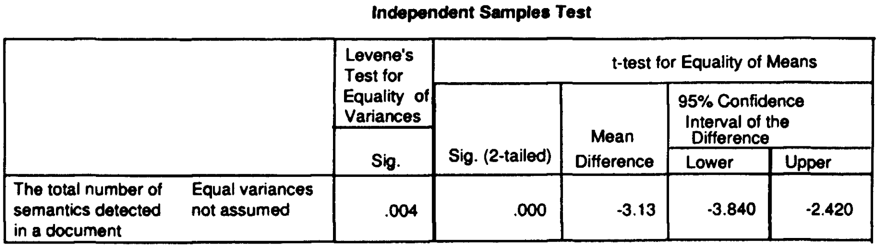
Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
The total number of semantics detected in a document	Equal variances assumed	.187	.000	3.83	3.113	4.547

Figure 9.30: *Significance from T-Test for the total number of semantics detected in the document between the car domain data sources*

9.31).

The ayalysis of the previous five tests, led to a conclusion that the text of commercial data sources have more semantics than the text of evaluation data sources in



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
The total number of semantics detected in a document	Equal variances not assumed	.004	.000	-3.13	-3.840	-2.420

Figure 9.31: *Significance from T-Test for the total number of semantics detected in the document between the chemistry domain data sources*

both domains used in the experiment. We believe that the large difference between these data sources is due to two main reasons:

1. In commercial data sources, they try to give the reader as much information as they can to attract him or her to buy their merchandise, whereas in evaluation data sources, they evaluate a product without considering the satisfaction of the reader, because they are showing facts which the reader needs to know when comparing items.

2. The commercial data sources have to attract the consumers to buy their products, therefore they put as much information as they can in a very short concise text. This makes it easier for the reader to identify the important concepts to them for comparison with other systems, while in evaluation data sources, the time constraint is not a crucial element with their readers who will be prepared to read more text as they want to find out more about the product.

9.3.2 Comparing the different domains

Is there any relation between the domain type and the number of semantics detected? Does a non-scientific domain have a larger number of semantics than a scientific one? Does this difference occur in both data sources? These questions and others concerned with comparing the two domains used are investigated in the tests reported in Appendix C. These results are not presented here as they are not part of the hypothesis demonstration. However they indicate that there could be a significant difference between the car domain (an example of a commercial area) and the chemistry domain (an example of a scientific domain). If time had allowed it would have been interesting to determine if this happened with other commercial and scientific domains.

9.4 Experimental Results

We have divided the significant results that we found into two parts, namely tests related to the logical content of the documents, and tests related to the physical structure of the documents.

9.4.1 Tests related to logical content of documents

We found that there is a significant number of useful semantics, hidden in a document, which are related to the tabular data in that document. After analysing 300 documents from different data sources, we found that there is hidden data related to the tabular data in the document which can be detected, extracted and represented as beneficial semantics related to that table. These semantics can be used for integrating these tables with data from other data sources.

Our experiment has shown that the number of detected semantics is normally distributed among all the 300 documents with a mean of 9.9 semantics per document. This means that the average number of semantics detected in a document is 9.9. The experiment sample documents can be categorised by the name of the domain that the sample is related to. Also, within each domain there are two types of data sources, commercial and evaluative.

We have used two domain types, scientific (chemistry) and non-scientific (cars). We found that the chemistry domain had fewer semantics than the cars domain, with the chemistry mean equal to 8.3 and the cars domain mean equal to 11.5. Thus, the cars domain has a higher number of semantics per document, and we think that the reasons for this are as follows:

1. The chemistry domain has well-defined concepts, therefore the users or the writers of the documents do not have many alternative terms to identify and describe the concepts. In other words the domain name concepts don't have

many synonyms or relationships and are well known to the community.

2. As a result of the first reason, the writers of a document don't have many things to explain in the text.

3. The cars domain has different terminologies between different countries and even between groups of people; therefore the writers of a document in this domain need to explain most of the attributes in their tables, as they are targeting a set of readers with a less coherent background.

We have shown in this thesis that detecting the hidden semantics will result in a better understanding of a table, and in its enrichment it with extra semantics.

Data source name	Domain name	Percentage
Imotors	Cars-Commercial	74.8
Thermoset	Chemistry-Commercial	87.9
Which	Cars-Evaluation	71
Eastman	Chemistry-Evaluation	89.2

Table 9.1: Percentage of semantics detected in documents using domain ontology

9.4.1.1 Significance between different domains

80.7 % of the detected semantics have been detected using a domain ontology approach. In our experiments, we used two approaches to detect the hidden semantics. One of these approaches used a domain ontology. This approach gives us most of the semantics that were detected manually in the text, as shown on Table 9.1. This table, shows the percentage of semantics detected by the domain ontology approach in each data source of the total number of semantics that were detected manually. Thus, most of the semantics are detected using a domain ontology approach. Therefore, any event that affects the domain ontology will affect the total number

Data source name	Domain name	number semantics
Imotors	Cars-Commercial	7.18
Thermoset	Chemistry-Commercial	5.53
Which	Cars-Evaluation	4.10
Eastman	Chemistry-Evaluation	2.82

Table 9.2: Number of semantics detected in each data source using domain ontology

of semantics detected, and any changes in the domain ontology will also affect the detection process. For instance, if the domain ontology becomes richer in concepts, then the total number of semantics that are detected will increase, and vice versa. This led us to conclude that a domain ontology plays an important role in detecting hidden semantics in documents, and any limitations in the domain ontology will also limit the number of semantics detected.

We also found that the number of semantics detected using a domain ontology approach in a non-scientific domain (cars) is higher than in the scientific domain (see table 9.2). But with respect to the percentage of the total number of semantics detected manually, the scientific domain has a higher percentage, as shown in Table 9.1.

9.4.1.2 Significance of the differences between data sources

There seems to be a significant difference between the number of semantics in the commercial and evaluation data sources in both our domains. Comparing the number of semantics detected in commercial data sources in the cars and chemistry domains has shown that they are always higher in the commercial data sources than in the evaluation data sources, see Table 9.3. However, this needs further investigation to determine if it holds in other domains.

We believe that the large difference between these types of data sources is due to a number of reasons:

Data source name	Domain name	Data source type	number semantics
Imotors	Cars	Commercial	9.59
Thermoset	Chemistry	Commercial	6.29
Which	Cars	Evaluation	5.76
Eastman	Chemistry	Evaluation	3.16

Table 9.3: number semantics for each data source

1. Commercial data sources try to give the reader as much information as they can to attract him or her to buy their merchandise, whereas in evaluation data source documents they evaluate a product without looking to the need to attract the reader to purchase. This is in part, because they are showing facts which the supplier and the reader might not like.

2. Commercial data sources have to attract the consumers to buy their products using a small space. Therefore, they put all the information they have into a very short text which makes it a quicker to read and also easier to detect by other systems, whereas in evaluation data sources, the time constraint is not a crucial element as their readers want as full a comparison as possible.

3. The length of a document has an effect on the number of semantics. The evaluation data sources are longer documents than commercial data sources, yet have less table semantics. This needs more investigation, but may indicate they are giving a fuller evaluation in the text of the points being made.

4. Also, most of the documents in the evaluation data sources describe new products and technologies which have not yet been included in the domain ontology. This may be a cause of the smaller number of semantics detected in them, but this needs further investigation

9.4.1.3 Significance in the prototype system

Our system SRD has detected 70 % of the total number of semantics detected manually in the sample documents. By comparing the total number of semantics detected by SRD with the total number of semantics detected manually, we found that for all data sources SRD detected a reasonable percentage of the total semantics, (see Table 9.4). This Table shows a significant difference between the number of semantics detected by SRD and manually, and we believe that the reason for this is not the method we are using, but two other reasons:

Data source	Domain name	SRD	Manually	Percentage
Imotors	Cars	9.59	12.38	77.5
Thermoset	Chemistry	6.29	9.09	70
Which	Cars	5.76	9.76	59.1
Eastman	Chemistry	3.16	6.8	46.5

Table 9.4: Number of semantics detected by SRD and Manually

1. There is a weakness in the programming and thus the keyword searching in our system is not as sophisticated as it might be. If we had used a better searching mechanism, we might have achieved better results. For example, when searching for engine size the system is able to detect " engine size " but not able to detect "engines-size" in the text.

2. The domain ontology used is limited. For example, in one of the documents there was a semantic value in the text mentioning that 'the car has 260 HPs' but 'HP' was not a term in the domain ontology, and therefore the SRD system did not detect this semantic.

However, there is a high correlation (0.875) between the two variables, see Table 9.5 and Figure 9.32. The other values have a similar behavior, which tells us that

Sig. (2-tailed)	Number of documents	Correlation
.000	300	.875

Table 9.5: Correlation between the number of semantics detected by SRD and Manually

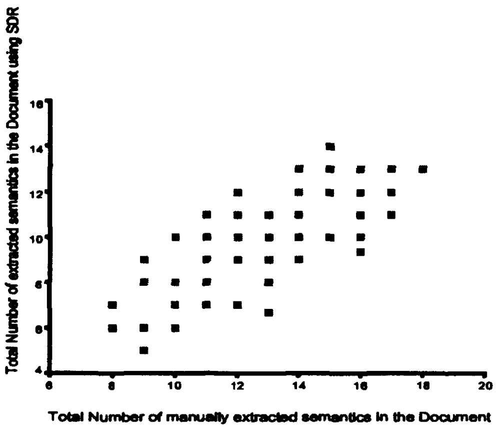


Figure 9.32: *An example of enhanced table metadata*

our system is detecting a reasonable percentage 77.5 % of the semantics in Imotors, 59.1% in Which , 70% in Thermoset, and 46.5% in Eastman (see Table 9.4) and that this aspect allows room for future improvements.

The high percentage of undetected semantics by the SRD when compared with the manually detected semantics has occurred because of the weakness of the domain ontology used in our system. We found that most of the missing semantics could be related by an ontology if the ontology was expanded. However, without a well defined ontology the missing semantics will remain undetected. We believe that this point needs more investigation using a richer ontology. However, it shows that an automatic detection system using an ontology is unlikely to achieve 100% due to the difficulty of getting a comprehensive ontology.

Data source name	Domain name	Percentage
Imotors	Cars-Commercial	88.3
Thermoset	Chemistry-Commercial	86.1
Which	Cars-Evaluation	64.8
Eastman	Chemistry-Evaluation	61

Table 9.6: Percentage of semantics in the adjacent paragraphs

9.4.2 Experimental significance related to the physical structure of documents

9.4.2.1 Adjacent paragraphs

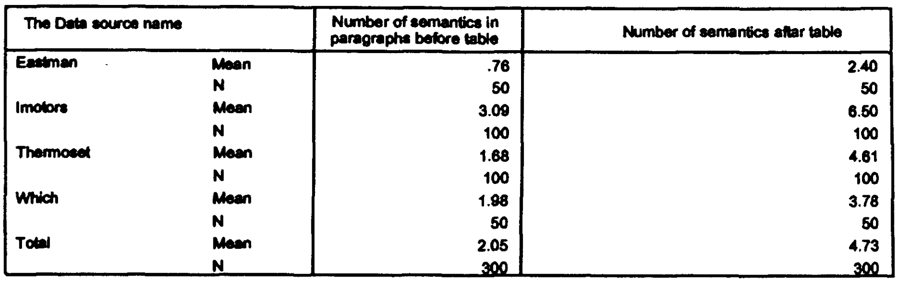
We found that 75.05 % of the total detected semantics come from the adjacent paragraphs to the table. In Table 9.6, the commercial data sources from both domains have a very high percentage of their semantics coming from the adjacent paragraphs, while in the evaluation data sources the percentages are not so high.

We believe there are three reasons for this:

1. The commercial data sources are reasonably short compared to the evaluation documents, and this will affect the spread of the semantics in the paragraphs. In commercial data sources, documents have 2 to 7 paragraphs, whereas in evaluation data sources they can go up to 13 paragraphs.

2. Evaluation data sources tend to talk about one concept or part of the table in each paragraph and start on a new topic in a new paragraph. This leads to the semantics being spread over the document.

3. Commercial data sources try to concentrate the information into one or two paragraphs to hold the reader's attention.



The Data source name		Number of semantics in paragraphs before table	Number of semantics aftar table
Eastman	Mean	.76	2.40
	N	50	50
Imotors	Mean	3.09	6.50
	N	100	100
Thermoset	Mean	1.68	4.61
	N	100	100
Which	Mean	1.98	3.78
	N	50	50
Total	Mean	2.05	4.73
	N	300	300

Figure 9.33: *An example of enhanced table metadata*

9.4.2.2 Paragraphs under the table

We found that 70 % of the detected semantics appear to be from the paragraphs after the table. It is common that the writer of the document will describe the table after showing it to the reader. In our sample, the table in commercial data sources usually comes in the middle of the document, whereas in evaluation data sources it is normally near the beginning of the documents.

In Figure 9.33, we can see that the mean of the number of semantics after a table is always higher than the mean of the number of semantics before the table for all data sources. In Imotors, the paragraphs after the table have produced 65% of the total semantics detected in the documents, also in Which they have produced 66%, Thermoset 74%, and in Eastman they have produced 76%. In total the paragraphs after the table give us 68% of the total number of semantics. This tells us that detecting semantics in the paragraphs after the table is more productive than using the paragraphs before the table.

We believe that by concentrating on the adjacent paragraphs and paragraphs under the table, a system will get most of the semantics, if not all of them presented in a document. In some cases, the tables are put at the end of the document, or in a certain place in the text, or refer to it by its number. We treat this type of document as if the table is in the first indicator position in the document text.

Data source	P with I	P with I and S	S in P containing I.	S in the 1st P
Imotors	80	100	89.5	51.7
Thermoset	83.3	100	92	38.2
Which	90.9	95	94.7	10
Eastman	88	90	98.2	13.6

Table 9.7: Percentage of semantics in paragraphs that contain indicators

9.4.2.3 Semantics and indicators

By tracing the types of indicators used in the text, we find that there are a number of types that the writers tend to use to point to the table from the text. The first one, and the most commonly used is indicating a table by its number, for example 'Table 2.1', and this appears when there is more than one table in the document. It is also used when the table is far away from the indicator in the document. The other types of indicators are 'the above table', 'the next table', 'the last table', 'the previous table', 'in the table below' and the use of 'Figure' instead of 'Table' in these phrases. These types of indicator always need to be close to the table. In some cases there is no indicator in the documents, and this is because the document is short and there was only one table.

In Table 9.7 P is Paragraph, S is Semantics, and I is Indicators. The column heading P with I and S means paragraphs with an indicator and semantics. This analysis shows that paragraphs containing indicators nearly always have semantics in them (Column 3) and that the semantics are always next to the indicator. Comparing between paragraphs, the paragraphs that have indicators have most of the detected semantics in them (Column 4). Therefore, in large documents, it is useful to search for the indicators first and concentrate on the paragraphs that contain them when looking for hidden semantics.

9.4.2.4 Paragraphs and indicators

Among those paragraphs that have indicators, the first paragraph has the highest number of semantics in it. We believe that this result needs more investigation, because the type of documents we used might affect this result. The documents we used are slightly short and might not show the real situation. However, in the evaluation data sources, the writers spread the semantics throughout the documents and sometimes they use different indictors, for example 'in the first column' and 'in the last row' appear in these documents showing a fuller analysis is being undertaken.

9.5 Summary

We have presented in this chapter the experimental design and the objectives of this experiment. We have also discussed the different types of test carried out in the experiments. After applying the SRD system to the text data and gathering the results of these tests, we have analysed the results using statistical analysis methods such as, t-test and Mann-Whitney test to detecrmine the significance of our results. At the end of the chapter we have drawn conclusions about the significance of our experiment and the results supporting our hypothesis.

CHAPTER 10

Conclusion and Future Work

10.1 Introduction

10.2 Conclusion

There are number of semantics related to the table in the surrounding text. These related semantics can be detected, extracted, and represented into a suitable format. These semantics facilitate overcoming heterogeneity between different tables from different documents. We have shown that any integration of these types of tables will not be as complete unless the tables metadata description, which is hidden in the surrounding text, is combined with the tables' metadata.

The related semantics can be detected using two approaches. The first approach is using the table header of table metadata to search the text. We extracted the metadata of the table and searched for the extracted words in the text. The second approach is to augment the table metadata with the corresponding related concepts, synonyms and relations from a domain ontology. The approach of using a domain ontology to enrich the searching mechanism, by experiment, we found detected more semantics than the first approach.

The extracted semantics ware then presented in two formats as semantic units and enhanced metadata. Semantic units are stand-alone units that represent the semantics with their corresponding concepts from the ontology with the value that a semantic represents. On the other hand, enhanced metadata is represented as the detected semantics, the corresponding concepts, and the related metadata associated with the table.
To ensure the effectiveness of linking these types of tables with the related semantics from the text, we have been able to define a number of different conversion functions that are able to convert the representation of the detected semantics into a suitable format for both types of semantics representation.

As an experiment, we applied our system SRD to 300 documents related to two domains, cars and chemistry. For each domain, we used documents from two different data sources, a commercial and an evaluation. The experiment has shown that documents from commercial data sources in doth domains have more semantics than those from evaluation sources. Also, documents from the scientific domain, chemistry, have fewer semantics than documents from the non-scientific domain, cars.
We found from the experiments that it is common that the writer of the document will describe the table after showing it to the reader. We found that 75.05 % of the total detected semantics comes from the paragraphs adjacent to the table. Also, we found that 71.8 % of the detected semantics appear to be from the paragraphs after the table. We believe that by concentrating on the adjacent paragraphs and paragraphs after the table it is possible to get most of the semantics, if not all of them, that are present in the text.

We also found that paragraphs that have indicators always have semantics in them, which are always next to the indicator. Thus, when comparing between paragraphs, the paragraphs with indicators have most of the detected semantics. Therefore, in large documents, it may be useful to search for the indicators first and concentrate on the paragraphs that contain them in the full detection process. By tracing the types of indicators, we found that there are a number of ways that the writers tend to use to indicate to a table in the text. The first one and the most commonly used is indicating to the table by its number for example 'Table 2.1'. The other types of indicators are 'the above table', 'the next table', 'the last table', 'the previous table', 'in the table below' and with 'figure' instead of table. This can be useful in directing the searching mechanism to concentrate on paragraphs that contain these types of indicators. Also, in large paragraphs, the hidden semantic is sometimes close to that indicator.

10.3 Future Work

10.3.1 Using large documents in experiment

In our experiments we have used documents that are relatively short (one page), and each document has only one table. Further work could use large documents which contain more than one table in the experiment and comparing the number of semantics detected with our results, and see if the size of documents affects the semantics detected and where they occur would be worth investigating. Also, analysing documents with more than one table will give the possibility of investigating how to distinguish the related semantics for each table without mixing the semantics.

10.3.2 Indicators and paragraphs

One of the areas that needs more investigation is the relation between indicators and the appearance of semantics in the same paragraph. As mentioned in Section 9.4.2.3, there is a strong relation between the number of semantics in one paragraph and the indicators in that paragraph. This is true in short documents, and it would be useful to investigate in large documents whether they have the same results.

10.3.3 Using other domains

In the real world, the number of concepts and semantics varies between different domains. Some domains are very rich areas which are full of synonyms, homonyms and relations between their concepts, and some domains represent areas which are narrow or poor in semantics. These differences between domains might affect the number of semantics that can be detected in documents and how they are found. As mentioned in the Section 9.4.1, there is a significant difference between the scientific and non-scientific domains. It would be useful to use different domains and compare them with the domains we have investigated, to measure the differences between these different domains. Also, we found that commercial web sites always have the highest number of semantics in both domains, and it would be useful to see if this relation is true in other domains.

10.3.3.1 Using other data sources

For each domain, we have used data from different data sources. We used data from a commercial data source and from an evaluation data source. In further work, the system could be applied to other data sources and differences with the results we had, measured to validate that the results can be applied to all different data sources would be useful future work. Also, in both data sources that have been used, the non-scientific domain has the highest number of semantics. It would be useful to see if the non-scientific domain will have more semantics than the scientific domain, in other data sources domains. Our work is encouraging in that the two domains are discreet and the results are very similar. However further work is needed to determine the generality of these results.

10.3.4 Word stemming

As presented in Section 2.3.2.1, we have used a simple search mechanism and word enrichment using ontology in searching the documents. Stemming is another mechanism that can be used. Measuring the benefits that stemming might give in detecting semantics from the documents would be a useful future investigation which might enhance the detection process. It is important, also, to measure the effort that stemming might take, and compare it with benefit it gives because stemming is usually resource and effort consuming. In this analysis we should remember that our simple approaches appear to be successful in locating a high proportion of the semantics presented in the text.

APPENDIX A

Car domain ontology

```
<!-- Content-type:text/plain;charset=iso-8859-1 -->
<?xml version = '1.0' standalone='no'?>
<!-- DAML is RDF -->
<rdf:RDF
  xmlns:rdf ='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:daml='http://www.daml.org/2001/03/daml+oil#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
>
<daml:Ontology rdf:about="">
  <rdfs:comment> An Ontology for Cars </rdfs:comment>
</daml:Ontology>
<!-- **** Currency ****  -->
  <daml:Class rdf:ID = 'Currency' >
    <daml:subClassOf rdf:resource = '#Price'/>
    <daml:label>Currency</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Type'>
    <daml:label>Type</daml:label>
    <daml:domain rdf:resource= '#Currency'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:DataTypeProperty rdf:ID = 'Country'>
    <daml:label>Country</daml:label>
    <daml:domain rdf:resource= '#Currency'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
```

```
<!-- **** VAT **** -->
  <daml:Class rdf:ID = 'VAT' >
    <daml:subClassOf rdf:resource = '#Price'/>
    <daml:label>VAT</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Type'>
    <daml:label>Type</daml:label>
  <comment>listOf{oneOf {Included, Excluded}} </comment>
    <daml:domain rdf:resource= '#VAT'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>

<!-- **** Price **** -->
  <daml:Class rdf:ID = 'Price' >
    <daml:subClassOf>
     <daml:Restriction>
       <daml:onProperty rdf:resource='#Price-of'/>
       <daml:toClass rdf:resource='#Car'/>
     </daml:Restriction>
    </daml:subClassOf>
    <daml:label>Price</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Amount'>
    <daml:label>Amount</daml:label>
    <daml:domain rdf:resource= '#Price'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:ObjectProperty rdf:ID = 'Price-of'>
    <daml:label>Price-of</daml:label>
    <daml:minCardinality>1</daml:minCardinality>
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:ObjectProperty>



<!-- **** Engine-size **** -->
  <daml:Class rdf:ID = 'Engine-size' >
    <daml:subClassOf rdf:resource = '#Engine'/>
    <daml:label>Engine-size</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Value'>
    <daml:label>Value</daml:label>
    <daml:domain rdf:resource= '#Engin-size'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:DataTypeProperty rdf:ID = 'type'>
    <daml:label>Type</daml:label>
    <comment>listOf{oneOf { CC , Liters }} </comment>
    <daml:domain rdf:resource= '#Engine'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
```

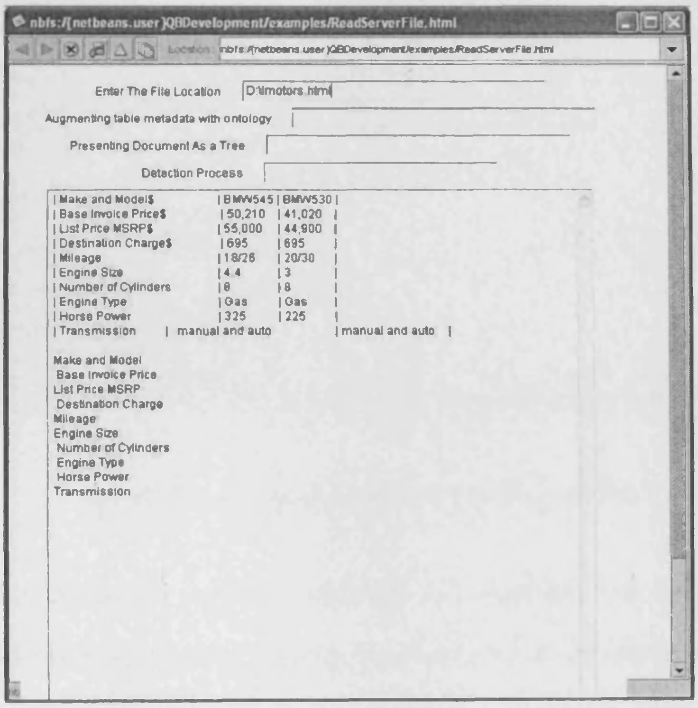
```
<!-- **** Engine ****  -->
  <daml:Class rdf:ID = 'Engine' >
    <daml:subClassOf>
      <daml:Restriction>
        <daml:onProperty rdf:resource='#Engine-of'/>
        <daml:toClass rdf:resource='#Car'/>
      </daml:Restriction>
    </daml:subClassOf>
    <daml:label>Engine</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Shape'>
    <daml:label>Shape</daml:label>
    <daml:domain rdf:resource= '#Engine'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:ObjectProperty rdf:ID = 'Engine-of'>
    <daml:label>Engine-of</daml:label>
    <daml:minCardinality>1</daml:minCardinality>
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:ObjectProperty>


<!-- **** Millage ****  -->
  <daml:Class rdf:ID = 'Mileage' >
    <daml:subClassOf>
      <daml:Restriction>
        <daml:onProperty rdf:resource='#Mileage-of'/>
        <daml:toClass rdf:resource='#Car'/>
      </daml:Restriction>
    </daml:subClassOf>
    <daml:label>Mileage</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Mejerment-type'>
    <daml:label>Magerment-type</daml:label>
    <comment>listOf{oneOf {Miles, Kiloliters}} </comment>
    <daml:domain rdf:resource= '#Mileage'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:DataTypeProperty rdf:ID = 'Number'>
    <daml:label>Number</daml:label>
    <daml:domain rdf:resource= '#Mileage'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:ObjectProperty rdf:ID = 'Mileage-of'>
    <daml:label>Mileage-of</daml:label>
    <daml:minCardinality>1</daml:minCardinality>
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:ObjectProperty>
```

```
<!-- **** Car ****  -->
  <daml:Class rdf:ID = 'Car' >
    <daml:subClassOf>
     <daml:Restriction>
       <daml:onProperty rdf:resource='#Has-price'/>
       <daml:toClass rdf:resource='#Price'/>
     </daml:Restriction>
    </daml:subClassOf>
    <daml:subClassOf>
     <daml:Restriction>
       <daml:onProperty rdf:resource='#Has-engine'/>
       <daml:toClass rdf:resource='#Engine'/>
     </daml:Restriction>
    </daml:subClassOf>
    <daml:subClassOf>
     <daml:Restriction>
       <daml:onProperty rdf:resource='#Has-mileage'/>
       <daml:toClass rdf:resource='#Mileage'/>
     </daml:Restriction>
    </daml:subClassOf>
    <daml:label>Car</daml:label>
  </daml:Class>
  <daml:DataTypeProperty rdf:ID = 'Type'>
    <daml:label>Type</daml:label>
    <daml:domain rdf:resource= '#Car'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:DataTypeProperty rdf:ID = 'Make'>
    <daml:label>Make</daml:label>
    <daml:domain rdf:resource= '#Car'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:DataTypeProperty rdf:ID = ' Model '>
    <daml:label> Model </daml:label>
    <daml:domain rdf:resource= '#Car'/>
    <daml:range  rdf:resource= 'String'/>
  </daml:DataTypeProperty>
  <daml:ObjectProperty rdf:ID = 'Has-price'>
    <daml:label>Has-price</daml:label>
    <daml:minCardinality>1</daml:minCardinality>
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:ObjectProperty>
  <daml:ObjectProperty rdf:ID = 'Has-engine'>
    <daml:label>Has-engine</daml:label>
    <daml:minCardinality>1</daml:minCardinality>
    <daml:maxCardinality>n</daml:maxCardinality>
  </daml:ObjectProperty>
  <daml:ObjectProperty rdf:ID = 'Has-mileage'>
    <daml:label>Has-mileage</daml:label>
    <daml:minCardinality>1</daml:minCardinality>
    <daml:maxCardinality>1</daml:maxCardinality>
  </daml:ObjectProperty>
  <daml :Class rdf:ID=" Cost ">
    <daml : sameClassAs rdf: resource =" # Price ">
  </daml :Class>
  <daml :Class rdf:ID=" Motor ">
    <daml : sameClassAs rdf: resource =" # Engine ">
  </daml :Class>
</rdf:RDF>
```

APPENDIX B

System Tutorial



nbfs:/[netbeans.user]QBDevelopment/examples/ReadServerFile.html

Location : nbfs:/[netbeans.user]QBDevelopment/examples/ReadServerFile.html

Enter The File Location: D:\imotors.html

Augmenting table metadata with ontology

Presenting Document As a Tree

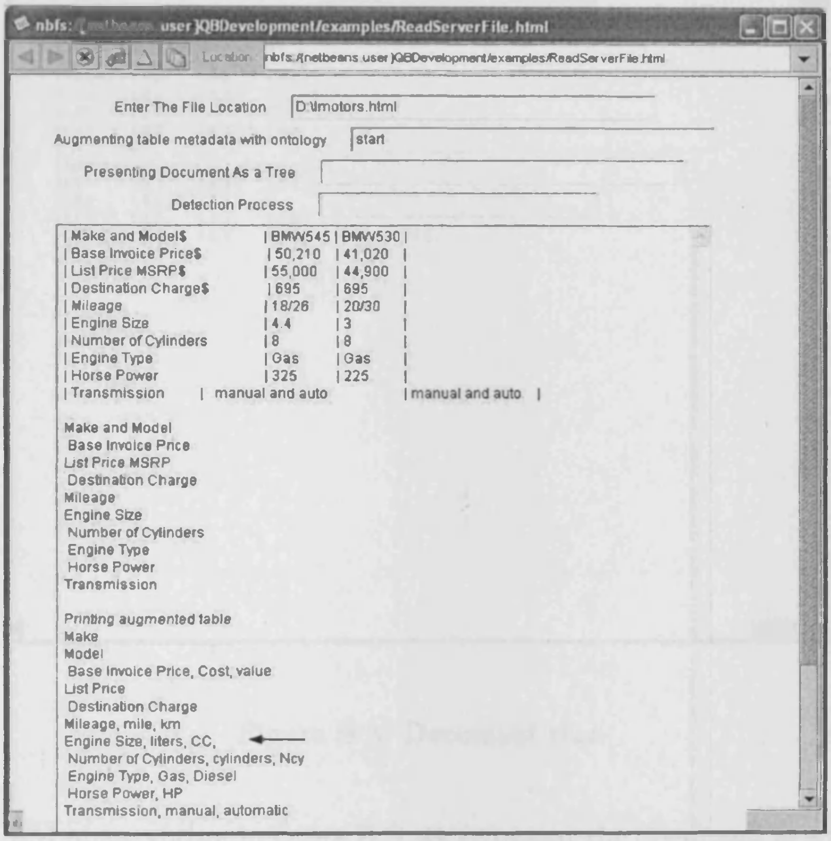
Detection Process

```
| Make and Model$          | BMW545 | BMW530 |
| Base Invoice Price$       | 50,210  | 41,020  |
| List Price MSRP$         | 55,000  | 44,900  |
| Destination Charge$       | 695     | 695     |
| Mileage                  | 18/26   | 20/30   |
| Engine Size              | 4.4     | 3       |
| Number of Cylinders      | 8       | 8       |
| Engine Type              | Gas     | Gas     |
| Horse Power              | 325     | 225     |
| Transmission        |  manual and auto           | manual and auto   |

Make and Model
 Base Invoice Price
List Price MSRP
 Destination Charge
Mileage
Engine Size
 Number of Cylinders
 Engine Type
 Horse Power
Transmission
```

Figure B.1: Software interface

Our system consist of four stages as shown in Figure B.1 . At the beginning we feed the system with a document from Imotors Web site. This document contains tabular data in it and a number of paragraphs. We assume that the location and the structure of the table is already known, therefore we feed the system with the table data. The system then extracts the table metadata and saves it in a table called SearchingTable.


nbfs://netbeans.user)QBDevelopment/examples/ReadServerFile.html
Location: nbfs://netbeans.user)QBDevelopment/examples/ReadServerFile.html

Enter The File Location D:\motors.html

Augmenting table metadata with ontology start

Presenting Document As a Tree

Detection Process

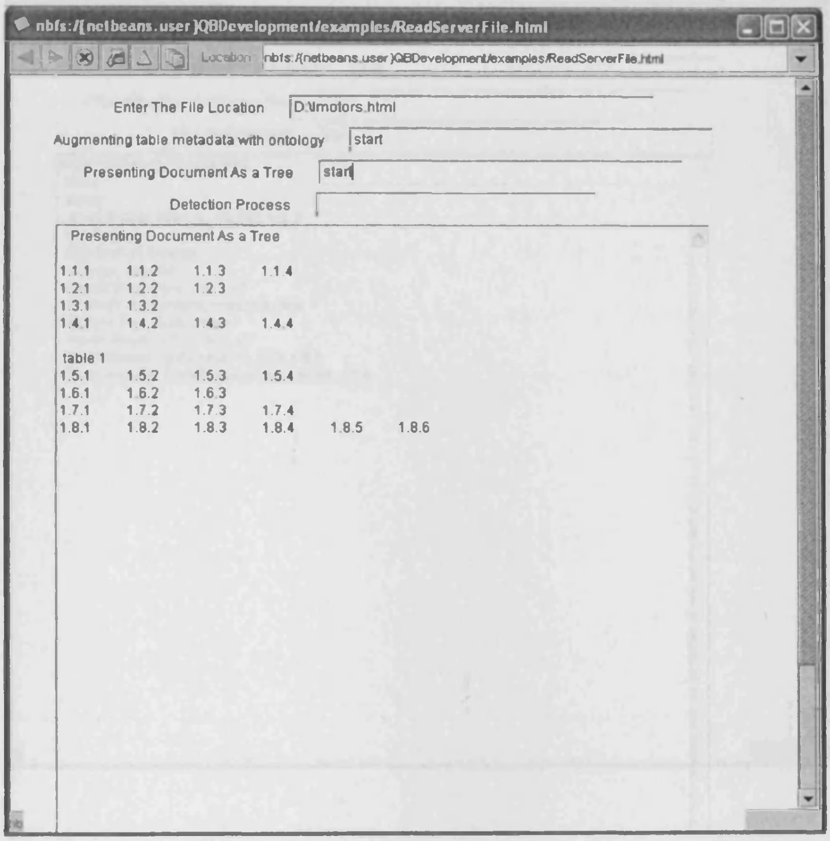
```
| Make and Model$            | BMW545 | BMW530 |
| Base Invoice Price$         | 50,210  | 41,020  |
| List Price MSRP$           | 55,000  | 44,900  |
| Destination Charge$         | 695     | 695     |
| Mileage                    | 18/26   | 20/30   |
| Engine Size                | 4.4     | 3       |
| Number of Cylinders        | 8       | 8       |
| Engine Type                | Gas     | Gas     |
| Horse Power                | 325     | 225     |
| Transmission       |  manual and auto           | manual and auto  |

Make and Model
 Base Invoice Price
List Price MSRP
 Destination Charge
Mileage
Engine Size
 Number of Cylinders
 Engine Type
 Horse Power
Transmission

Printing augmented table
Make
Model
 Base Invoice Price, Cost, value
List Price
 Destination Charge
Mileage, mile, km
Engine Size, liters, CC,
 Number of Cylinders, cylinders, Ncy
 Engine Type, Gas, Diesel
 Horse Power, HP
Transmission, manual, automatic
```

Figure B.2: Augmenting the table metadata

In the second phase, we use an ontology to augment the table metadata and to enrich the search keywords by adding the concepts from the ontology which are related to the table metadata. If a matching word is found in the ontology, then all synonyms, relations, and properties of that concept will be extracted and added to the SearchingTable. We can see that in figure B.2, the metadata 'Engine-size' has been enriched with its representations 'liters' and 'cc'.


nbfs:/[netbeans.user]QBDevelopment/examples/ReadServerFile.html
Location
nbfs:/(netbeans.user)QBDevelopment/examples/ReadServerFile.html

Enter The File Location: D:\motors.html

Augmenting table metadata with ontology: start

Presenting Document As a Tree: start

Detection Process:

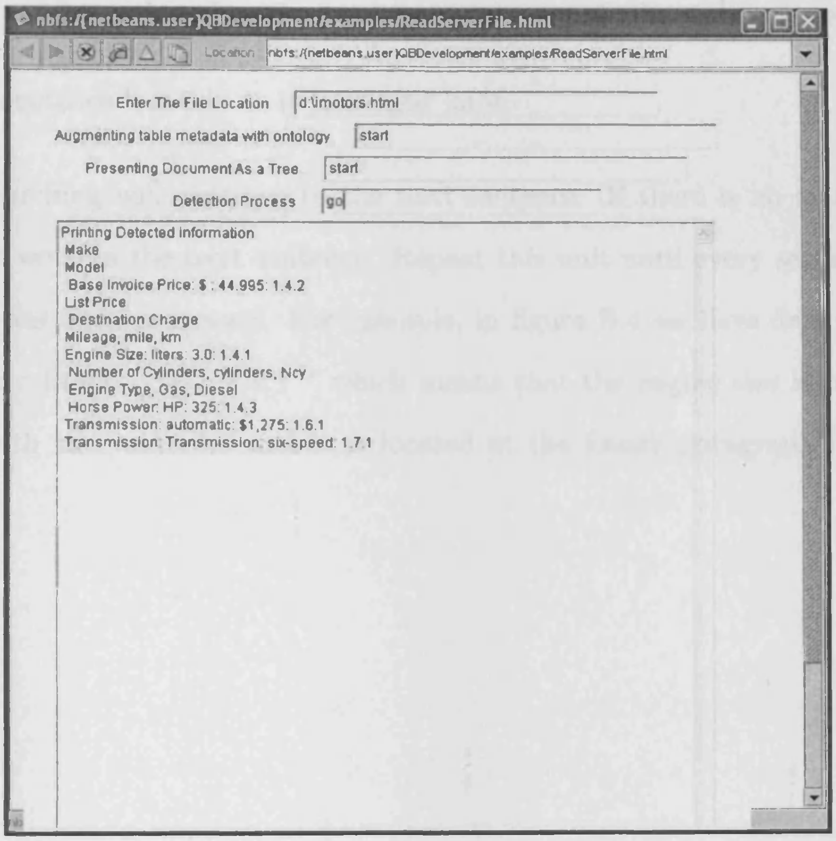
```
Presenting Document As a Tree

1.1.1     1.1.2     1.1.3     1.1.4
1.2.1     1.2.2     1.2.3
1.3.1     1.3.2
1.4.1     1.4.2     1.4.3     1.4.4

table 1
1.5.1     1.5.2     1.5.3     1.5.4
1.6.1     1.6.2     1.6.3
1.7.1     1.7.2     1.7.3     1.7.4
1.8.1     1.8.2     1.8.3     1.8.4     1.8.5     1.8.6
```

Figure B.3: Document tree

In the third phase shown in figure B.3, we represent the document in a tree format as it makes it easier to search the content of that document. As shown in figure B.3, we start the tree root with the document number (1.X.X.X). This root has a number of branches which represent paragraphs in that document (e.g. 1.2.X.X, 1.3.X.X, and 1.4.X.X). The third level represents the sentences in each paragraphs (e.g. 1.1.1, 1.3.1, 1.7.2, ect). In some cases, the table is located at the top or at the bottom of the document and in this case we assume that the first place that the table is indicated in the text is its location in the text. For example, the table in this document is indicated at the end of paragraph four, therefore we assume that this is the table location.



nbfs://{netbeans.user}QBDevelopment/examples/ReadServerFile.html
Location
nbfs://{netbeans.user}QBDevelopment/examples/ReadServerFile.html

Enter The File Location d:\\motors.html

Augmenting table metadata with ontology start

Presenting Document As a Tree start

Detection Process go

```
Printing Detected information
 Make
 Model
  Base Invoice Price: $ : 44.995: 1.4.2
 List Price
  Destination Charge
 Mileage, mile, km
 Engine Size: liters: 3.0: 1.4.1
  Number of Cylinders, cylinders, Ncy
  Engine Type, Gas, Diesel
  Horse Power: HP: 325: 1.4.3
 Transmission: automatic: $1,275: 1.6.1
 Transmission: Transmission: six-speed: 1.7.1
```

Figure B.4: Detection process

In the fourth phase, we search the document tree for words equivalent to the words held in the SearchingTable. We start this phase with the tree leaves (words) in the first sentence and first paragraph, we compare it with the SearchingTable. If there is a matching word, we then begin searching the sentence for mathematical symbols (=, <, >, etc.) or (is, are, equal, etc.). If there is such a symbol located next to the matching word, then we extract the word with the next word or value after the mathematical symbol and store it as:

- The original word from the search table with the related concept from the domain ontology.
- The matching word in the sentence.

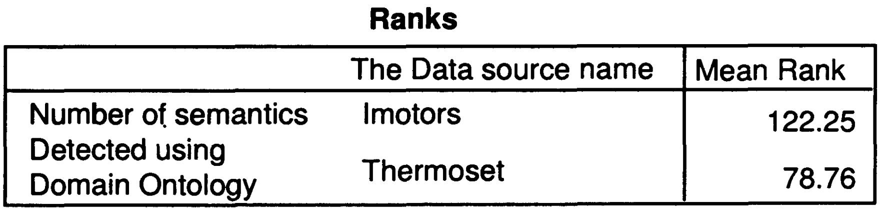
- The word or value after the mathematical symbol, along with the equation.
- The sentence location as it is needed later.

Tree searching will continue to the next sentence. If there is no match, we go to the first word in the next sentence. Repeat this unit until every sentence in the document has been processed. For example, in figure B.4 we have detected that " engine size : liters : 3.0 : 1.4.1 " which means that the engine size is represented in liters with the value 3.0 and it is located at the fourth paragraph in the first sentence.

APPENDIX C

Comparing the different domains

In this appendix, we test the relationship between the number of semantics in commercial web sites in the two domains. Also, we compare the number of semantics in evaluation data sources in the two domains. We have performed the five tests described in the next sections.



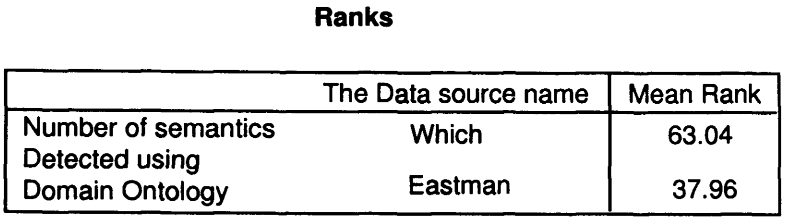
Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Domain Ontology	Imotors	122.25
	Thermoset	78.76

Figure C.1: *Mean Ranks from Mann-Whitney test for the number of semantics detected using the domain ontology approach in commercial data sources*

C.1 Testing domain ontology approach

If we are using a domain ontology approach to detect a document's semantics, is there any difference between documents from a scientific domain (chemistry) and a

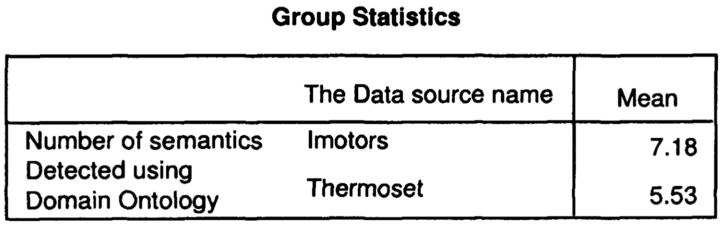


Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Domain Ontology	Which	63.04
	Eastman	37.96

Figure C.2: *Mean Ranks from Mann-Whitney test for the number of semantics detected using the domain ontology approach in evaluation data sources*

non-scientific domain (cars)? the Mann-whitney test and t-test show there is. From Figure C.1, we can see that the Imotors web site has higher mean rank (122.25) than the Thermoset web site (78.76) and also in Figure C.2 shows the Which web site has higher mean rank (63.04) than the Eastman (37.96). Also, the means (see Figures C.3 and C.4) of data sources for the non-scientific domain are higher than the means of the scientific domain.

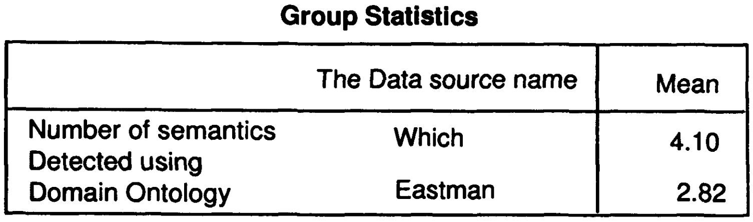


Group Statistics

	The Data source name	Mean
Number of semantics Detected using Domain Ontology	Imotors	7.18
	Thermoset	5.53

Figure C.3: *Means from T-Test for the number of semantics detected using the domain ontology approach in commercial data sources*

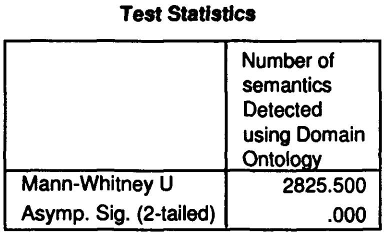
Figures C.5 and C.6 show that there is a significant difference between the scientific and non-scientific domains in both data sources, as shown by the Mann-Whitney test. The t-test, (see Figures C.7 and C.8) , with values of 'Sig.(2-tailed)' = .000 means that the significance is high, and that the lower and upper values in the 95 % confidence do not have any zeros and this gives us confidence that there is difference



Group Statistics

	The Data source name	Mean
Number of semantics Detected using Domain Ontology	Which	4.10
	Eastman	2.82

Figure C.4: *Means from T-Test for the number of semantics detected using the domain ontology approach in evaluation data sources*

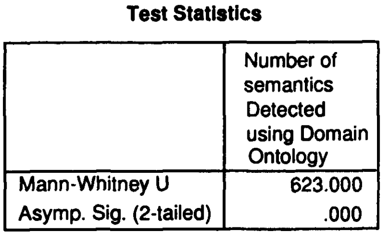


Test Statistics

	Number of semantics Detected using Domain Ontology
Mann-Whitney U	2825.500
Asymp. Sig. (2-tailed)	.000

Figure C.5: *Significance from Mann-Whitney for the number of semantics detected using the domain ontology approach between commercial data sources*

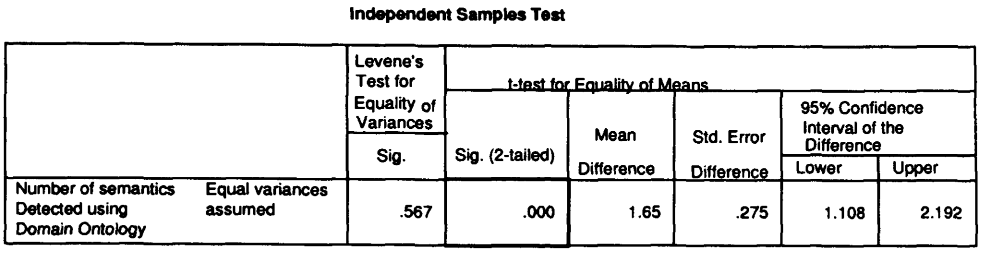
between the scientific and non-scientific domains.



Test Statistics

	Number of semantics Detected using Domain Ontology
Mann-Whitney U	623.000
Asymp. Sig. (2-tailed)	.000

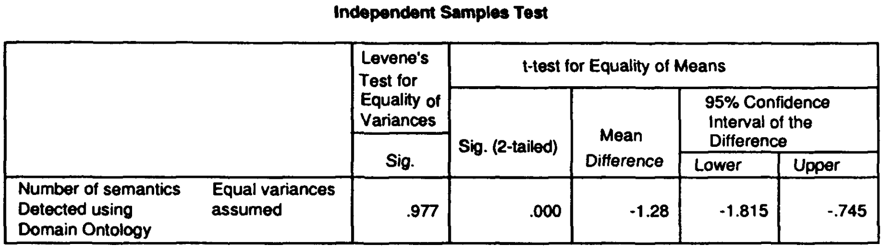
Figure C.6: *Significance from Mann-Whitney for the number of semantics detected using the domain ontology approach between evaluation data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means				
						95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Number of semantics Detected using Domain Ontology	Equal variances assumed	.567	.000	1.65	.275	1.108	2.192

Figure C.7: *Significance from T-Test for the number of semantics detected using the domain ontology approach between commercial data sources*



Independent Samples Test

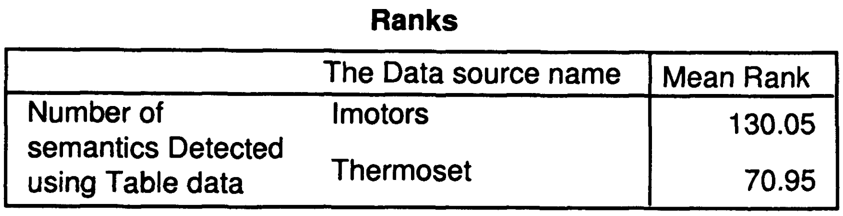
		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics Detected using Domain Ontology	Equal variances assumed	.977	.000	-1.28	-1.815	-.745

Figure C.8: *Significance from T-Test for the number of semantics detected using the domain ontology approach between evaluation data sources*

C.2 Testing the table data approach

In this test, we are comparing the two domains to determine which one has more semantics detected using the table data approach. The Imotors web site (Figure C.9), is a data source for the non-scientific domain and has a higher mean rank (130.05) than the Thermoset website (70.95). Also the Which web site, see Figure C.10, has a higher mean rank (68.55) than the Eastman website (32.45). The means for these data sources, (Figures C.11 and C.12) also show that the data sources from the non-scientific domain have a higher semantic content in this test than data sources from a scientific domain.

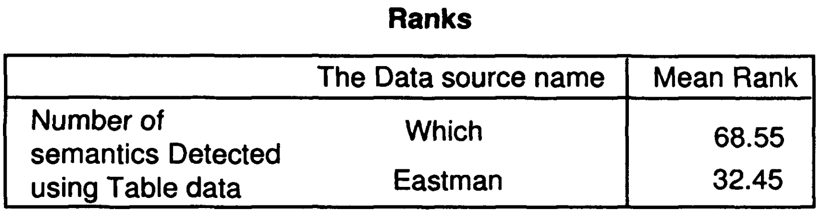
The Mann-Whitney and the independent sample T-tests show (Figures C.13,



Ranks

	The Data source name	Mean Rank
Number of semantics Detected using Table data	Imotors	130.05
	Thermoset	70.95

Figure C.9: *Mean Ranks from Mann-Whitney test for the number of semantics detected using the table data approach for commercial data sources*



Ranks

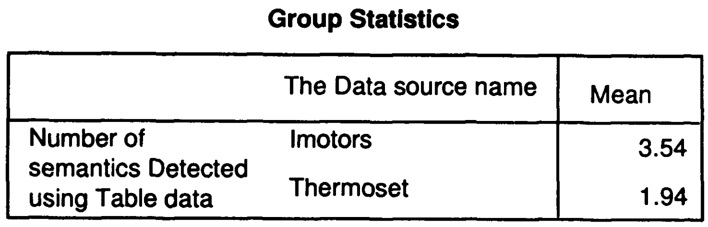
	The Data source name	Mean Rank
Number of semantics Detected using Table data	Which	68.55
	Eastman	32.45

Figure C.10: *Mean Ranks from Mann-Whitney test for the number of semantics detected using the table data approach for evaluation data sources*

C.14, C.15 and C.16) that there is a significant difference between the data sources from the car domain and the chemistry domain. The test results show that a non-scientific domain has more semantics detected by the Table data approach than the scientific domain. It is interesting that the results for the table approach is similar to the results for the ontology approach. This indicates that the two domains make use in similar ways of alternative terminology in the text.

C.3 Comparing semantics detected before the table

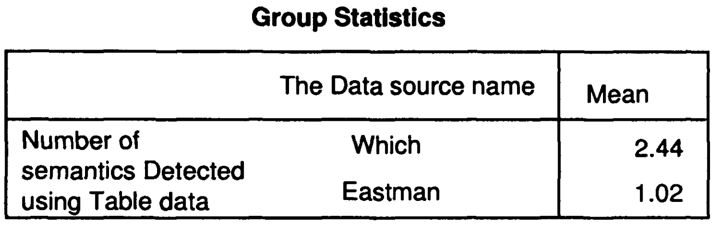
We have tested in sections C.2 and C.1, the differences between the different domains using the number of semantics detected, using different approaches, we are here doing



Group Statistics

	The Data source name	Mean
Number of semantics Detected using Table data	Imotors	3.54
	Thermoset	1.94

Figure C.11: *Means from T-Test for the number of semantics detected using the table data approach for commercial data sources*



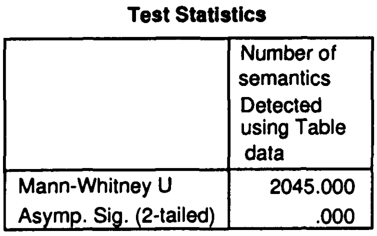
Group Statistics

	The Data source name	Mean
Number of semantics Detected using Table data	Which	2.44
	Eastman	1.02

Figure C.12: *Means from T-Test for the number of semantics detected using the table data approach for evaluation data sources*

the same tests, while concentrating on the physical position of the semantics in the documents. We compare the number of semantics detected in paragraphs before the table.

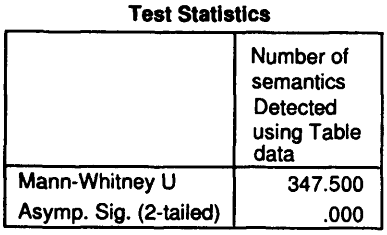
The Mann-whitney test, (Figures C.17 and C.18), show that the mean ranks for



Test Statistics

	Number of semantics Detected using Table data
Mann-Whitney U	2045.000
Asymp. Sig. (2-tailed)	.000

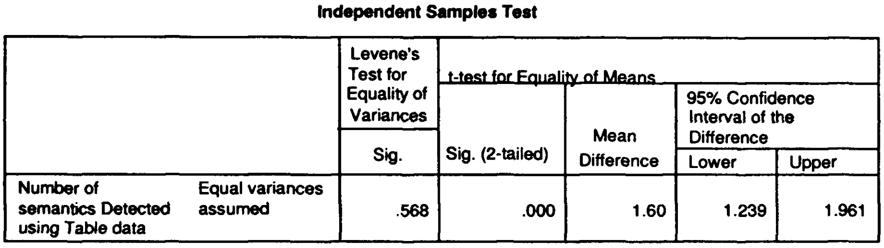
Figure C.13: *Significance from Mann-Whitney test for the number of semantics detected using the table data approach between commercial data sources*



Test Statistics

	Number of semantics Detected using Table data
Mann-Whitney U	347.500
Asymp. Sig. (2-tailed)	.000

Figure C.14: *Significance from Mann-Whitney test for the number of semantics detected using the table data approach between evaluation data sources*



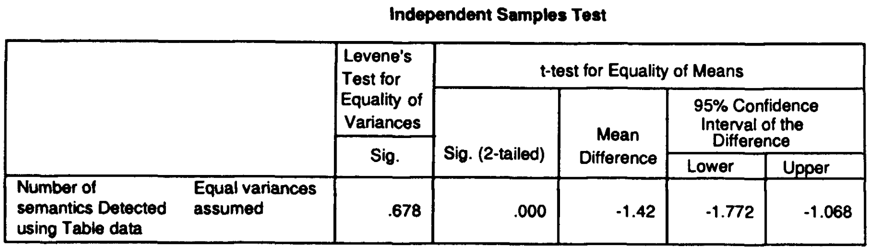
Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics Detected using Table data	Equal variances assumed	.568	.000	1.60	1.239	1.961

Figure C.15: *Significance from T-Test for the number of semantics detected using the table data approach between commercial data sources*

Imotors and Which web sites are higher than for Thermoset and Eastman. Also, the t-test, (Figures C.19 and C.20), show that there is a significant difference in means between the scientific and non-scientific domains. Imotors web site gives us 3.09, while the Thermoset web site gives 1.68 and the mean for the Which web site is 1.98, whereas in Eastman, it is only 0.76.

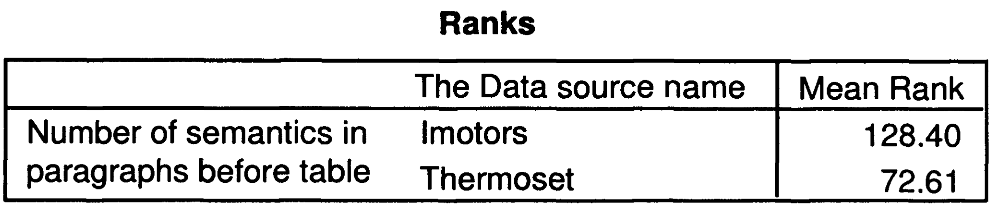
The Mann-Whitney and t-test results for this data (Figures C.21, C.22, C.23 and C.24,) show a significant difference between the scientific and non-scientific domains in the number of semantics detected in the paragraphs before the table.



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics Detected using Table data	Equal variances assumed	.678	.000	-1.42	-1.772	-1.068

Figure C.16: *Significance from T-Test for the number of semantics detected using the table data approach between evaluation data sources*



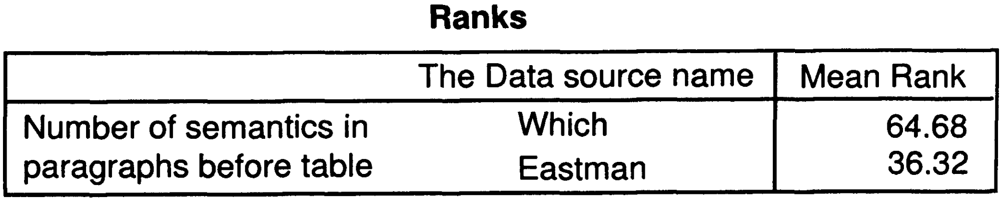
Ranks

	The Data source name	Mean Rank
Number of semantics in paragraphs before table	Imotors Thermoset	128.40 72.61

Figure C.17: *Mean Ranks from Mann-Whitney test for the number of semantics detected before the table for commercial data sources*

C.4 Comparing semantics detected after the table

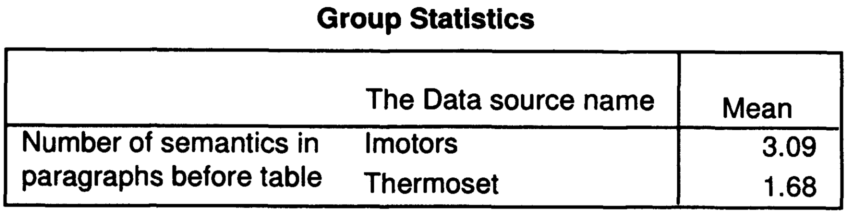
Like the test in section C.3, our test here concentrates on the physical location (paragraphs after tables) of the detected semantics in the documents, and which of



Ranks

	The Data source name	Mean Rank
Number of semantics in paragraphs before table	Which Eastman	64.68 36.32

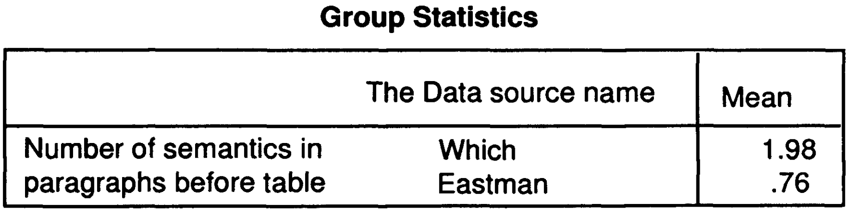
Figure C.18: *Mean Ranks from Mann-Whitney test for the number of semantics detected before the table for evaluation data sources*



Group Statistics

	The Data source name	Mean
Number of semantics in paragraphs before table	Imotors	3.09
	Thermoset	1.68

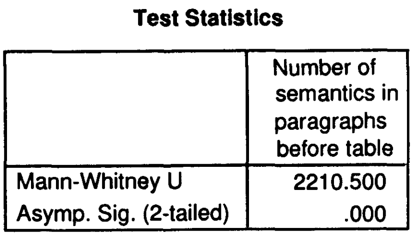
Figure C.19: *Means from T-Test for the number of semantics detected before the table for commercial data sources*



Group Statistics

	The Data source name	Mean
Number of semantics in paragraphs before table	Which	1.98
	Eastman	.76

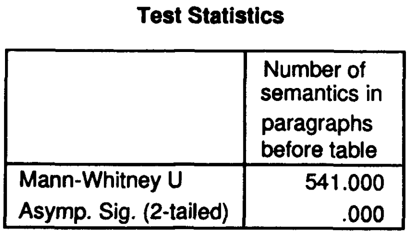
Figure C.20: *Means from T-Test for the number of semantics detected before the table for evaluation data sources*



Test Statistics

	Number of semantics in paragraphs before table
Mann-Whitney U	2210.500
Asymp. Sig. (2-tailed)	.000

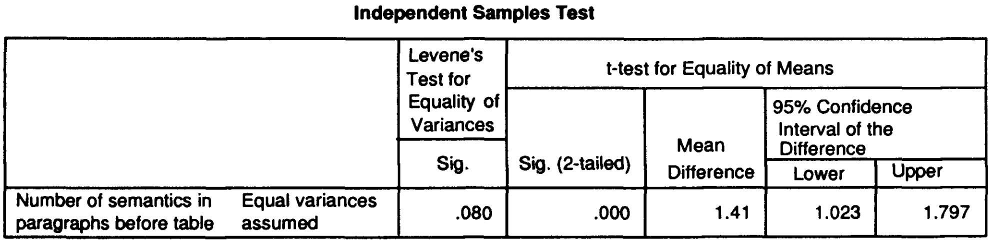
Figure C.21: *Significance from Mann-Whitney test for the number of semantics detected before the table between commercial data sources*



Test Statistics

	Number of semantics in paragraphs before table
Mann-Whitney U	541.000
Asymp. Sig. (2-tailed)	.000

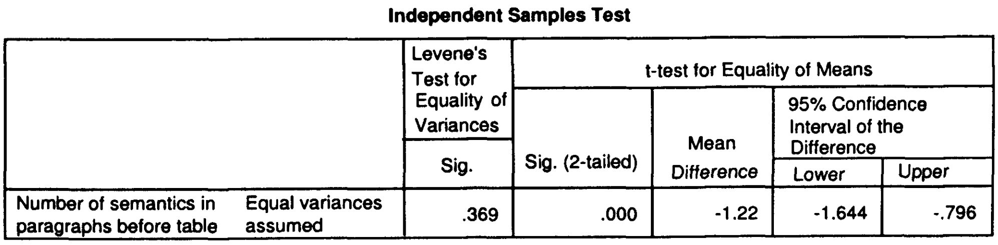
Figure C.22: *Significance from Mann-Whitney test for the number of semantics detected before the table between evaluation data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
		Sig.	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Number of semantics in paragraphs before table	Equal variances assumed	.080	.000	1.41	1.023	1.797

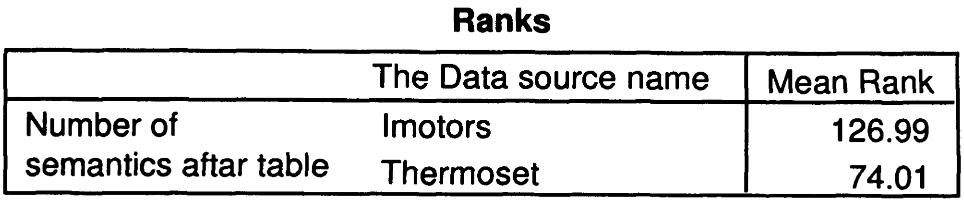
Figure C.23: *Significance from T-Test for the number of semantics detected before the table between commercial data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
		Sig.	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Number of semantics in paragraphs before table	Equal variances assumed	.369	.000	-1.22	-1.644	-.796

Figure C.24: *Significance from T-Test for the number of semantics detected before the table between evaluation data sources*

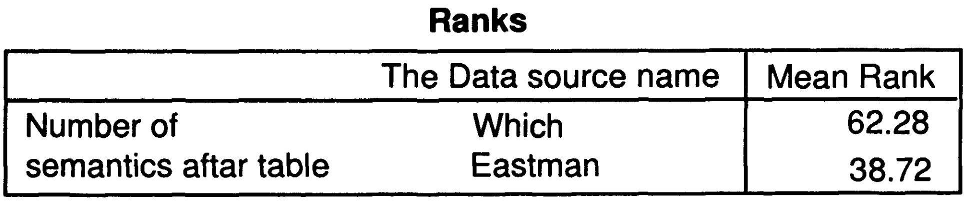


Ranks

	The Data source name	Mean Rank
Number of semantics aftar table	Imotors	126.99
	Thermoset	74.01

Figure C.25: *Mean Ranks from Mann-Whitney test for the number of semantics detected after the table for commercial data sources*

the two domains has more semantics in this part of the document. The results of this test, (Figures C.27 and C.28), show that the non-scientific domain again has more semantics than the scientific domain in this area. The Imotors web site has 41 % more semantics than the Thermoset website and the Which website has 58 % more semantics than the Eastman website. The mean ranks (Figures C.25 and C.26) show Imotors (129.99) is higher than in Thermoset website (74.01) and Which website with a mean rank (62.28) is higher than in Eastman (38.72).

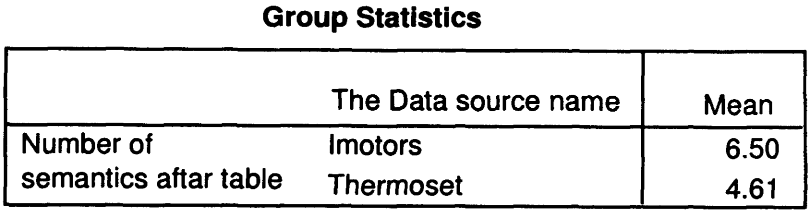


Ranks

	The Data source name	Mean Rank
Number of semantics aftar table	Which	62.28
	Eastman	38.72

Figure C.26: *Mean Ranks from Mann-Whitney test for the number of semantics detected after the table for evaluation data sources*

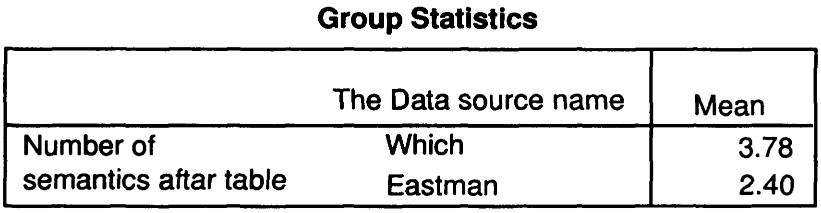
The Mann-Whitney and the t-test, (Figures C.29, C.30, C.31 and C.32) show a significant difference between the scientific and non-scientific domains in the number of semantics detected in the paragraphs after the table.



Group Statistics

	The Data source name	Mean
Number of semantics aftar table	Imotors	6.50
	Thermoset	4.61

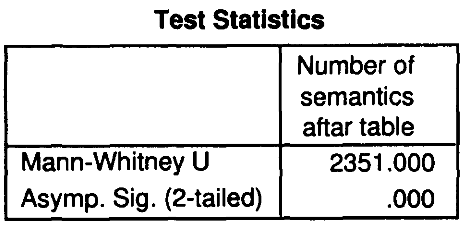
Figure C.27: *Means from T-Test for the number of semantics detected after the table for commercial data sources*



Group Statistics

	The Data source name	Mean
Number of semantics aftar table	Which	3.78
	Eastman	2.40

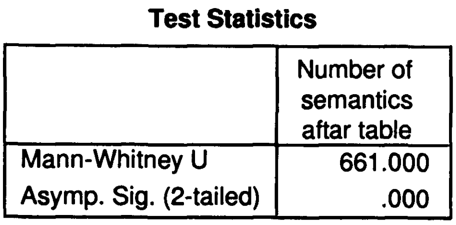
Figure C.28: *Means from T-Test for the number of semantics detected after the table for evaluation data sources*



Test Statistics

	Number of semantics aftar table
Mann-Whitney U	2351.000
Asymp. Sig. (2-tailed)	.000

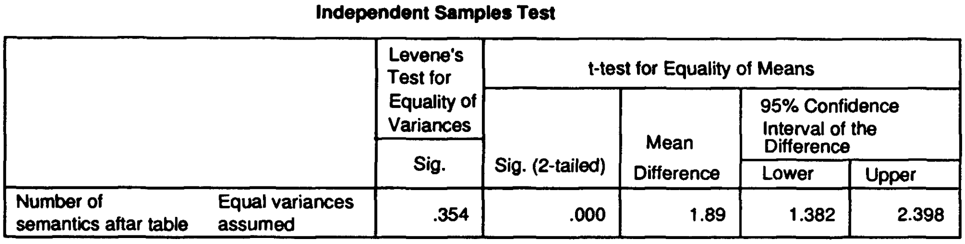
Figure C.29: *Significance from Mann-Whitney test for the number of semantics detected after the table between commercial data sources*



Test Statistics

	Number of semantics aftar table
Mann-Whitney U	661.000
Asymp. Sig. (2-tailed)	.000

Figure C.30: *Significance from Mann-Whitney test for the number of semantics detected after the table between evaluation data sources*



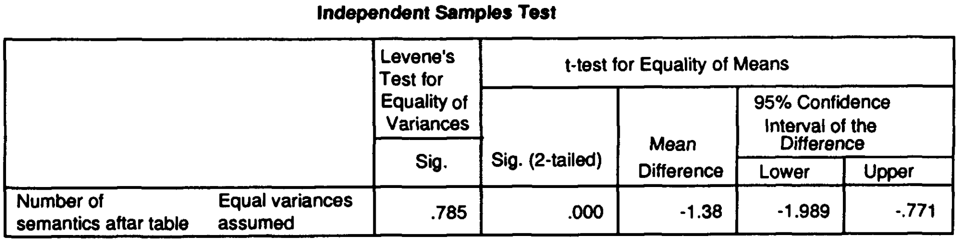
Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics aftar table	Equal variances assumed	.354	.000	1.89	1.382	2.398

Figure C.31: *Significance from T-Test for the number of semantics detected after the table between commercial data sources*

C.5 Comparing the total semantics detected

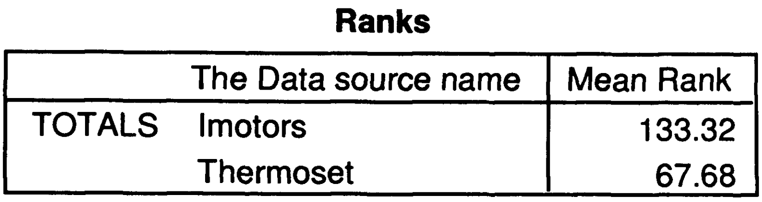
The last test we undertook was to compare the total number of semantics detected in documents to see if there is a significant difference between scientific and non-scientific domains. As expected, the total number of semantics detected in the



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
Number of semantics aftar table	Equal variances assumed	.785	.000	-1.38	-1.989	-.771

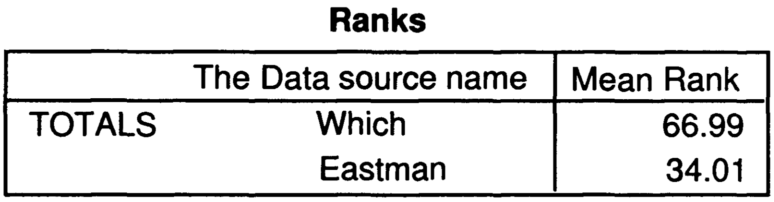
Figure C.32: *Significance from T-Test for the number of semantics detected after the table between evaluation data sources*



Ranks

	The Data source name	Mean Rank
TOTALS	Imotors	133.32
	Thermoset	67.68

Figure C.33: *Mean Ranks from Mann-Whitney test for the total number of semantics detected in the document for commercial data sources*



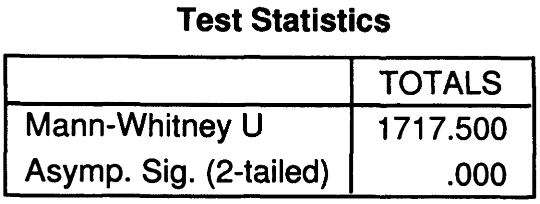
Ranks

	The Data source name	Mean Rank
TOTALS	Which	66.99
	Eastman	34.01

Figure C.34: *Mean Ranks from Mann-Whitney test for the total number of semantics detected in the document for evaluation data sources*

non-scientific domain is higher than in the scientific domain. Figures C.33 and C.34 show that the data sources from the non-scientific domain have a higher mean rank (Imotors:133.32 , and Which:66.99) than the data sources from the scientific domain (Thermoset:67.68, and Eastman:34.01). The Mann-Whitney test (Figures C.35 and C.36) show that, there is a significant difference between the scientific and non-scientific domains.

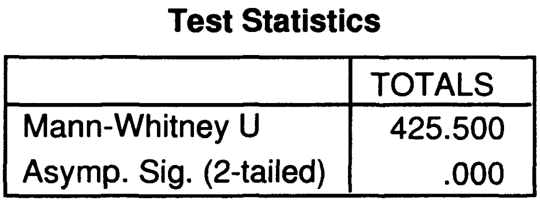
Figures C.37 and C.38, show that the data sources from the non-scientific domain have higher means (Imotors:9.59 , and Which:5.76) than the data sources from the scientific domain (Thermoset:6.29, and Eastman:3.16). The T- Test on these results(Figures C.39 and C.40), show that there is a significant difference between the scientific and non-scientific domains.



Test Statistics

	TOTALS
Mann-Whitney U	1717.500
Asymp. Sig. (2-tailed)	.000

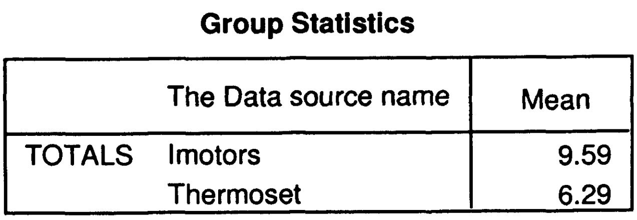
Figure C.35: *Significance from Mann-Whitney test for the total number of semantics detected in the document between commercial data sources*



Test Statistics

	TOTALS
Mann-Whitney U	425.500
Asymp. Sig. (2-tailed)	.000

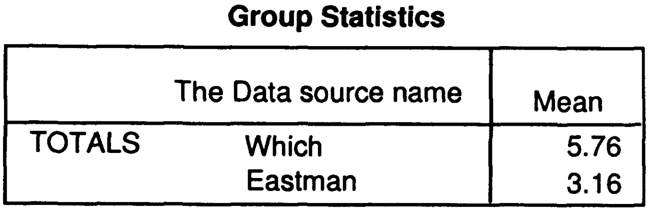
Figure C.36: *Significance from Mann-Whitney test for the total number of semantics detected in the document between evaluation data sources*



Group Statistics

	The Data source name	Mean
TOTALS	Imotors	9.59
	Thermoset	6.29

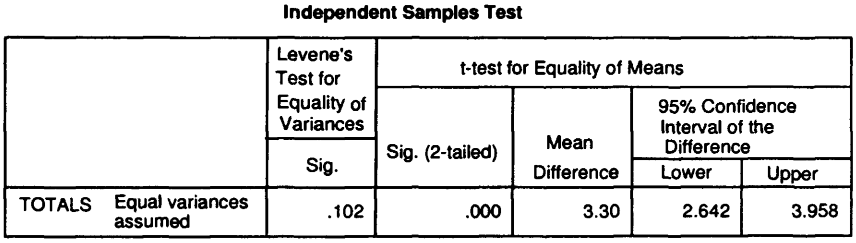
Figure C.37: *Means from T-Test for the total number of semantics detected in the document for commercial data sources*



Group Statistics

	The Data source name	Mean
TOTALS	Which	5.76
	Eastman	3.16

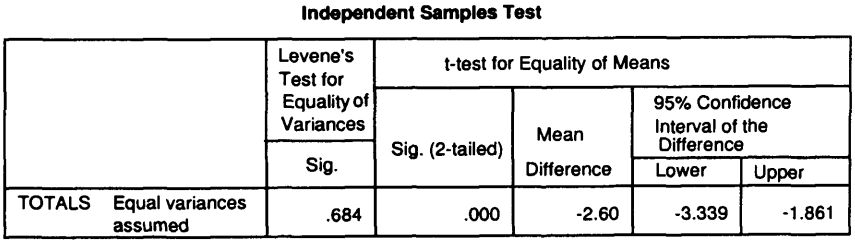
Figure C.38: *Means from T-Test for the total number of semantics detected in the document for evaluation data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
TOTALS	Equal variances assumed	.102	.000	3.30	2.642	3.958

Figure C.39: *Significance from T-Test for the total number of semantics detected in the document between commercial data sources*



Independent Samples Test

		Levene's Test for Equality of Variances	t-test for Equality of Means			
					95% Confidence Interval of the Difference	
		Sig.	Sig. (2-tailed)	Mean Difference	Lower	Upper
TOTALS	Equal variances assumed	.684	.000	-2.60	-3.339	-1.861

Figure C.40: *Significance from T-Test for the total number of semantics detected in the document between evaluation data sources*

Bibliography

[ACC$^+$97] Serge Abiteboul, Vassilis Christophides, Sophie Cluet, Tova Milo, Guido Moerkotte, and Jerome Simeon. Querying documents in object databases. *International Journal on Digital Libraries*, 1:5–19, 1997.

[Ade98] Brad Adelberg. Nodose - a tool for semi-automatically extracting structured and semistructured data from text documents. In Laura Haas and Ashutosh Tiwary, editors, *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data: June 1–4, 1998, Seattle, Washington, USA*, volume 27(2), pages 283–294. ACM Press, 1998.

[AG02] Saleh Alrashed and W. A. Gray. Detection approaches for table semantics in text. *Lecture Notes in Computer Science*, 2423:287–291, 2002.

[AG03a] Saleh Alrashed and W. A. Gray. Semantic detection for tabular data in text. In *In 7th World Multiconference on Systemics, Cybernetics and*

Informatics, pages 211–223, Orlando, Florida , USA, August 2003. IEEE, Computer Society.

[AG03b] Saleh Alrashed and W. A. Gray. Utilising semantic conversion functions to link tabular data. In *In The 9th. International Conference on Information Systems Analysis and Synthesis: ISAS '03*, pages 211–223, Orlando, Florida , USA, July 2003. IEEE, Computer Society.

[AHKI97] H. Ahonen, O. Heinonen, M. Klemettinen, and Inkeri Verkamo. Mining in the phrasal frontier. In Jan Komorowski and Jan Zytkow, editors, *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, volume 1263, pages 343–350, Berlin, June 24–27, 1997. Springer.

[AHKV97] H. Ahonen, O. Heinonen, M. Klemettinen, and I. Verkamo. Applying data mining techniques in text analysis. Technical Report C-1997-23, University of Helsinki, Department of Computer Science, March 1997. In: A. Hameurlain and A. Min Tjoa, editors, Proceedings of the 1997 DEXA Database and Expert Systems Applications, Toulouse, France, September 1-5. LNCS, Vol. 1308, pages 419–429.

[AK97] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. In *Proceedings Workshop on Management of Semistructured Data*, Tucson, USA, 1997.

[Alb93] L.K. Alberts. *YMIR: an ontology for engineering design.* Phd thesis, University of Twente, Enschede, The Netherlands, May 1993.

[AM97] Paolo Atzeni and Giansalvatore Mecca. Cut & paste. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 12–15, Tucson, Arizona, 1997.

[AM02] A. Antonacopoulos and H. Meng. A ground-truthing tool for layout analysis performance evaluation. *Lecture Notes in Computer Science*, 2423:236–244, 2002.

[AMM97] Paolo Atzeni, Giansalvatore Mecca, and Paolo Merialdo. To weave the web. In *VLDB '97*, pages 206–215, 1997.

[AQM$^+$97] Serge Abiteboul, Dallan Quass, Jason McHugh, Jennifer Widom, and Janet L. Wiener. The Lorel query language for semistructured data. *International Journal on Digital Libraries*, 1(1):68–88, 1997.

[Bat95] John A. Bateman. On the relationship between ontology construction and natural language: A socio-semiotic view. *International Journal of Human-Computer Studies*, 43(5/6):929–944, 1995.

[BB99] C. Bornhoevd and A. P. Buchmann. A prototype for metadata-based integration of Internet sources. *11th International Conference on Advanced Information Systems Engineering, CAiSE'99, Heidelberg, Germany, Jun. 14-18*, 1999.

[BC98] Alen Bryman and Duncan Cramer. *Quantitative Data Analysis with SPSS for Windows.* Routledge, London, 1998.

[BCWW97] M. Barley, P. Clark, K. Williamson, and S. Woods. The neutral representation project. *In Proceeding AAAI-97 Spring Symposium on Ontological Engineering,Stanford University, California, USA. AAAI Press.*, 1997.

[BDHS96] Peter Buneman, Susan Davidson, Gerd Hildebrand, and Dan Suciu. A query language and optimization techniques for unstructured data. In *Proceedings of theACM SIGMOD International Conference on Man-*

agement of Data, volume 25, 2 of *ACM SIGMOD Record*, pages 505–516, New York, June 4–6 1996. ACM Press.

[BGM96] S. Borgo, N. Guarino, and C. Masolo. Towards an ontological theory of physical objects. In Sandhurst, editor, *In Proceedings of IMACS-IEEE/SMC Conference Computational Engineering in Systems Applications (CESA 96), Symposium on Modelling, Analysis and Simulation.*, 1996.

[BLvHH00] Tim Berners-Lee, Frank van Harmelen, and Ian Horrocks. Daml-ont initial release. Technical report, The DARPA Agent Markup Language Homepage, 2000.

[BMM03] O. Bodenreider, J. Mitchell, and A. McCray. Biomedical ontologies. In *Proceedings of the 8th Pacific Symposium on Biocomputing (PSB 2003)*, 2003.

[Bor99] C. Bornhovd. Semantic metadata for the integration of web-based data for electronic commerce. In *Proceedings of the International Workshop on Advance Issues of ECommerce and Web-based Information System, Santa Clara, USA*, 1999.

[BR99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*, chapter 10: User Interfaces and Visualization, pages 257–323. Addison Wesley, Reading, US, 1999.

[Bre02] Thomas M. Breuel. Two geometric algorithms for layout analysis. *Lecture Notes in Computer Science*, 2423:188–199, 2002.

[Bur97] J. F. M. Burg. *Linguistic instruments in requirements engineering.* PhD thesis, Vrije Universiteit, Amsterdam, 1997.

[CCMM98] R. Cattoni, T. Coianiz, S. Messelodi, and C. Modena. Geometric layout analysis techniques for document image understanding: a review. a review, IRST, Trento, Italy, 1998.

[CGMH$^+$94] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey D. Ullman, and Jennifer Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *16th Meeting of the Information Processing Society of Japan*, pages 7–18, Tokyo, Japan, 1994.

[CGW96] Hamish Cunningham, Robert J. Gaizauskas, and Yorick Wilks. A general architecture for language engineering (gate) - a new approach to language engineering. January 30 1996. Comment: 52 page technical report, LaTeX 2e source.

[CHH$^+$01] Dan Connolly, Frank Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. DAML+OIL (March 2001) reference description. Note, W3C, March 2001.

[CSV98] R. Casati, Barry Smith, and A. C. Varzi. Ontological tools for geographic representation. In N. Guarino, editor, *Formal Ontology in Information Systems*, pages 77–85. IOS Press, Amsterdam, 1998.

[CV97] Roberto Casati and Achille C. Varzi. Spatial entities. In Oliviero Stock, editor, *Spatial and Temporal Reasoning*, pages 73–96. Kluwer Academic Publishers, Dordrecht, 1997.

[DC02] David Durand and Paul Caton. Semantic heterogeneity among document encoding schemes. Final Report 60NANB0D0115, Scholarly Technology Group, Brown University, January 2002.

[DEW96] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison-shopping agent for the world-wide web. Technical Report TR-96-01-03, University of Washington, Department of Computer Science and Engineering, January 1996.

[DN92] Abdel Kader Diagne and John Nerbonne. Flexible semantics communication in integrated speech/ language architectures. In G. Görz, editor, *1. Konferenz Verarbeitung natürlicher Sprache (KONVENS '92), 7.-9. Oktober*, pages 348–352, Nürnberg, Germany, 1992. Springer.

[Doe98] David Doermann. The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding: CVIU*, 70(3):287–298, 1998.

[DR02a] D. Dhanya and A. Ramakrishnan. Optimal feature extraction for bilingual OCR. *Lecture Notes in Computer Science*, 2423:25–36, 2002.

[DR02b] D. Dhanya and A. Ramakrishnan. Script identification in printed bilingual documents. *Lecture Notes in Computer Science*, 2423:13–24, 2002.

[DS97] Adamantis Diamantopoulos and Bodo Schlegelmilch. *Taking the Fear Out of Data Analysis.* The Dryden Press, London, 1997.

[DSK$^+$96] D. Doermann, J. Sauvola, H. Kauniskangas, C. Shin, M. Pietikainen, and A. Rosenfeld. The development of a general framework for intelligent document image retrieval. In *In Document Analysis Systems*, pages 605–932, Malvern, Pennsylvania, October 14-16 1996.

[ECJL98] D. W. Embley, D. M. Campbell, Y. S. Jiang, and S. W. Liddle. A conceptual-modeling approach to extracting data from the Web. *Lecture Notes in Computer Science*, 1507:78–91, 1998.

[EJN99] D. W. Embley, Y. S. Jiang, and Y.-K. Ng. Record-boundary discovery in web documents. In *SIGMOD'99*, pages 467–478, 1999. Accepted for publication.

[EX01] D. W. Embley and L. Xu. Locating and reconfiguring records in unstructured multiple-record Web documents. *Lecture Notes in Computer Science*, 1997:256–276, 2001.

[FBY92] W. B. Frakes and R. Baeza-Yates. *Information Retrieval Data Structures & Algorithms.* Prentice Hall, Englewood Cliffs, N.J., 1992.

[FD98] Ronen Feldman and Ido Dagan. Knowledge Discovery in Textual Databases (KDT). *Knowledge Discovery in Databases*, 1998.

[FFG94] F. Fadel, Mark S. Fox, and Michael Gruninger. A generic enterprise resource ontology. In *Proceedings of the Third Workshop on Enabling Technologies - Infrastructures for Collaborative Enterprises*, 1994.

[FFR96] A. Farquhar, R. Fikes, and J. Rice. The Ontolingua server: a tool for collaborative ontology construction. In *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge Based Systems Workshop (KAW95)*, Banff, Canada, 1996.

[FHH+00] D. Fensel, I. Horrocks, V. Harmelen, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In R. Dieng, editor, *Proceedings of the 12th. European Workshop on Knowledge Acquisition, Modeling and Management (EKAW-00)*, number 1937 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 2000.

[FHH+01] D. Fensel, F. Harmelen, I. Horrocks, D. McGuinness, and P. Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–44, 2001.

[FKBY$^+$97] R. Feldman, W. Klösgen, Y. Ben-Yehuda, G. Kedar, and V. Reznikov. Pattern based browsing in document collections. In Jan Komorowski and Jan Zytkow, editors, *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, volume 1263 of *LNAI*, pages 112–122, Berlin, June 24–27 1997. Springer.

[Fla98] Sharon Flank. A layered approach to NLP-based information retrieval. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 397–403, San Francisco, California, 1998. Morgan Kaufmann Publishers.

[FLdS01] Irna M. R. Evangelista Filha, Alberto H. F. Laender, and Altigran S. da Silva. Querying semistructured data by example: The qsbye interface. In *Workshop on Information Integration on the Web*, pages 156–163, 2001.

[FLGD87] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, November 1987.

[FM01] Richard Fikes and Deborah L. McGuinness. An axiomatic semantics for RDF, RDF Schema, and DAML+OIL. KSL Technical Report KSL-01-01, Stanford University, 2001.

[FPSS96] Usama Fayyad, Gregory (Piatetsky-Shapiro), and Padhraic Smyth. The (kdd) process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, November 1996.

[Gai97] Brian Gaines. Editorial: Using explicit ontologies in knowledge-based system development. *International Journal of Human-Computer Studies*, 46(2/3):181, 1997.

[GBM97] N. Guarino, S. Borgo, and C. Masolo. Logical modelling of product knowledge: Towards a well-founded semantics for step. In Sandhurst, editor, *Proceedings of the European Conference Product Data Technology Days*, pages 183–190, April 1997.

[GBMS99] Cheng Hian Goh, Stéphane Bressan, Stuart Madnick, and Michael Siegel. Context interchange: new features and formalisms for the intelligent integration of information. *ACM Transactions on Information Systems*, 17(3):270–270, July 1999.

[GdF03] Rosario Girardi and Carla Gomes de Faria. A generic ontology for the specification of domain models. In *Proceedings of the 1st International Workshop on Component Engineering Methodology (WCEM'03)*, 2003.

[GF92] M. R. Genesereth and R. E. Fikes. *Knowledge Interchange Format Version 3.0 Reference Manual.* Computer Science Department, Stanford University, Stanford, California, 94305, June 1992.

[GG96] N. Gotts and J. Goodday. A connection based approach to common-sense topological description and reasoning. *The Monist*, 79(1):51–75, 1996.

[GGPH03] James Geller, Huanying Gu, Yehoshua Perl, and Michael Halper. Semantic refinement and error correction in large terminological knowledge bases. *Data Knowledge Engineering*, 45(1):1–32, 2003.

[GL02] Michael Gruninger and Jintae Lee. Ontology: applications and design. *Communications of the ACM*, 45(2):39–41, February 2002.

[GMS96] C. H. Goh, M. E. Madnick, and M. D. Siegel. Context interchange: Representing and reasoning about data semantics in heterogeneous systems. Technical Report 33928, MIT Sloan School of Management, 1996.

[GN87] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, Los Altos, 1987.

[GO94] Thomas R. Gruber and Gregory R. Olsen. An ontology for engineering mathematics. In Pietro Torasso Jon Doyle and Erik Sandewall, editor, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 258–269, Bonn, FRG, May 1994. Morgan Kaufmann.

[GPS98] A. Gangemi, D. M. Pisanelli, and G. Steve. Ontology integration: experiences with medical terminologies. In N. Guarino, editor, *Formal Ontology in Information Systems*, pages 19–28. IOS Press, Amsterdam, 1998.

[Gru93] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Gua95] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, 43(5/6):625–640, 1995.

[Gua97a] N. Guarino. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. *Lecture Notes in Computer Science*, 1299:139–170, 1997.

[Gua97b] Nicola Guarino. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2/3):293–310, 1997.

[GW00] Nicola Guarino and Christopher Welty. Ontological analysis of taxonomic relationships. In *International Conference on Conceptual Modeling / the Entity Relationship Approach*, pages 210–224, 2000.

[Har91] Donna Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42:7–15, 1991.

[HD95] O. Hori and D. S. Doermann. Robust table-form structure analysis based on box-driven reasoning. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, page 218. IEEE Computer Society, 1995.

[Hei95] Sandra Heiler. Semantic interoperability. *ACM Computing Surveys*, 27(2):271–273, June 1995.

[HG01] F. Hakimpour and A. Geppert. Resolving semantic heterogeneity in schema integration: An ontology base approach. In Chris Welty and Barry Smith, editors, *Proceedings of International conference on Formal Ontologies in Information Systems FOIS'01. ACM Press*, Ogunquit, Maine, USA, October 2001. ACM Press.

[HGF98] L. E. Hodge, W. A. Gray, and N. J. Fiddian. A toolkit to facilitate the querying and integration of tabular data from semistructured documents. *Lecture Notes in Computer Science*, 1405:171–172, 1998.

[HHL99] Jeff Heflin, James Hendler, , and Sean Luke. Shoe: A knowledge representation language for internet applications. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1999.

[Hoc94] Rainer Hoch. Using IR techniques for text classification in document analysis. Research Report RR-94-19, Deutsches Forschungszentrum für

Künstliche Intelligenz, Deutsches Forschungszentrum für Künstliche
Intelligenz GmbH
Erwin-Schrödinger Strasse
Postfach 2080
67608 Kaiserslautern
Germany, 1994.

[Hod01] Leigh E. Hodge. *Approaches for the Reuse of Tabluar Data in Semi-structured Textual Documents.* PhD thesis, Cardiff School of Computer Science, Cardiff, UK, 2001.

[Hor02] Ian Horrocks. Reasoning with expressive description logics: Theory and practice. In Andrei Voronkov, editor, *Automated Deduction – CADE-18*, volume 2392 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, July 27-30 2002.

[HPH02] Ian Horrocks, Peter Patel, and Harmelen Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI-02)*, pages 792–797, Menlo Parc, CA, USA, July 28- August 1 2002. AAAI Press.

[HPM02] Jeffrey A. Hoffer, Mary B. Prescott, and Fred R. McFadden. *Modern Database Management.* Prentice Hall, 6 edition, January 2002.

[HROM00] Thomas S. Huang, Yong Rui, Michael Ortega, and Sharad Mehrotra. Information retrieval beyond the text document. June 2000. Yong Rui, Michael Ortega, Thomas S. Huang, Sharad Mehrotra, "Information Retrieval Beyond the Text Document" *Library Trends*, Vol. 48, No. 2, pp. 437-456.

[HSW97] G. Heijst, A. Schreiber, and B. Wielinga. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, 46(2-3):183–292, 1997.

[Hul96] David A. Hull. Stemming algorithms: a case study for detailed evaluation. *J. Am. Soc. Inf. Sci.*, 47(1):70–84, 1996.

[Hur00] Matthew Hurst. *The Interpretation of Tables in Text.* PhD thesis, The University of Edinburgh, 2000.

[KCT99] P. D. Karp, V. K. Chaudhri, and J. Thomere. Xol: An xml-based ontology exchange language. Technical report, Stanford University, California, 1999.

[KLW95] Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4):741–843, 1995.

[KP96] Wessel Kraaij and Ren&#233;e Pohlmann. Viewing stemming as recall enhancement. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 40–48. ACM Press, 1996.

[Kro93] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Linguistic Analysis, pages 191–202, 1993.

[Lan91] Ewald Lang. The LILOG ontology from a linguistic point of view. In O. Herzog and C.-R. Rollinger, editors, *Text understanding in LILOG: integrating computational linguistics and artificial intelligence, Final* *report on the IBM Germany LILOG-Project*, pages 464–481. Springer, Berlin, 1991. Lecture notes in artificial intelligence, 546.

[LD01] Martin S. Lacher and Stefan Decker. RDF, Topic Maps, and the Semantic Web. *Markup Languages: Theory & Practice*, 3(3):313–331, Summer 2001.

[Leh96] F. Lehmann. Machine-negotiated, ontology-based EDI (electronic data interchange). *Lecture Notes in Computer Science*, 1028:27–??, 1996.

[LRO96] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In T. M. Vijayaraman, Alejandro P. Buchmann, C. Mohan, and Nandlal L. Sarda, editors, *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pages 251–262, Mumbai (Bombay), India, 3–6 September 1996. Morgan Kaufmann.

[LS93] G. F. Luger and W. A. Stubblefield. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Benjamin-Cummings, Redwood Cliffs, CA., 1993.

[Mac91] Robert M. MacGregor. Inside the loom description classifier. *SIGART Bull.*, 2(3):88–92, 1991.

[Mad95] Stuart E. Madnick. From VLDB to VMLDB (very MANY large data bases): Dealing with large-scale semantic heterogenity. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases*, pages 11–16, Zurich, Switzerland, 11–15 September 1995. Morgan Kaufmann.

[Mah96] K. Mahesh. Ontology development for machine translation: Ideology and methodology. Technical report, Memoranda in Computer and Cognitive Science, MCCS-96-292. Las Cruces, NM, New Mexico State University, Computing Research Laboratory., 1996.

[MC00] M. Mitra and B. B. Chaudhuri. Information retrieval from documents: A survey. *Information Retrieval*, 2(2/3):141–163, 2000.

[McG98] Deborah L. McGuinness. Ontological issues for knowledge-enhanced search. Technical report, In Proceedings of Formal Ontology in Information Systems, Washington, D.C, 1998.

[MFHS02a] Deborah McGuinness, Richard Fikes, Hendler Hendler, and Lynn Andrea Stein. DAML+OIL: An ontology languages for the semantic web. *IEEE Intelligent Systems*, 2002.

[MFHS02b] Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. IEEE intelligent systems: DAML + OIL: An ontology language for the Semantic Web. *IEEE Distributed Systems Online*, 3(11), 2002.

[Min74] Marvin Minsky. A framework for representing knowledge. Technical report, Massachusetts Institute of Technology, cambridge, 1974.

[MKSI98] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. Domain specific ontologies for semantic information brokering on the global information infrastructure. In *Proceedings of the First International Conference on Formal Ontologies in Information Systems*, Trento, Italy, June 1998.

[Mot99] E. Motta. *Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving.* IOS Press, 1999.

[Myl80] John Mylopoulos. An overview of knowledge representation. In *Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling*, pages 5–12. ACM Press, 1980.

[Ner96] John Nerbonne. Computational semantics linguistics and processing. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 461–484. Blackwell Publishers, Oxford, 1996.

[Niy94] Debashish Niyogi. A knowledge-based approach to deriving logical structure from document images. Technical Report 94-35, Department of Computer Science, SUNY Buffalo, August 1994.

[Paz98] Luca Pazzi. Three points of view in the characterization of complex entities. In *In N. Guarino (ed.) Formal Ontology in Information Systems.* IOS Press, 1998.

[PC97] Pallavi Pyreddy and W. Bruce Croft. TINTIN: A system for retrieval in text tables. In *DL'97: Proceedings of the 2nd ACM International Conference on Digital Libraries*, Databases, pages 193–200, 1997.

[Per99] Asuncion Gomez Perez. Evaluation of taxonomic knowledge and knowledge bases. In *Proceedings of Twelfth Workshop on Knowledge Acquisition, Modeling and Management*, Alberta, Canada, October 16-21 1999.

[Pol96] Riccardo Poli. Ontology and knowledge organization. In *Proceedings of 4th Conference of the International Society of Knowledge Organization*, pages 313–319, Indeks, Frankfurt, 1996.

[Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[PW92] Popovic and Willett. The effectiveness of stemming for natural-language access to slovene textual data. *Journal of the American Society for Information Science*, 43:384–390, 1992.

[Qui68] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, 1968.

[RB98] Martin Rajman and Romaric BESANÇON. Text mining: Natural language techniques and text mining applications. June 22 1998.

[RBD98] Reind Riet, Hans Burg, and Frank Dehne. Linguistic instruments in information systems design. In N. Guarino, editor, *Formal Ontology in Information Systems*, pages 39–60. IOS Press, Amsterdam, 1998.

[RD88] G. A. Ringland and D. A. Duce. *Approaches to knowledge representation: an introduction.* Research Studies Press, Hertfordshire, UK, 1988.

[Rei84] R. Reiter. Towards a logical reconstruction of relational database theory. In M. L. Brodie, J. Mylopoulos, and J. W. Schmidt, editors, *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*, pages 191–233. Springer, New York, 1984.

[RHI01] Nicolas Roussel, Oliver Hitz, and Rolf Ingold. Web-based cooperative document understanding. In *Proceedings of Sixth International Conference on Document Analysis and Recognition (ICDAR '01)*, pages 368–377, Seattle, Washington, September 10-13 2001. IEEE.

[RL94] Ellen Riloff and Wendy Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, 1994.

[SCH$^+$02] S. Staab, P. Clark, J. Hendler, I. Horrocks, P. Patel-Schneider, M.-C. Rousset, G. Schreiber, M. Uschold, and F. van Harmelen. Ontologies KISSES in Standardization. *IEEE Intelligent Systems, Trends & Controversies*, 17(2):70–79, March/April 2002.

[SFDB99] Rudi Studer, Dieter Fensel, Stefan Decker, and V. Richard Benjamins. Knowledge engineering : Survey and future directions. In Frank Puppe, editor, *Proceedings of the 5th Biannual German Conference on Knowledge-Based Systems (XPS-99)*, volume 1570 of *LNAI*, pages 1–23, Berlin, March 3–5 1999. Springer.

[SFJ02] Urvi Shah, Tim Finin, and Anupam Joshi. Information retrieval on the semantic web. In Konstantinos Kalpakis, Nazli Goharian, and David Grossmann, editors, *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM-02)*, pages 461–468, New York, November 4–9 2002. ACM Press.

[SL97] D. Smith and M. Lopez. Information extraction for semi-structured documents. In *Proc. Workshop on Management of Semistructured Data*, Tucson, 1997.

[SM00] S. Staab and A. Maedche. Ontology engineering beyond the modeling of concepts and relations. In *Proceedings of the ECAI'2000 Workshop on Application of Ontologies and Problem-Solving Methods*, Amsterdam, 2000. IOS Press.

[Sow99] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Thomson Learning, Stamford, Connecticut, 1999.

[SQ02] Elisa Smith and Xiaohui Qiu. Relating statistical image differences and degradation features. *Lecture Notes in Computer Science*, 2423:1–12, 2002.

[SSR94] Edward Sciore, Michael Siegel, and Arnon Rosenthal. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Transactions on Database Systems*, 19(2):254–290, June 1994.

[SWS$^+$00] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Pattern Analysis and Machine Intelligence*, 22:1349–1380, 2000.

[TEG$^+$95] Samson W. Tu, Henrik Eriksson, John Gennari, Yuval Shahar, and Mark A. Musen. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: Application of protg-ii to protocol-based decision support. In *Proceedings of the 7th Conference on Artificial Intelligence in Medicine*, 1995.

[TSN00] E. Tjong, Kim Sang, and John Nerbonne. Learning the logic of simple phonotactics. In James Cussens and Saso Dzeroski, editors, *Learning Language in Logic*, volume 1925 of *Lecture Notes in Computer Science*, pages 110–124. Springer-Verlag, June 2000.

[UG96] Mike Uschold and Michael Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 1(2):93–155, June 1996.

[vO02] Jacco van Ossenbruggen. Towards semantic web document engineering. In *W3C Workshop on Delivery Context*, Sophia-Antipolis, France, 4-5 March 2002.

[Wan89] Y. Wand. A proposal for a formal model of objects. In *In 'Object-Oriented Concepts, Databases, and Applications, edited by Won Kim and Frederick H.Lochovsky.* 1989. Also published in: ACM SIGMOD Record, Vol.18 No.3, Sep.1989, p.50.

[Wel98] C. Welty. The ontological nature of subject taxonomies. In *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS'98)*, TRENTO, ITALY, June 1998. IOS Press.

[WH02] Y. Wang and J. Hu. Detecting tables in html documents. In J. Hu D. Lopresti and R. Kashi, editors, *In Fifth IAPR International Workshop on Document Analysis Systems*, volume 2423, pages 249–260, Princeton, New Jersey, USA, August 2002. Springer-Verlag Heidelberg.

[Wie96] Gio Wiederhold. *Intelligent Integration of Information*, volume 6. Kluwer Academic Publishers, Boston, July 1996.

[WS93] Bob Wielinga and A. T. Schreiber. Reusable and sharable knowledge bases: A european perspective. In *International Conference on Building and Sharing of Very Large-Scaled Knowledge Bases*, Tokyo, Japan, 1993.

[YTT01] M. Yoshida, K. Torisawa, and J. Tsujii. A method to integrate tables of the world wide web. In *Proceedings of the First International Workshop on Web Document Analysis*, volume 1, pages 31–34, Seattle, Washington, USA, September 2001. ICDAR'01.