This item was submitted to Loughborough's Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/

Automated Retrieval and Extraction of Training Course Information from Unstructured Web Pages

Daniela Xhemali

Apricot Training Management
Limehurst House
Bridge Street
Loughborough
Leicestershire, LE11 1NH
United Kingdom

Centre for Innovative and Collaborative Engineering (CICE)
Loughborough University
Loughborough
Leicestershire, LE11 3TU
United Kingdom

AUTOMATED RETRIEVAL AND EXTRACTION OF TRAINING COURSE INFORMATION FROM UNSTRUCTURED WEB PAGES

By
Daniela Xhemali

A dissertation thesis submitted in partial fulfilment of the requirements for the award of the degree Doctor of Engineering (EngD), at Loughborough University

[9 July 2010]

© BY DANIELA XHEMALI (2010)

Apricot Training Management
Limehurst House
Bridge Street
Loughborough
Leicestershire, LE11 1NH
United Kingdom

Centre for Innovative and Collaborative Engineering
Department of Civil & Building Engineering
Loughborough University
Loughborough
Leicestershire, LE11 3TU
United Kingdom

ACKNOWLEDGEMENTS

I would like to express my heartfelt thanks to a number of people whose support and encouragement has made this research possible. First of all I would like to thank my supervisors, Prof. Chris Hinde and Dr. Roger Stone for their true commitment to this project and their invaluable advice and guidance. I can honestly say that I have never met people more dedicated to their work and to their students’ success than Prof. Hinde and Dr. Stone. I feel very lucky to have had them as my supervisors for this difficult project.

An immense thank you goes to my family, especially my fiancé Will, who have seen me go through a rollercoaster of emotions these past four years. I must apologise for some stressful moments that I have brought to their lives, especially during the last year, and thank them for their love, support, understanding and encouragement and for always believing in me. I also thank my friends, particularly Daniel Sills, who have been there for me when I needed them.

I also extend my thanks to the sponsoring company, Apricot Training Management, for treating me like one of their employees and supporting me throughout this research. I could not have hoped to work with kinder and more supportive people.

Last but not least, thank you to the CICE Centre staff (Jo, Sara and Jen) for their administrative support and thank you to Mr. Samra for his technical assistance.

ABSTRACT

Web Information Extraction (WIE) is the discipline dealing with the discovery, processing and extraction of specific pieces of information from semi-structured or unstructured web pages. The World Wide Web comprises billions of web pages and there is much need for systems that will locate, extract and integrate the acquired knowledge into organisations' practices. There are some commercial, automated web extraction software packages, however their success comes from heavily involving their users in the process of finding the relevant web pages, preparing the system to recognise items of interest on these pages and manually dealing with the evaluation and storage of the extracted results.

This research has explored WIE, specifically with regard to the automation of the extraction and validation of online training information. The work also includes research and development in the area of automated Web Information Retrieval (WIR), more specifically in Web Searching (or Crawling) and Web Classification. Different technologies were considered, however after much consideration, Naïve Bayes Networks were chosen as the most suitable for the development of the classification system. The extraction part of the system used Genetic Programming (GP) for the generation of web extraction solutions. Specifically, GP was used to evolve Regular Expressions, which were then used to extract specific training course information from the web such as: course names, prices, dates and locations.

The experimental results indicate that all three aspects of this research perform very well, with the Web Crawler outperforming existing crawling systems, the Web Classifier performing with an accuracy of over 95% and a precision of over 98%, and the Web Extractor achieving an accuracy of over 94% for the extraction of course titles and an accuracy of just under 67% for the extraction of other course attributes such as dates, prices and locations. Furthermore, the overall work is of great significance to the sponsoring company, as it simplifies and improves the existing time-consuming, labour-intensive and error-prone manual techniques, as will be discussed in this thesis. The prototype developed in this research works in the background and requires very little, often no, human assistance.

KEY WORDS

Web page, Information Retrieval, Information Extraction, Web Classifier, Naïve Bayes Classifiers, Genetic Programming, Regular Expressions.

PREFACE

The research behind this thesis was undertaken between 2006 and 2010 in partial fulfilment of the requirements of an Engineering Doctorate (EngD) at the Centre for Innovative and Collaborative Engineering (CICE), Loughborough University. The research programme was supervised by CICE, funded by the Engineering Physical Sciences Research Council and sponsored by Apricot Training Management Limited, an independent not-for-profit organisation.

The EngD is a four-year, industry based, doctoral programme. Whilst as prestigious as a Philosophy Doctorate (PhD), the EngD has enhanced doctorate features such as: management training, master level courses, applied research focus to meeting essential industry demands and high degree of involvement from existing organisations.

The EngD is examined on the basis of a thesis containing at least three (but no more than five) research publications and/or technical reports. This thesis is supported by two published journal papers and three published conference papers. As these papers are an essential part of this thesis, they should be read when referenced in conjunction with the thesis. The papers can be found in Appendices F to J.

USED ACRONYMS / ABBREVIATIONS

ANN	Artificial Neural Network
ATM	Apricot Training Management (the sponsor)
CRM	Customer Relationship Management
DOM	Document Object Model
DT	Decision Trees
DTC	Decision Tree Classifier
EngD	Engineering Doctorate
FIFO	First In First Out
GP	Genetic Programming
HTML	Hypertext Mark-up Language
LCP	Link Checker Pro
ML	Machine Learning
MPN	Multilayer Perceptron Network
MS	Microsoft
NB	Naïve Bayes
NLP	Natural Language Processing
NN	Neural Networks
OO	Object Oriented
OS	Operating System
PHP	PHP: Hypertext Pre-processor
POSIX	Portable Operating System Interface for Unix
RE	Regular Expression
ROC	Receiver Operating Characteristic
SSP	System Scheduler Professional
SVM	Support Vector Machines
URL	Uniform Resource Locator
WIE	Web Information Extraction
WIR	Web Information Retrieval
WLV	Web Link Validator
WWW	World Wide Web
XML	eXtensible Mark-up Language

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 INTRODUCTION

This chapter provides an introduction to the problem under investigation. It defines the background to the research undertaken, lists the aims and objectives and presents justification for the research. The sponsoring company is also introduced and the structure of the thesis, together with the accompanying papers, is laid out.

1.2 SUBJECT DOMAIN AND PROBLEM DEFINITION

The Internet is rapidly becoming the first place people go to when seeking answers, conducting research, carrying out work or communicating with family and friends. The flexibility of the Internet and the increase in unlimited, high-speed broadband connections have won over billions of people of all ages. According to Google (Google, 2009), over 1.4 billion people, which is nearly a quarter of the world's population, use the Internet, with more than 200 million new people coming online every year. The freedom offered by the Internet has meant that an astonishing number of individuals and companies upload their information online for other people to see. Despite the existence of many duplicated web pages and inconsistency in the information presented online and sometimes even completely misleading or useless information being published, the Internet remains a place many people use to extract information.

The size of the Web, however, is overwhelming. According to Yadav et al. (2009) and De Kudner (2006; 2010), there exist at least 19 billion indexed web pages on the Web, and it has been estimated that the volume of data stored in global databases doubles every 20 months (Witten and Frank, 1999). This magnitude of data, as well as the varying nature of the web and the inexperience of many users with web searching, makes it difficult for people to locate the exact information they need. Experimental studies (Holscher & Strube, 2000) investigating the effects of web experience and domain-specific knowledge in web searching tasks, showed that experience and domain knowledge counted for only 24% increase in web searching success (web novices achieved 40% success rate, whilst experts achieved 64% success rate in the web-searching tasks assigned to them during these studies). In relation to the workplace, IBM research (IBM, 2010) has shown that employees spend 25% of their time just looking for information, and every week, 42% of people use incorrect information on which to base their decisions, resulting in the need to redo the work. Furthermore, 50% of all web searches are unsuccessful (Feldman, 2004).

The above shows that in order to achieve a greater competitive edge, organisations, whose success depends on accurate, up-to-date information being mined from the Web, need to embrace techniques that do not depend solely on the employees' efforts in finding and collecting information useful to the business. Web Information Retrieval (WIR) and Web Information Extraction (WIE) are such techniques, as they aim to automate the discovery, processing and extraction of specific pieces of information from semi-structured and/or unstructured web pages at a much faster speed than human beings. Applications, which claim to already achieve this (see section 2.5) tend to concentrate only on the WIE side, leaving the user responsible for searching and finding the appropriate sites containing the items of interest manually, which does not resolve the effort and time issues mentioned above. These applications also rely heavily on the involvement of the user throughout the WIE process, which is far from ideal, as will be discussed later on.

The research presented in this thesis deals with both WIR and WIE as they were deemed to be complementary to each other in assisting the sponsoring company with its problems. The existence of a WIR system means that only web pages containing training information are this way considered and analysed by the WIE system. The specific techniques used in this research include Naïve Bayes Networks for the automatic classification of web pages and Genetic Programming principles for the evolution of optimised solutions to web extraction. These techniques were enhanced to not only improve the entirely manual and expensive process of finding and capturing information at the sponsoring company, but also to contribute to the WIR and WIE research communities.

1.3 THE INDUSTRIAL SPONSOR

Apricot Training Management Limited (ATM) is a not-for-profit business that is dedicated to helping individuals and employers from all sectors to understand their skills needs and to find the training and staffing solutions that meet these needs. This is achieved by providing a brokerage service that develops the skills and employability of both the current and potential workforce. ATM is largely funded through public sector funds and grants. Established in November 2002, ATM was set up in response to the difficulty many businesses experience in sourcing and procuring the training to meet their needs. The company's vision is:

"To be the best provider of employer-led employability and training brokerage service in the East Midlands and a powerful advocate of the importance of the connectivity between business performance, individual performance and economic regeneration." (ATM, 2009)

New businesses are the lifeblood of the economy. The vast majority of employers in the country are small and medium size enterprises (SMEs) and they are constant source of new ideas and innovation. ATM see such new and small businesses as important customers and over the last seven years, ATM has assisted over 2000 of them in the East Midlands with their recruitment, training and development needs.

ATM is committed to becoming the market leader for specialist employability brokerage services in the East Midlands and to provide superior service and good value for money to both their clients and their funding stakeholders. They have recognised that a WIR/WIE system may be able to offer them considerable benefits, especially when dealing with large numbers of customers. Such system would help the company to not only manage the recent volume of orders but also to provide the highest quality services and achieve their goals.

1.4 RESEARCH JUSTIFICATION AND SCOPE

The justification for this research derives from the need for change at ATM. ATM's advisors are responsible for identifying and analysing organisations' training needs and then creating customised training plans by finding suitable courses for their employees. Preferably, between three and five different course options should be sourced for each client. This is time consuming, especially considering there are many providers offering similar courses on similar subjects.

Figure 1.1 shows a typical process the advisors go through to find courses. Some courses are also found by ordering the latest prospectuses from different providers. These are catalogued, shelved and manually entered into the database.

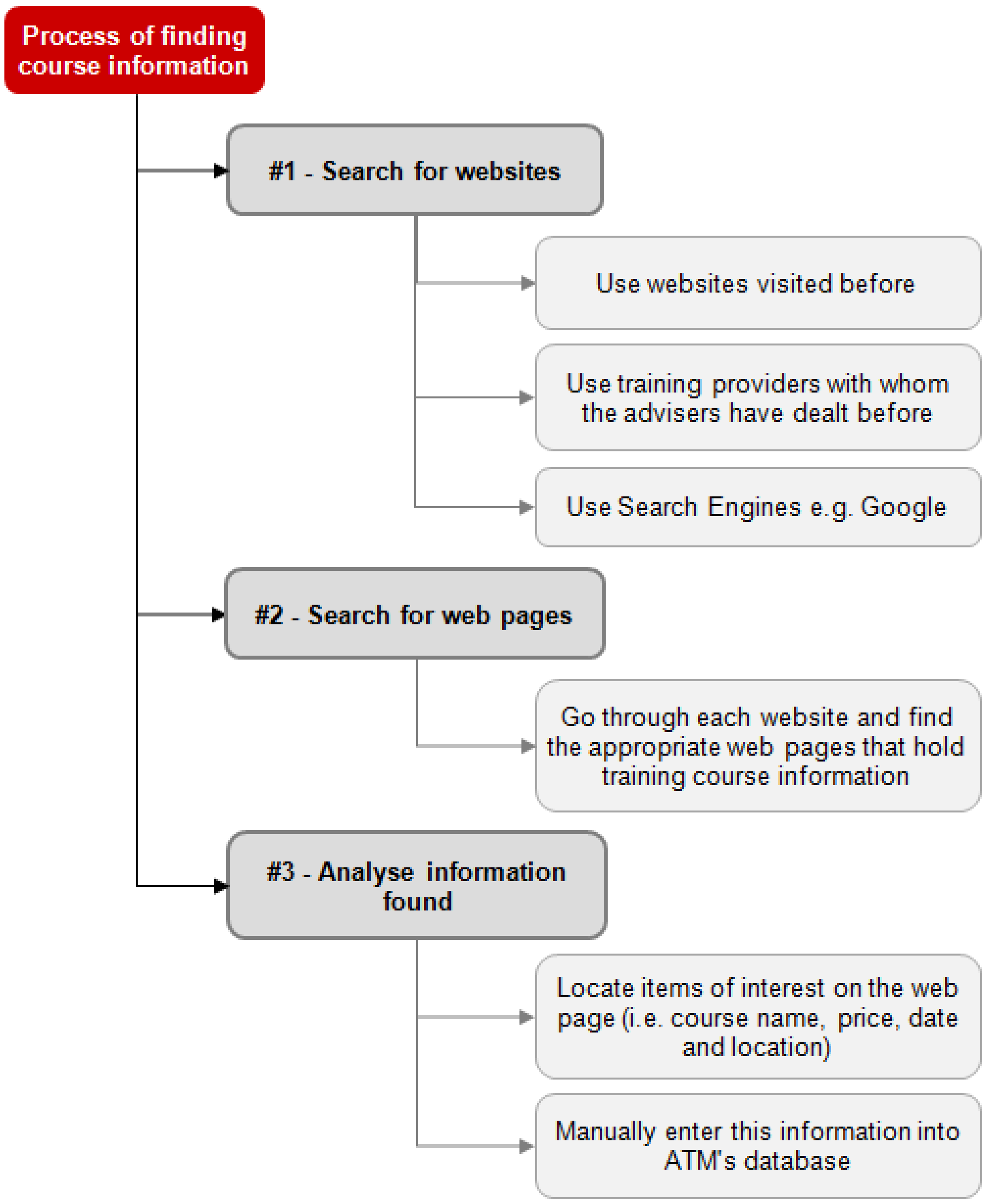

Process of finding course information
#1 - Search for websites
Use websites visited before
Use training providers with whom the advisers have dealt before
Use Search Engines e.g. Google
#2 - Search for web pages
Go through each website and find the appropriate web pages that hold training course information
#3 - Analyse information found
Locate items of interest on the web page (i.e. course name, price, date and location)
Manually enter this information into ATM's database

Figure 1.1: Typical Process at ATM

All three main tasks (#1, #2 and #3) are very demanding for ATM. Task #3 in particular is the most time consuming. Despite one of the employees working full time on updating the database, it is impossible to keep it constantly up-to-date, due to the limited life expectancy of some course information such as dates and prices and limitations in the accessibility of accurate course information on provider websites and/or printed literature.

According to an internal report at ATM by Greasley (2006):

- By Nov 2005, out of the 804 providers stored in the company's database, only 10% of these providers were being used.
- By April 2006, out of the 878 providers in the database, only 50 of these providers were in use. That is a disappointing 5.7% of all providers, which meant that ATM was potentially missing out on better deals from the remaining training providers.

The advisors have no simple method of knowing whether the courses in the database are up-to-date, thus, they rarely use the database at present. It is a general belief in the company that it is less risky to search for courses on the Web each time they are needed, rather than risk proposing an out-of-date course to an important client.

In summary, the problems the company is facing are:

- No clear process to follow → Chaotic
- Manual searching → Time consuming
- Manual data entry → Out of date quickly
- Manual database maintenance → Costly

The above show that ATM lacks the technology to guarantee reliability and accuracy of the information in the long term and their methods of capturing, organising and managing information waste time and resources, thus there is much need for an automated system, which will reduce human effort in finding courses, whilst increasing efficiency and accuracy.

1.5 AIMS AND OBJECTIVES

The main aim for this project is to develop a prototype system, which will automate the retrieval and extraction of training information from the web into the sponsoring company's database, guaranteeing an always up-to-date collection of training information, whilst keeping the user involvement to a minimum.

The specific objectives are to:

- review related work in the field of WIR and WIE.
- analyse ATM's business processes and project requirements.
- model a database to store the results from each stage of the system.
- automate the retrieval of web pages from the Web.
- automate the classification of the previously retrieved pages into *relevant* and *irrelevant* categories.
- automate the extraction of training course attributes from the *relevant* web pages into ATM's database.
- automate system execution to run at scheduled times.
- critically evaluate the project, reviewing the success and achievements and recommend further improvements.

Please note that the second objective was proposed with the sole purpose of helping to identify the existing problems at ATM in relation to the discovery and gathering of training course information from the web. This objective is not concerned with a detailed analysis of ATM as a whole, as this was not necessary for the achievement of the project aim.

1.6 STRUCTURE OF THE THESIS

This thesis is organised into the following chapters:

Chapter 1 (Introduction) described the background to the research, defined its aims and objectives, introduced the sponsoring company and gave justification for the research. The papers included in this thesis are also outlined here.
Chapter 2 (Background and Literature Review) discusses the relevant literature related to the project, concentrating on past and present research work in WIR and WIE.
Chapter 3 (Research Methodology) describes existing research methodologies and justifies the methodology chosen for this research.
Chapter 4 (The Research Undertaken) concentrates on the system design and implementation making use of different diagrams to explain various functionalities in the system.
Chapter 5 (Findings and Evaluation) describes and evaluates the results obtained from testing each component of this research. Evaluations of the main stages of the project, such as the aims and objectives, data samples and the overall approach used are also presented.
Chapter 6 (Conclusions and Recommendations) concludes by summarising the contributions of this research to existing theory and practice, presenting the impacts and implications of the research on the sponsoring company and wider industry and suggesting ideas for further improvements.

Appendices A to E contain additional support information to various aspects discussed in the thesis. References to this information are provided where appropriate in the body of the thesis.
Appendices F to J contain peer-reviewed papers published during the research period. These papers are an integral part of the thesis and thus, they should be read alongside the report.

1.7 SUMMARY OF PAPERS

The following table summarises the papers resulting from this research and included in this thesis. Details of each paper, such as the title, status and a small description are presented, together with the location of each paper in the appendices.

Table 1.1: Summary of Papers

Location	Title	Journal/Conference	Status	Description
Appendix F	Embarking on a Web Information Extraction Project	The 2007 UK Conference on Computational Intelligence, London	Published	This paper discusses our early efforts to build an automatic web extraction system. It reflects on the variety of technologies that can be used for this kind of project and introduces some of the issues that can be encountered along the way.

Appendix G	Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages	International Journal of Computer Science Issues	**Published**	This paper concentrates on the comparison of Neural Networks (NN), Naïve Bayes (NB) and Decision Tree (DT) classifiers for the automatic analysis and classification of attribute data from training web pages. The research discussed in this paper shows that our enhanced NB classifier not only outperforms the traditional NB classifier, but performs just as well, if not better than some more popular, rival techniques like NN and DT classifiers.
Appendix H	Genetic Evolution of Regular Expressions for the Automated Extraction of Course Names from the Web	The 2010 International Conference on Genetic and Evolutionary Methods, Las Vegas, U.S.A.	**Published**	This paper describes a novel approach to automating WIE through genetically evolved REs. Particularly, the attention is focused on the extraction of online information related to training course names/titles. The evolved REs are evaluated using a NB Network.
Appendix I	Domain-Independent Genotype to Phenotype Mapping through XML Rules	International Journal of Computer Science Issues	**Published**	This paper discusses an innovative approach to mapping Genotypes to Phenotypes through XML rules. It specifically concentrates on the mapping process using two very different domains – Regular Expressions (REs) and Software Program Statements. The paper shows that our Genotype-Phenotype system can be applied to any domain that requires the use of REs and it can be adapted to work for any other domain with minimum effort.
Appendix J	Genetic Evolution of 'Sorting' Programs through a Novel Genotype-Phenotype Mapping	The 2010 International Conference on Evolutionary Computation, Valencia, Spain	**Accepted for Publishing**	This paper presents an adaptable genetic evolutionary system, which includes an innovative approach to mapping genotypes to phenotypes through XML rules. Specifically, the paper concentrates on the evolution of "Sorting" programs. Experiments show that the evolutionary system is successful and can be adapted to work for challenging domains with minimum effort.

1.8 SUMMARY

This chapter has provided a general introduction to the research project and has justified the need for this research. The aims and objectives were also listed together with an introduction to the project sponsor. The following chapter analyses past and present research work related to WIR and WIE as a background to the research.

2 BACKGROUND AND LITERATURE REVIEW

2.1 INTRODUCTION

As previously mentioned, this project comprises two main research areas: Web Information Retrieval (WIR) and Web Information Extraction (WIE). These may sound similar, however they are distinctly different, yet complementary to each other. WIR retrieves relevant documents from collections, whereas WIE extracts relevant information from documents (Eikvil, 1999). In other words, the output from WIR serves as input to a WIE process. WIR, in itself, consists of two distinct areas: Web Searching and Web Classification. Both WIR and WIE are important to this research, as one needs to locate the valid sources before extracting the appropriate information from them.

This chapter explores the strengths and weaknesses of each of these three research areas (Web Searching – section 2.2; Web Classification – section 2.3; Web Extraction – section 2.4), by surveying past and present work in these fields. Gaps in existing literature are highlighted in order to demonstrate the contribution of this research. References to the papers published during this research and other documents included in the appendices are provided where necessary to avoid repetition.

2.2 WEB SEARCHING

Web searching has been highly popular since the expansion of the Web in the 1990s. More and more individuals and organisations are uploading their data and services online for internet users to access. According to Miniwatts Marketing Group (2009), the growth of web users between years 2000 and 2009 was just below 400% and by December 2009, there were over 1.8 billion internet users from over 235 countries. A study by Maurice de Kunder (2006; 2010) at Tilburg University calculated that there exist at least 19 billion indexed web pages on the web (as of May 24, 2010). However, it is not just the number of users and overall websites that keeps growing; research has shown that websites are also growing in size (O'Neil et al., 2003, Ntoulas et al., 2004), with public sites containing on average 441 pages, which is an increase of 28 pages from year 2001.

These figures show that too many web pages exist to be able to find information without the help of advanced technological tools. One such tool includes what is known as a Search Engine. Search engines can be described as software programs, which search for information given a query and return a list of web pages that provide that information. For effective searching, web pages must first be discovered and indexed. This task is performed by a program called a Crawler or a Spider. Crawling usually begins with an initial set of links (URLs) also known as "seed URLs". The Crawler finds and retrieves all new links within the content of the seed URLs and stores them in an indexed repository. The indexes aid the quick retrieval of the links at later stages. Each unvisited URL in the repository is searched for additional links. Crawling terminates when all the links have been visited.

2.2.1 SEARCH ENGINE REQUIREMENTS

Search engines work by crawling the Web, indexing all the pages found and ranking them according to their relevance in relation to the user's query. They have three main components (Brin & Page, 1998): a crawler, an indexer and a query server (Figure 2.1). The *crawler* is the part of the system which collects web pages from the Web. The *indexer* examines these pages and stores them as collections of words and structures. The *query server* gets the query from the internet user and returns the relevant result(s) for that query after analysing the stored collections.

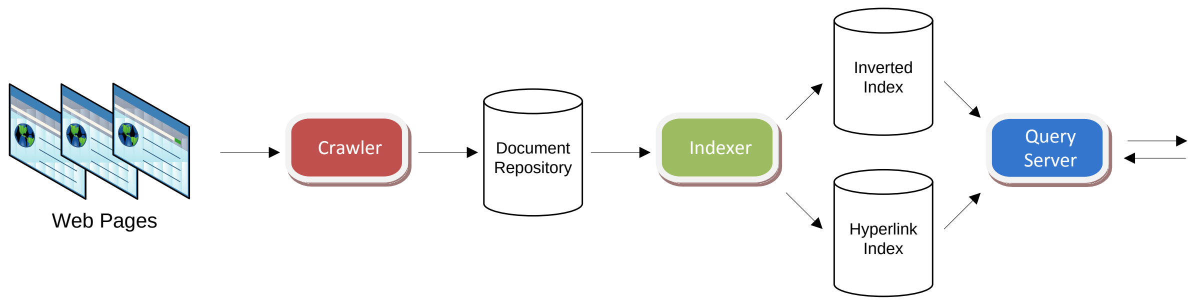


Figure 2.1: Search Engine Components

The following lists some important requirements for designing a successful search engine (Brin & Page, 1998):

- All links found should be visited to obtain all other links present in them. This is simple for links of the nature <a href="…">Link Caption </a>, however there are links in other formats, such as JavaScript, which are more difficult to retrieve.
- Broken links are not to be stored, as they have no use and only waste database space.
- The same link should not be crawled twice, as this has consequences on speed and general performance.
- The crawler should not cause delays or other problems to the web pages it is crawling. Similarly to (Cho et al., 1998), this research deals with this issue by adapting the crawler to only visit the same server every 1000 links. This helps to respect the client servers (see Appendix A for a discussion of the Copyright Infringement risks involved with WIR and WIE technologies)
- Care should be taken to adhere to the rules set by the sites being crawled. Some websites include a robot.txt file, which gives instructions to crawlers, guiding them through the privacy policies of the site. The following examples show three different cases of search engines: being allowed to crawl the whole site (1), not being allowed to crawl any part of the site (2) and not being allowed to crawl certain areas of the site (3):

```
User-agent: *                                                    (1)
Disallow:
```

```
User-agent: *                                                    (2)
Disallow: /
```

```
User-agent: *                                                    (3)
Disallow: /images
Disallow: /private
```

- Search engines need to be able to deal with dynamic pages and media features such as images etc. The media features are not relevant to this research as the information required is only in text format.
- Searching must be able to deal with vast amounts of data, due to the large size of the Web.

- It is important for traditional search engines to present their findings in appropriate user interfaces, to help users access the results quickly and easily. The search engine for this research does not require a user interface, because ATM wants to access the results using their CRM software package.
- Search engines need to be able to deal with phrases including Boolean logic i.e. AND, OR, NOT operators to incorporate various queries together. This is also not applicable to this research, for the same reasons as the previous point.
- Care should be taken to allow for multi-language queries. This requirement is not met by many crawlers and it is not necessary for this research either, as ATM is only interested in courses held within the UK, thus English is the only language required.
- Regional settings should also be taken into account, so results are localised if appropriate, e.g. searching for information from UK sites only.
- Sub-searching is another feature that not all crawlers achieve. This involves the ability to refine queries by removing/adding additional terms to them. This functionality is not a requirement for ATM, as users are not to be involved in the searching process.

The search engine developed in this research satisfies all the points in the above list, except those identified as irrelevant to the project requirements.

2.2.2 STATISTICAL ANALYSIS

One would think that because search engines analyse the same set of web pages, the same web page rankings would be returned for the same query. This is however not the case. Overlap studies show that approximately 50% of the indexed pages in any search engine database exist only in that database (UC Berkeley, 2006).

A research study conducted by the Queensland University of Technology and Pennsylvania State University (2007) also showed surprising results when comparing the overlap of first page results from the four most popular search engines: Google (http://www.google.com or http://www.google.co.uk), Yahoo! (http://www.yahoo.com), Windows Live a.k.a. MSN Search (http://uk.msn.com) and Ask a.k.a. Ask Jeeves (http://www.ask.com). The study revealed that out of the 776,435 results returned:

- The four search engines shared only 0.6% of the overall results.
- The overall number of results returned exclusively by one search engine alone was 88.3%.
- Any two search engines overlapped by 8.9% of the overall results
- Any three of the search engines overlapped by 2.2% of the overall results.

These figures show that submitting a query on any of the existing search engines does not mean getting the highest ranked web page i.e. the most relevant page answering the query. This is confirmed by further statistics produced by the above research. The study shows that by searching only Google a searcher might miss 72.7% of the Web's best first page search results. By searching only Yahoo!, a searcher might miss 69.2% of the best results. An MSN search would cause a searcher to miss 69.9% of the best results and searching only Ask Jeeves would miss out 73% of the best first page search results. More than a fourth of the time (26.1%) the above four search engines completely disagreed on the top five, non-sponsored search results.

Figure 2.2 and Figure 2.3 show a comparison of the top 100 results obtained from Google and Yahoo on the 21st of April, 2010, when searching for "plumbing courses Leicestershire" and "management courses Leicestershire" respectively. The circles represent the exact list of web pages retrieved when using Google and Yahoo. The empty circles represent web pages unique to each engine, whereas the filled circles represent the web pages that both search engines displayed in their top 100 list. The figures clearly show that the two search engines disagree not only on the web pages that make the list, but also on the position of each page on the list i.e. their ranking. Indeed, the two search engines did not grant the same ranking to any of the web pages on their lists.

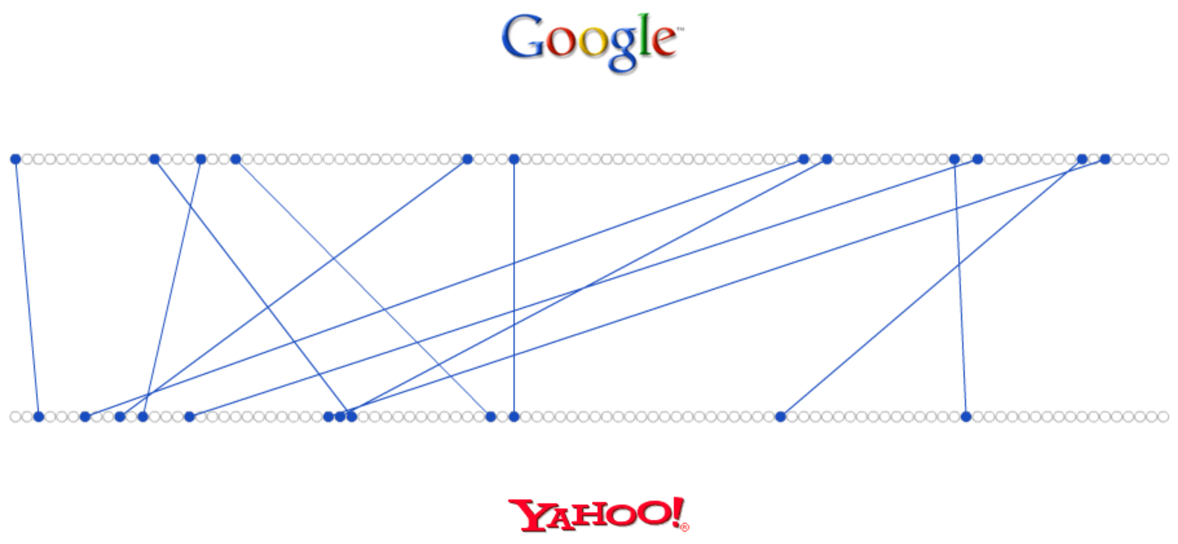


Figure 2.2: Google vs. Yahoo – Plumbing Courses, Leicestershire

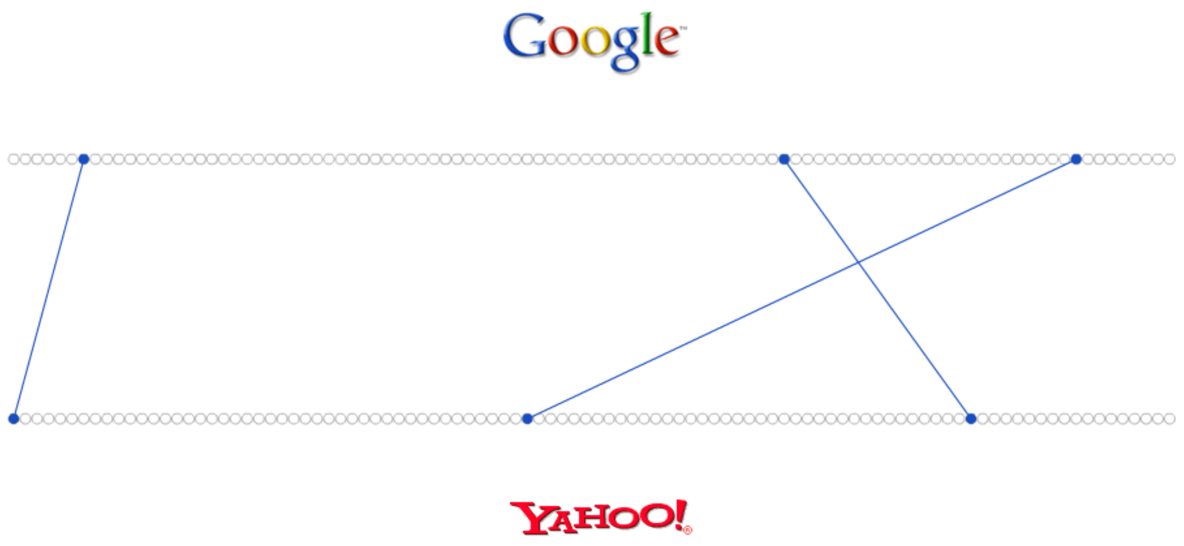


Figure 2.3: Google vs. Yahoo – Management Courses, Leicestershire

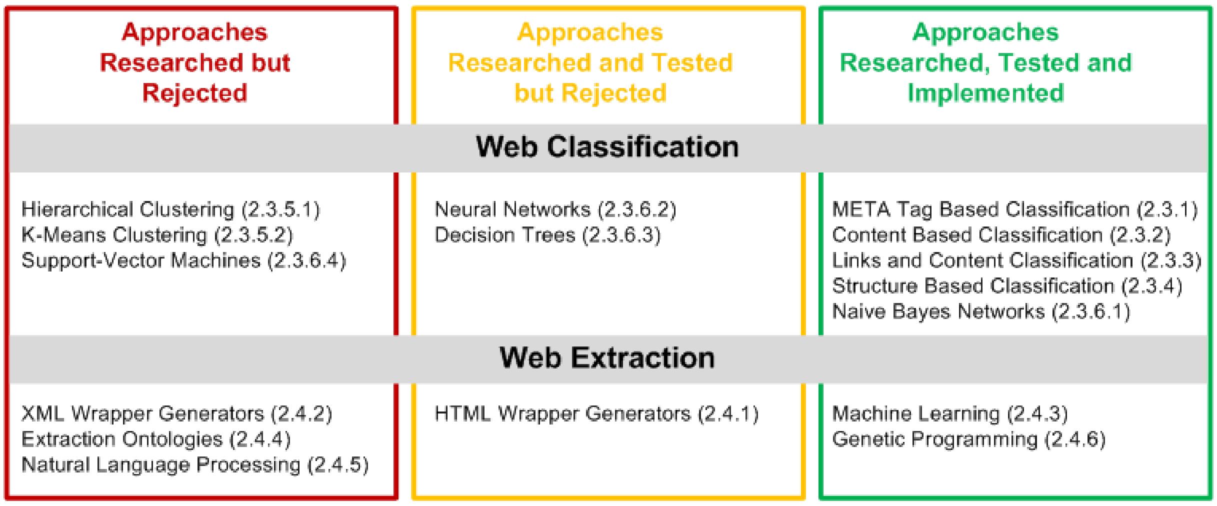
These results show that search engines view the Web quite differently from each other. Paolillo (2005) argued that this is due to search engines producing what is known as a "snowball sample". New web pages are discovered by visiting all the links found in a previous web page. Thus, the starting point of each "snowball", i.e. the initial list of "seed

URLs" used by the crawler, is vital. Gerrand (2007) carried out experiments on a number of search engines and reported that one can have no confidence that search engines look at the same part of the web, or that they crawl more than a very small fraction of the public web (16%, according to Lawrence & Giles, 1999). Other researchers expressed their doubts on the consistency of the search methodologies used by search engines, given that their ranking algorithms are never published (Lawrence & Giles, 1999; Van Couvering, 2007).

These differences in crawling coverage mean that users should query multiple search engines for optimal results (Yadav et al., 2009). However, even after discovering all the links that may contain the answer to one's query, the user still needs to visit each link and manually locate, verify and extract the relevant information on each page. This is impractical and time consuming, thus, automated retrieval and extraction systems are essential, particularly to people or organisations who search the web for information regularly, like ATM.

The system developed in this research is such a system, because although it includes a web crawling module, it goes beyond the capabilities of search engines and actually provides users with the exact information they require, in this case training course information. Chapter 5 discusses the results of the crawler in comparison with other crawling solutions available.

The following sections in this chapter concentrate on the different techniques that can be applied to web classification and web extraction (Figure 2.4). Furthermore, some commercial WIE solutions are discussed and their limitations are disclosed.



Approaches Researched but Rejected	Approaches Researched and Tested but Rejected	Approaches Researched, Tested and Implemented
	Web Classification	
Hierarchical Clustering (2.3.5.1) K-Means Clustering (2.3.5.2) Support-Vector Machines (2.3.6.4)	Neural Networks (2.3.6.2) Decision Trees (2.3.6.3)	META Tag Based Classification (2.3.1) Content Based Classification (2.3.2) Links and Content Classification (2.3.3) Structure Based Classification (2.3.4) Naive Bayes Networks (2.3.6.1)
	Web Extraction	
XML Wrapper Generators (2.4.2) Extraction Ontologies (2.4.4) Natural Language Processing (2.4.5)	HTML Wrapper Generators (2.4.1)	Machine Learning (2.4.3) Genetic Programming (2.4.6)

Figure 2.4: Summary of Approaches Researched

2.3 WEB CLASSIFICATION

The crawling element of WIR systems, although challenging, is the outcome of well designed Object Oriented or Procedural Programming. The classification part however is where the 'intelligence' is injected, as this is the part that needs to logically analyse the data and categorise the information based on the 'knowledge' accumulated throughout the classification process. The following concentrates on the different techniques for document classification/categorisation.

Based on the web page areas on which they concentrate, web classification techniques can be grouped in the following categories:

- META tags based classification (Pierre, 2001; Yang et al., 2004; Martin & Shen, 2006; Mohammadian, 2008; Romero & Ventura, 2007)
- Content based classification (Zhang and Ling, 2001; Wang et al., 2003, Wang et al. 2006; Addin et al., 2007)
- Links and text content based classification (Lu & Getoor, 2003; Calado et al., 2006; Indra et al., 2007; Baykan, 2008)
- Structure based classification (Elsas & Efron, 2000; Asirvatham et al., 2001; Grosky & Deshpande, 2004; Shah & Deshpande, 2008)

2.3.1 META TAGS BASED CLASSIFICATION

META tags are optional HTML structures used to describe different characteristics such as overall web page description and main keywords describing the content of the web page. These tags are placed in the <head> section of the page and look as follows: <META name="keywords-used", content="overall-description-of-page">.

META tags are useful in helping retrieval programs to quickly 'understand' the nature of websites. However, with META tags being optional features, many websites choose not to include them. Another difficulty, which is more concerning, is that some sites are described by keywords, which do not reflect the real content of these websites. Instead, highly scored keywords are used, in the hope that search engines will be tricked into classifying these sites differently and ranking them higher. This can be very misleading.

Pierre (2001) used metadata to automatically classify websites into industry categories. Some of the tests performed concluded that text found in META tags resulted in better features than text inside the BODY tags. A targeted crawler was therefore designed to look for metadata first and only in their absence degrade to other text. This may sort out the optionality problem, however it does not consider the use of inconsistent metadata in describing websites.

Martin and Shen (2006) published their concerns about the inconsistency of metadata between different sources within the same category, which could affect classification results. They used two online movie databases - "Rotten Tomatoes" (http://rottentomatoes.com) and the "Internet Movie Database" (http://www.imdb.com) - to test their method of discovering attribute and category similarities in different sources through the metadata derived from these sources. The research used an efficient algorithm to compute word similarities (Martin and Azmi-Murad, 2005), whilst using fuzzy association rules to allow for the possibility of noise in the data. Mohammadian (2008) also used the metadata values and fuzzy rules to aid the classification of security data.

META tags alone are considered insufficient in producing accurate classifications of sites (Yang et al. 2004; Romero & Ventura, 2007). However, used together with other components, they can improve accuracy. The research presented in this thesis examines the META tag information from web pages together with other important elements such as the page title, link information and the general content of each page.

2.3.2 CONTENT BASED CLASSIFICATION

This type of classification is based purely on the content of web pages and it is not concerned with information from links, META tags, formatting etc.

These techniques generally work by indexing websites i.e. extracting all the terms used in web pages and storing them in a database or file. Once the indexing is completed, the web page classification and ranking is performed, which normally includes the following metrics:

- Term frequency – It counts the number of times each term is found in each document, to determine the relevance of each word in that document.
- Location – It assumes that the main topic of a web page becomes apparent closer to the top of the page, thus location of terms is important.

Another possible metric is linked to multiple word queries. In this case, the distance between all the words used in the query needs to be calculated for all the words in each document and the smaller the distance, the higher the page ranking will be.

Many techniques such as Neural Networks, Bayesian Filtering etc. are based on keyword analysis. These will be discussed in detail in section 2.3.6.

2.3.3 LINKS AND CONTENT BASED CLASSIFICATION

This type of classification bases its findings not only on the content analysis (as described above), but also on the information provided by links or URLs. Links may sometimes give information on the web page by describing its content through the URL itself and anchor information e.g. <a href=http://www.link.domain/**page-desc**>**Link Title**</a>

Some researchers (Lu & Getoor, 2003; Indra et al., 2007; Baykan, 2008) have dedicated their research to proving that URLs have enough features to aid web classification without needing to analyse any other attributes. Their experiments showed that their technique proved more useful than some search engines. Calado et al. (2003) also maintained that URL-based classifiers perform much better than content-based classifiers, due to the unreliability of the content of web pages. However, further experiments (Calado et al., 2006) showed that classifiers based on the combination of links and page content increased the accuracy of the results from 86% to 90%.

Other researchers (Attardi et al., 1999; Zhu et al., 2007) argue that basing the classifier on links alone is too limiting. For example some websites generate their pages dynamically, in which case the same URL is used for multiple pages. There are also web pages, which use vague links that conceal the content of the page e.g. if one did not know that IEEExplore manages research papers, the following link would be of little help in categorising this page: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5376433.

Attardi et al. (1999) merged the above idea of analysing links with the analysis of different page structures (discussed below) such as the page title, headings, formatting etc. The researchers believed that combining different classification techniques improves efficiency, because this way the best features of each technique are merged together to increase the recall and precision values of the retrieval process.

The research presented in this thesis follows the same belief. The classifier developed takes into consideration various web features such as META tags, page title, links and the content of web pages. This combination has been successful, as discussed in chapter 5.

2.3.4 STRUCTURE BASED CLASSIFICATION

Structure based classifications are based on analysing the structure of web pages rather than just the text content or links on the page. The popularity of this technique is based on the fact that humans have no problem in mentally categorising documents before even reading their content. They are able to do so by using visual clues such as headings, subheadings etc. Humans also take into account the positioning of text and links to realise their importance e.g. information closer to the top of the page is sometimes more important than the information given elsewhere on the page, in order to capture the attention of users quicker. Images are also a good indicator of what a document could be describing.

These observations made researchers realise that something similar could be applied to WIR systems in order to achieve similar results when classifying documents (Elsas & Efron, 2000; Asirvatham, 2001; Shah & Deshpande, 2008). The web page structures analysed include:

- Headings, subheadings etc.
- Page title
- Tables and other blocks of organised data
- Images
- Data styling
  - Font size; weight (e.g. bold)
  - Upper-case vs. lower-case letters
- Indentation
- Term / section positioning on the page etc.

Asirvatham et al. (2001) proposed a method for automating the categorisation of web pages, based on the contribution of each feature to the web page. The weights of these contributions were calculated and compared and the final weights were then used to classify all documents into broad categories such as: Information, Research or Personal Pages. In contrast to the above, Grosky & Deshpande (2004) focused their research on using the HTML tag hierarchies created from each web page. At the heart of their idea, lay the concept of dividing each web page into N x M non-overlapping blocks and finding the tag covering the maximum area for each block. The spatial arrangements of the centre co-ordinates of these blocks were calculated using Delaunay triangulation and feature point histograms. The intersection of these histograms resulted in the web page descriptors, which were then used in the classification of the document. This was tested against Individualised and Genre categories and although it performed adequately on the Genre categories, it failed on many of the Individualised ones.

Elsas & Efron, (2000) used HTML tag metrics to classify web pages from statistical websites by exploring web page features such as tables, indexes, table of contents as well as the textual content of the page. Similarly, Shah & Deshpande (2008) used structural information to aid the classification of web documents without the need for negative example training.

Structure based classification is popular because, if done correctly, it can increase the accuracy of the classification process. However, the Web is unpredictable and by no means standardised (Xhemali et al., 2007), thus much care needs to be taken to avoid producing unstable results.

The above sections investigated the different classification approaches in relation to the web page areas, on which they concentrate. In relation to the actual techniques used to achieve web classification, these can be grouped in the following main categories:

- Unsupervised Classification: Clustering
- Supervised Classification: Neural Networks (NNs), Decision Trees (DTs), Support-Vector Machines (SVM), Bayesian Categorisation.

2.3.5 UNSUPERVISED CLASSIFICATION

Unsupervised learning techniques differ from supervised ones, as they do not need to train systems beforehand. Instead, they rely on finding patterns within the data provided in order to make intelligent decisions.

Clustering is one of the main unsupervised learning techniques used in web classification. Clustering can be further categorised into Hierarchical and K-Means Clustering:

2.3.5.1 Hierarchical Clustering

The key point in Clustering is that no one piece of information provided is the answer to a query. Clustering does not make predictions about specific queries such as 'Who is likely to commit credit card fraud'. Clusters aim to analyse the data by finding definite patterns that exist in the dataset (Segaran, 2007). Some of the situations that can be dealt with using Clustering are: 'Determining shelf locations of items which are commonly purchased together', 'Retrieving and cataloguing online blogs' etc.

Figure 2.5 shows that hierarchical clustering works by finding the two closest data items and merging them into one cluster. The newly created cluster is then merged with the next closest data item and the process continues until all items are merged into one big cluster. This can then be converted to a tree-like structure called a Dendogram or Taxonomy as shown in the right hand side of the above figure.

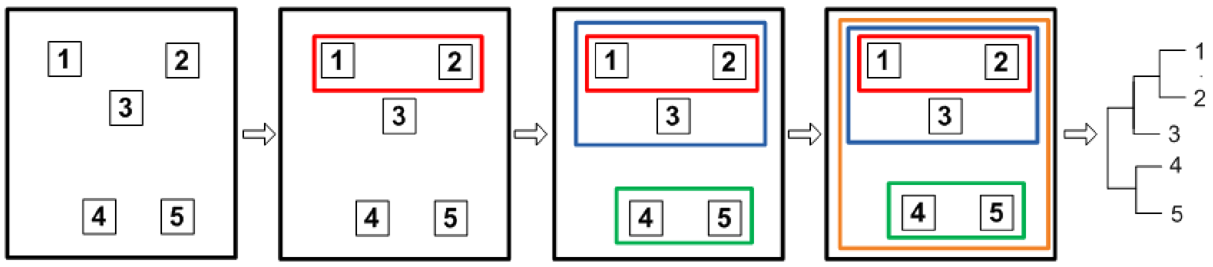


Figure 2.5: Hierarchical Clustering

Data may be clustered based on the 'Frequent Word Sequences' and the 'Frequent Word Meaning Sequences' (Li et al., 2008), where the *word meaning* represents the semantics of a certain word. Some research works try to reduce the dimensions of the data in order to build smaller generalised suffix trees (Tata et al., 2004; Mocian, H., 2009).

2.3.5.2 K-Means Clustering

K-Means Clustering differs from Hierarchical Clustering, because the data is separated into very distinct groups, instead of being merged into one cluster (Segaran, 2007).
Figure 2.6 shows the process of classification using K-Means Clustering. The black filled

circles are called centroids and they are initially placed randomly amongst the data through an initial guess of the centre locations of each data group. Each data item is then linked to the centroid closest to it; in this case items 1 and 2 are closest to the upper centroid, whereas items 3, 4 and 5 are closest to the bottom one. Once such links are determined, the centroids are moved to the average locations of the items assigned to them. However, at this point the situation may change. The third picture shows that item 3 is now closer to the upper centroid than the bottom one, thus this item is re-assigned and the centroids are moved again to the average locations of their new assigned items. This process is repeated until all data is separated into clear clusters.

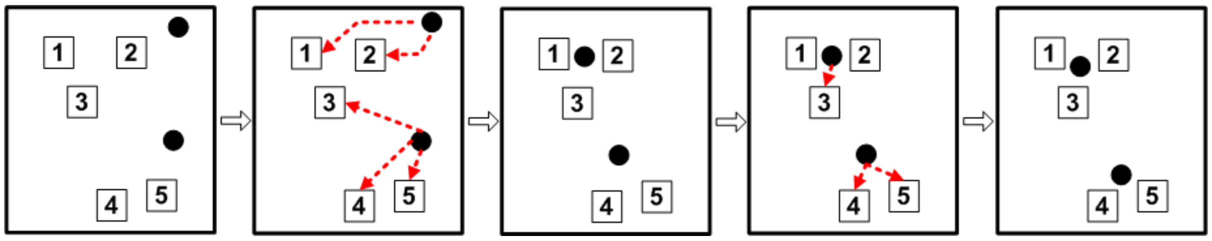


Figure 2.6: K-Means Clustering

K-means requires the user to predefine in advance the number of clusters they want out of the clustering process (Peters, 2006). Depending on the dataset size, this may be impractical as sometimes the number of resulting clusters is not known in advance. Another difficulty with K-Means Clustering is deciding on the number of centroids to be placed at the start of the clustering process, as well as deciding on their random positioning. Depending on this positioning, the data configurations may be very different, which may affect not only the speed of the clustering process but also the end results.

Due to these problems, it comes as no surprise that K-Means Clustering is usually applied as a secondary phase after other clustering methods have attempted the classification first (Segaran, 2007).

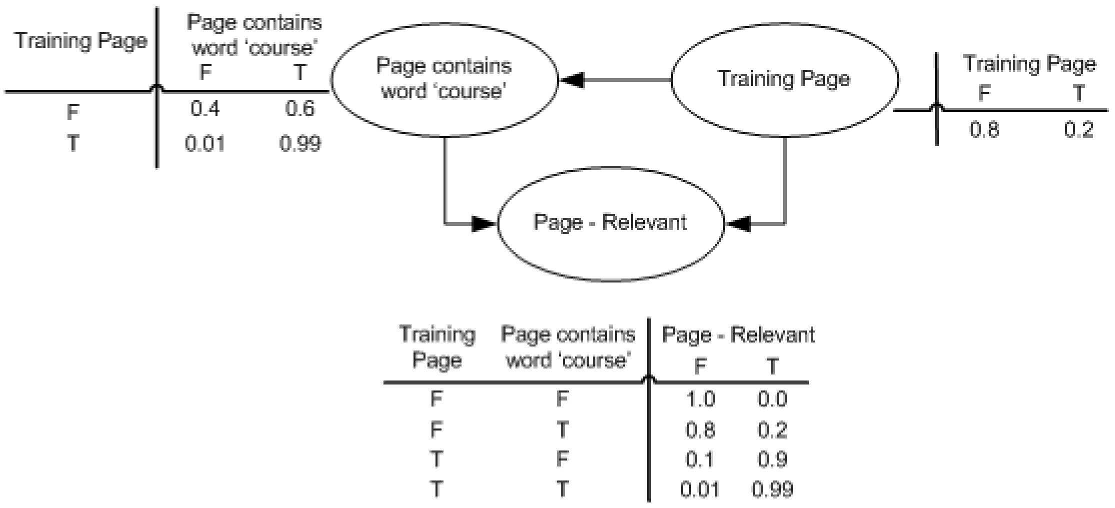
The above problems have determined that clustering techniques are too process intensive and expensive, thus they were not used in this research.

2.3.6 SUPERVISED CLASSIFICATION

Supervised learning used in web page classification is based on training systems to ‘learn’ the features that make a document fit into one category or another. Training is done using labelled examples which may either be positive examples only (Denis et al., 2003; Calvo et al., 2007) or a combination of positive and negative examples (Zhang & Ling, 2001; Wang et al., 2003). In this research, training is achieved through a mixed set of *relevant* web pages (i.e. pages that contain training course information) and *irrelevant* pages (see section 4.3.4). Using a generous number of training documents for each category could improve the system’s accuracy in the classification of new web pages. This is not a mandatory requirement, as supervised classifiers can be applied to new web pages after minimal training; however the latter may produce questionable results. The following discusses some of the supervised techniques considered for this research.

2.3.6.1 Naïve Bayes Networks

Bayes Networks or Bayesian Classifiers are graphical models, which analyse relationships between variables of interest and deduce any uncertainties in the subject domain. Each Bayes Network consists of the variables being analysed, the links connecting the variables, where the link directions show the dependencies between the variables, and a conditional probability table which shows the set of probabilities associated with each variable. Figure 2.7 shows a basic representation of a Bayes Network, whereby the fact that a page is a 'Training Page' has an impact not only on the page containing the word 'course' but also on the page being *relevant*. Variable 'Page contains word 'course'' also has an impact on the page being relevant. The conditional probability tables in Figure 2.7 show that if a page is a Training Page and it contains the word 'course', then there is a 99% chance that this page is *relevant*.



Training Page	Page contains word 'course' F	T
F	0.4	0.6
T	0.01	0.99

Training Page F	T
0.8	0.2

Training Page	Page contains word 'course'	Page - Relevant F	T
F	F	1.0	0.0
F	T	0.8	0.2
T	F	0.1	0.9
T	T	0.01	0.99

Figure 2.7: Basic Bayesian Network for Relevance Detection in Web Pages

There exist special cases of Bayes Networks, called Naïve Bayes (NB) Networks, which are popular in machine learning applications due to their simplicity in allowing each attribute to contribute towards the final decision equally and independently from other attributes (Zhang and Ling, 2001). This simplicity equates to computational efficiency, which makes NB techniques attractive and suitable for many domains, particularly those with a large number of attributes (Wang et al., 2003).

However, the very same thing that makes them popular, is also the reason given by some researchers, who consider this approach to be weak (Rennie et al., 2003). The conditional independence assumption is strong, and makes NB-based systems incapable of using two or more pieces of evidence together, however, used in appropriate domains, they offer quick training, fast data analysis and decision making, as well as straightforward interpretation of test results. There is some research (Zhang, 2002; Langseth & Nielsen, 2006) trying to relax the conditional independence assumption by introducing latent variables in their tree-shaped or hierarchical NB classifiers.

In the research presented in this thesis, however, an analysis of a number of training web pages has shown that the domain for this research can be analysed using NB classifiers (details of this technique in relation to the research are explained in chapter 4).

Various researchers have attempted to enhance the standard NB rule or use it in collaboration with other techniques. Addin et al. (2007) coupled a NB classifier with K-Means clustering to simulate damage detection in engineering materials. NBTree (Wang et al. 2006) induced a hybrid of NB and DTs by using the Bayes rule to construct the decision tree. The research in this thesis uses NB networks alongside Genetic Programming to evolve WIE solutions.

Other researchers (Denis et al., 2003; Calvo et al., 2007) modified their NB classifiers to learn from positive and unlabeled examples. Their assumption was that finding negative examples is very difficult for certain domains, particularly in the medical industry (Yu et al., 2004). Finding negative examples for the training courses domain, however, is not at all difficult, as this simply corresponds to finding web pages that do not contain training course information in them, thus the above is not an issue for this research.

2.3.6.2 Neural Networks

Neural Networks (NNs) are powerful techniques of representing complex relationships between inputs and outputs. They were inspired by the neural structure of the brain, where a neuron is a cell in the brain responsible for the collection, processing and distribution of electrical signals (Russell & Norvig, 2003). Much is still unknown about the full power of the brain however, thus these networks are only basic representations of the real neurons and their interconnections. NNs are composed of nodes or units connected by direct links, which serve to propagate activating signals between these units. This is done in two stages:

- Each unit computes the total weight (TW) of its inputs:

  $$TW_i = \sum_{j=0}^{n} W_{j,i}\, a_j$$

  where n is the number of inputs; $W_{j,i}$ is the weight associated with units j and i, and a_j is the activation signal.

- An Activation Function g is then applied to the above result to derive the output (a_i):

  $$a_i = g(TW_i) = g\left(\sum_{j=0}^{n} W_{j,i}\, a_j\right)$$

NNs are much more complicated than Bayes networks. They can manage dependency amongst data, due to the multiple layers feeding off each other.

There exist various types of NNs. One common type, commonly used in classifications, is the Multilayer Perceptron Network (MPN), which includes one or more hidden layers in between the input and output layers. All layers communicate with each other via synapses, which have an associated weight related to them. This weight shows the amount of influence one neuron has on another neuron in the network. The weights in the hidden layers may be changed as the NN expands its knowledge.
Note that, in most situations, there is no way to determine the best number of hidden layers without training several networks and estimating the generalisation error of each. The number

of training cases, the amount of noise in the target data and the complexity of the overall function are amongst the factors that would affect the number of hidden layers and the number of neurons or midnodes in these layers. Some ‘rules of thumb’ for determining the best number of midnodes have been published, e.g.: Berry and Linoff (1997) suggested that the number of midnodes in the hidden layers should not be more than twice the number of inputs; Boger and Guterman (1997) stated that there should be as many midnodes in the hidden layers as dimensions needed to capture 70-90% of the variance of the input data set. A more recent, standard configuration of NNs is that the number of midnodes in the hidden layer is the same as the number of inputs (Russell & Norvig, 2003). The latter was used in the experiments in this thesis, as it fits well with the complexity of the domain (see section 5.4.2).

One way of training neural networks is by using the Backpropagation algorithm. This algorithm updates the weights of the synapses for each new training example in order to reflect the new ‘knowledge’ gained. It is called Backpropagation because it moves backwards through the network adjusting the weights accordingly. The adjustment needs to be done slowly, because otherwise the network may overcompensate when it is trained with noisy or uncertain data and make incorrect classifications.

Crestani (1993) found the capabilities of the Backpropagation algorithm very suitable for building an Adaptive IR System (AIRS) in the form of a sub-symbolic knowledge representation. AIRS modifies the user’s query by using the knowledge acquired during the training phase. It is therefore possible for the adapted query to retrieve more documents than the original query. However, the adaptation process may sometimes give more importance to the knowledge ‘learnt’ from the training stage, thus changing the original query to a point where it starts losing its initial keywords.

A much later research project (Chau & Chen, 2007) used a Feedforward/Backpropagation Neural Network and a Support-Vector Machine (SVM) for better performance in filtering web documents, based on the analysis of their content and structure. The content of each web page was analysed together with the content of its neighbouring pages. The neighbouring pages considered were pages that had links pointing to the original page; pages pointed to by the original page and pages, which were pointed to by any of the parents of the original page. The content and context analysis resulted in a set of 14-feature scores for each page. These scores were then used as input nodes to the Neural Network. This same set of feature scores was also used by the SVM. This research believed that using two powerful techniques may drastically improve classification, however they did not combine the two techniques to create one more sophisticated one. They simply used them one after the other on the same set of data, which meant that the system took much longer to produce results.

The main advantage of NNs is their ability to handle complex data, dealing with single pieces of information as well as multiple, dependent attributes. NNs manage this complexity without compromising on performance efficiency. They can be trained incrementally and they do not require much space to store the training models as these are only numerical values representing the weights of the synapses (Segaran, 2007).

One of the biggest disadvantages of NNs however, is that they are very complex and they can be enormous for certain domains, containing a large number of nodes and synapses. Some researchers have managed to convert NNs into sets of rules in order to discover what the NN has learnt (Towell & Shavlik, 1993; Fletcher & Hinde, 1994), however many other researchers still refer to NNs as a ‘black box’ approach (Tu, 1996; Tal, 2003; Segaran, 2007), due to the difficulty in understanding the decision making process of the NN, which can lead to not understanding whether or not testing has succeeded.

2.3.6.3 Decision Trees

Decision Trees (DTs) are decision support tools, which build models of behaviour for various situations. This is done through graphs showing the potential routes to solving a problem and the possible consequences for each route. Figure 2.8 shows the structure of a typical decision tree. For example, in this research a question could be ‘Is feature x relevant?’, possible answers would be ‘Yes’ or ‘No’ and possible conclusions or consequences would be the predicted categories i.e. *relevant* or *irrelevant* for the web page incorporating most or all relevant features.

Decision trees are easy to interpret, as graphical means are simpler to follow than pure explanatory text. Training a decision tree classifier is however quite complex. This is because each chosen question or attribute needs to be the most optimal choice, in order for it to divide the data into unmixed groups. The ideal division would produce data that is all of the same kind in each group e.g. ‘all positive’ or ‘all negative’. Creating such clean divisions, especially in large data sets, is very difficult (Russell & Norvig, 2003).

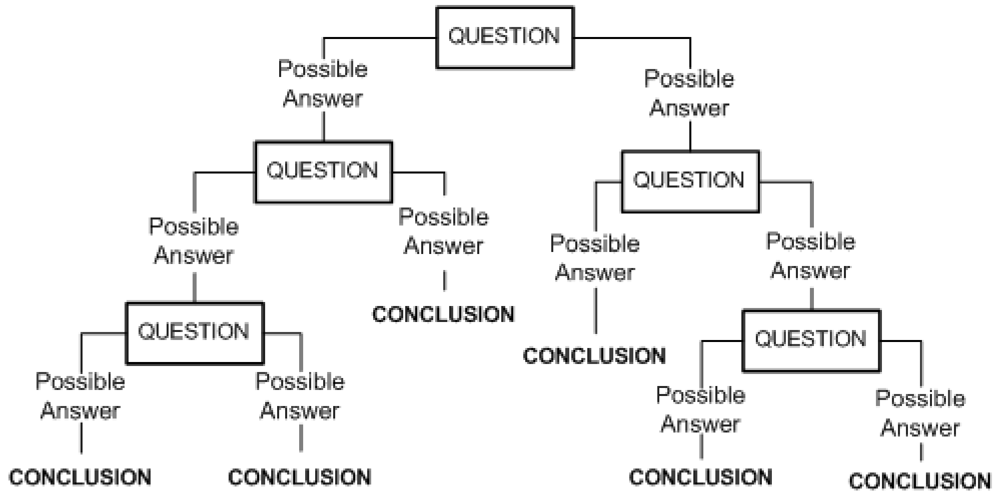


Figure 2.8: Decision Tree Structure

Entropy is a concept used to determine the amount of disorder in a set (S) of n examples, and as such it helps to measure the accuracy of the data division.

$$Entropy(S) = -K \sum_{i=1}^{n} P_i * \log_2(P_i)$$

where:

P_i is the proportion of S belonging to class i
K is a constant dependent on the choice of measuring units

Low entropy means that the division has produced mostly homogeneous data and Entropy = 0 means that the division has produced perfect results.

For each possible division the Information Gain for that division needs to be calculated to determine the dividing variable. The following calculates the Information Gain of attribute A, relative to a collection of examples in set S:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values\ (A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

For example, for a DT that has a root question or node split into two other subset nodes, the Information Gain would be calculated as follows:

$$Gain = Entropy(S) - \ Weight_1 * Entropy\ (Subset_1) - \ Weight_2 * Entropy\ (Subset_2)$$

where:

$$Weight_1 = \frac{Count(Subset_1)}{Count(S)}$$

$$Weight_2 = \frac{Count(Subset_2)}{Count(S)}$$

Maximising the Information Gain helps to create the right nodes in the decision tree.

WebClass (Hu et al., 2000) is a web classification application designed to search the homepages of geographically distributed groups of people, who share common interests. This application takes into consideration both the content of web pages and their HTML structures. It modifies the standard decision tree approach by associating the tree root node with only the keywords found, depth-one nodes with descriptions and depth-two nodes with the hyperlinks found. The system however, only achieved 73% of the accuracy achieved by manual classification, thus more research had to be done on this system.

The second version of WebClass (Ceci & Malerba, 2003) implemented various classification models such as: Bayes networks, Decision Trees, K-Means clustering and SVMs in order to compare findings of WebClassII. The research extended its goals to classifying HTML documents into hierarchies of categories as done by Yahoo!. However, findings showed that for increasing feature set sizes, the overall recall fell to only 39.75%.

Estruch et al. (2006) stated that their reason for making use of decision trees was because they believed this technique allows for both the structure and the content of each web page to determine the category in which they belong. The research adapted the decision tree inference strategy from the Centre Splitting Method (Thornton, 2000) in order to consider only sets of attributes rather than all possible attributes. The researchers believed that this was the change that differentiated their approach from normal decision trees and made it possible for their system to deal with structured attributes such as lists, trees etc.

Advantages of decision trees include the simplicity of interpreting models. Calculating the entropy and information gain also helps to consider the most important factors in a dataset first and place them at the top of the tree. Decision trees, differently from NB Classifiers, can cope with combinations of terms, thus they can produce better results for some domains. A weakness of decision trees however, is that they do not support incremental training (Segaran, 2007). Training such a system means using the entire training dataset each time, which makes it difficult as one would have to start the training from scratch every time the system needs to be trained with a new piece of evidence.

Another disadvantage is that decision trees can get out of hand with the number of nodes created for some problems. According to Russell & Norvig (2003), with six Boolean attributes there would be need for 18,446,744,073,709,551,616 distinct nodes. This shows that decision trees can become very large and complex especially for feature-packed documents such as web pages.

2.3.6.4 Support-Vector Machines

Support-Vector Machines (SVM) are popular methods used to classify items by identifying what is called a Maximum-Margin Hyperplane, which splits the data into different classes or categories (Haykin, 2008). The dividing line needs to be drawn in a position that makes the parallel lines drawn to touch the nearest data items from each class, be as far away from the dividing line as possible. SVM view each data item as an x-dimensional vector. The data items nearest to the line are called the support vectors and the algorithm that finds these support vectors and uses them to find the dividing line is called the support-vector machine.

Once a dividing line is found, classifying new items is easy as it only requires plotting the new items onto the graph and seeing on which side of the line they fall. Finding the dividing line in the first place is where the intelligence and complexity lie. The following figure shows two different situations, the first being a simple case where the dividing line is a straight linear line and the second one showing a more complex example where a straight line cannot be found to divide the data.

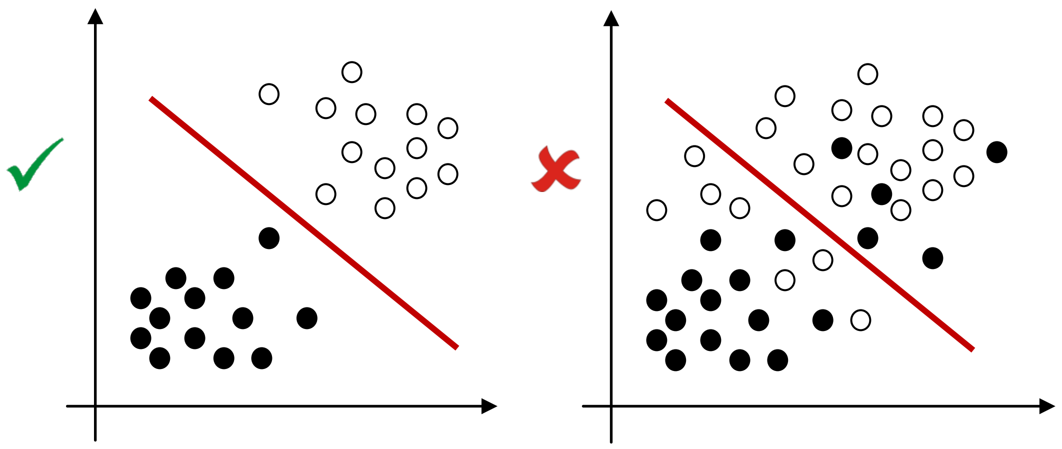


Figure 2.9: Plots of Data and Dividing Lines

In cases like the second one, a possible solution is to transform the data onto a different space e.g. a three-dimensional space. This is called Polynomial Transformation and it transforms data onto different axes, where it is then possible to get a different view of the data that may help to divide it in the normal linear fashion. This is however not possible all the time. There are cases when the data is so complex, it would need to be transformed into spaces with thousands or even unlimited number of dimensions before it could become suitable for linear division. In these situations, instead of transforming the space (which might be impossible), SVM methods and other linear classifiers use what is known as the Kernel Trick, which involves replacing the dot-product function[1] with a new function that returns what the dot-product would have returned if the data had first been transformed to a higher dimensional space (Chen & Bhattacharya, 2006; Segaran, 2007).

SVM techniques are used to solve many data-intensive scientific problems due to their ability to manage high-dimensional datasets well. Some of the areas SVM have been used on are:

- Classifying facial expressions (Michel & El Kaliouby, 2003; Ryan et al., 2010)
- Predicting the structure of proteins (Cai et al., 2001; Huang & Shi, 2005)
- Handwriting recognition (Ahmad et al., 2002)
- Predicting the damage potential of earthquakes (Tesfamariam & Liu, 2010)

The above are areas of great importance, thus Support-Vector Machine classifiers need to be trained and tested extensively. Training algorithms for SVM systems require mathematical concepts that are computationally challenging and expensive.

Joachims (1998) was the first one to apply the SVM technique to text classification. In his research Joachims explored two classes of non-linear SVMs: polynomial classifiers and radial basis functions, however he observed only small benefits compared to linear models. Dumais (1998) continued on Joachims' footsteps and also used SVMs for text classification. Their research works differed in many ways however, with Dumais chosing binary instead of weight values of page features, 300 features instead of all the features of the page and linear models instead of non-linear models. However, the results from both works were very similar in their accuracy, which shows that SVMs are flexible in the way they handle data.

Basu et al. (2003) concentrated on the classification of news web pages. Their research tried to discover whether the automatic classification of such pages could be improved by first filtering the vocabulary of each page and reducing the number of features used during processing. Zhang and Gong (2010) also focused on the problem with synonymous features and they used Support-Vector Machines with latent semantic analysis to try to resolve it.

The main advantage of SVM classifiers is that once they are trained, they are very fast at classifying data, because all they need to check is on which side of the dividing line the new data falls. These classifiers are well suited for large datasets, as they are powerful when all parameters have been identified correctly. However, like NNs, SVM classifiers are also seen as a 'black box' approach. It is even more difficult to interpret the results of an SVM classifier due to the transformation they perform of the data onto higher dimensional spaces, which is practically impossible to perform manually.

Despite the benefits of the above approaches, this research is in collaboration with a small organisation, thus the project dataset and the organisation's hardware and software limitations had to be considered before deciding on a classification technique. SVM and Clustering would be too expensive and process intensive for the organisation, thus they were considered inappropriate for this research. NNs, DTs and NB networks were taken into consideration and experiments were set up to compare the three techniques (Xhemali et al., 2009). NB networks were found to be most suitable for this research as explained in chapter 5.

[1] An operation which takes two vectors X and Y and returns a real-valued scalar quantity by: $X * Y = |X||Y| \cos\theta$ where θ is the angle between the two vectors (Arfken, 1985).

2.4 WEB EXTRACTION

According to the National Institute of Standards and Technology (NIST, 2010), information extraction can be defined as: “The extraction or pulling out of pertinent information from large volumes of texts”. Eikvil (1999) views WIE as: “An attempt to convert information from different text documents into database entries”. The above is precisely what this research is trying to achieve, specifically, the extraction of training course information from web pages into ATM’s database.

There are different technologies that attempt to automate the extraction of information from the web into more manageable data stores, such as databases or excel spreadsheets. The traditional approach is through Wrappers, which are specialised programs that attempt to convert web data into more structured formats, which are simpler to manipulate and extract. Early approaches to creating Wrappers were based on manual techniques (Hammer et al. 1997; Huck et al. 1998), where individuals were in charge of examining the web pages, manually finding the areas of interest on these pages and then creating patterns for extracting them. The tool created by Hammer et al. (1997) is a good representation of this approach, because, for it to work, it requires a person to manually create a declarative specification that establishes the location of the areas of interest on each web page and how this data should be packaged into objects. These techniques were labour intensive, time consuming, difficult to create and to maintain. Thus, the objective for many researchers has since been to find ways to automate the process of generating wrappers for the extraction of web information.

The main approaches that have been investigated include: HTML wrapper generators, which rely on finding patterns in the HTML code behind web pages; XML wrapper generators, which convert the HTML code to a more structured format first (XML) and then apply the extraction rules to the XML document; Machine Learning techniques, which aim to inject some intelligence into the decision making process, resulting in generalised extraction rules; Extraction Ontologies, which are able to reason about relations in the data, thus handling similar structures well; Natural Language Processing, which attempt to ‘understand’ the content of the web page through complex grammatical rules. Another approach that could be used to automate web extraction is Genetic Programming (GP), although few researchers have investigated using GP to aid WIE so far, as will be discussed later on. The following sections give a brief description of each approach in relation to the research ideas in this thesis. More details on these techniques can also be found in Xhemali et al. (2007, 2010).

2.4.1 HTML WRAPPER GENERATORS

Wrapper generators based on the analysis of HTML pages, rely on finding patterns in the structure of the source code behind each page. Some of the structures considered include: HTML tags, font sizes, font weights, letters cases, indentation etc.

The normal technique used in these approaches involves the analysis of an initial web page, picked at random from a predefined website. The pattern recognition process is then performed on this page and the resulting code is applied to all other pages of the website.

Webfoot (Soderland, 1997) attempted to extract weather forecast information related to different locations on different days by analysing an initial web page as above. The page was first divided into sentence-length logical segments based on page layout characteristics. The segments were then inputted into Crystal (Soderland et al., 1995), a Natural Language Processing (NLP) system that learned text extraction rules from example.

Webfoot succeeded in using NLP tools with non-grammatical text found on the web. This is an achievement, because NLP normally requires the data to be in full grammatical sentences. On the downside however, Webfoot was a static approach as it relied solely on the fact that all weather websites described the forecast in exactly the same way (e.g. there needed to be a new line on the page starting with the weekday, followed by ":" or "-" or "…", followed by the forecast … etc.)

Yang & Zhang (2001) also based their approach on analysing the content of one initial HTML page, however their method did not require prior knowledge of the content or structure of the web page. Unlike Webfoot, this research believed HTML tags were not stable enough to be considered in the pattern finding process. Instead areas of interest on the page were separated based on apparent visual boundaries, such as: headings, subheadings etc. Web pages were arranged as tree structures where all the tree nodes represented the different patterns found on the pages. All the patterns found were then compared to each other in order to generate wrappers that applied to various types of web pages. This approach is a step forward from Webfoot, however it still fails to achieve satisfactory results from web pages with a lot of information on them or those with complex table-based layouts.

A different approach from the above two was introduced by Crescenzi et al. (2001), who created RoadRunner, a fully automatic wrapper generator. Similarly to the work of Yang & Zhang (2001) RoadRunner does not rely on any prior knowledge of the web pages, which helps towards the creation and maintenance of the wrappers. However, RoadRunner differs from the above approach as it analyses two different web pages from the same website at a time. The two pages are compared to determine all the similarities and mismatches between them. The similarities and mismatches are then used to identify relevant structures for each website and generalise the wrapper. Thanks to this approach, RoadRunner does not rely on a training set (user-specific examples) and it can handle flat as well as nested HTML structures.

Despite the fact that RoadRunner was one of the first techniques to achieve automatic extraction, the field names, corresponding to the data extracted from each website, had to be manually renamed, thus complete separation between processing and users was not achieved.

Reis et al. (2004) followed in the footsteps of RoadRunner and worked with two HTML pages at a time. In contrast however, this Brazilian news extraction tool was based on the concept of tree-edit distance[2] to evaluate the similarities in structure amongst the pages. It also required a training set to achieve the page clustering (i.e. grouping together pages with matching formats).

Similarly to the research in this thesis, all the above approaches assume that web pages within the same website inherit common format and layout characteristics. In this research however, this is more of an observation than an assumption.

The above research works concentrate on extracting large 'chunks' of data from the web, whereas this research concentrates on extracting specific information such as course prices,

[2] Edit distance between two trees T_A and T_B is the cost of the minimal set of operations required to transform T_A into T_B.

locations, dates etc., thus in this case the extraction algorithm has to be more 'aggressive' in order to identify detailed rules to match these precise pieces of information.

2.4.2 XML WRAPPER GENERATORS

XML (eXtensible Markup Language) approaches have become an important part of WIE projects, due to web irregularities and HTML inconsistencies. The main idea behind these approaches is the conversion of the HTML code behind each web page to the more structured, XML format. WIE techniques are then applied to the XML document produced.

By converting the HTML document to XML, the data is separated from its layout. This is because, unlike HTML, which is designed to focus on how the data looks on a page, XML is more concerned with describing the data itself. This is achieved through the Document Type Definition (DTD) or XML Schema, which describes the data. XML tags are used to aid the description of the data. Unlike HTML tags however, XML tags are not predefined; programmers must define their own tags, which gives more control over the description process. An XML document however, would be of no use by itself, as its tags do not mean anything to computers. XML documents have to be further manipulated by external packages, such as WIE systems.

Liu (1999) developed mechanisms that provided a clean separation of the semantic meaning of information on each web page from the process of wrapper generation. Myllymaki (2002) also converted each HTML web page to XML as the first step of the extraction process, however in this case the HTML code was first translated to XHTML (eXtensible HTML), which is 'repaired[3]' from any abnormalities such as missing tags, etc.. The XHTML code was then converted to XML. Despite the success of this research in relying less on HTML and more on the content itself, it was still too rigid in the way it found items of interest on the target web pages, e.g. they would look into the cells of a table and extract only the information in bold. Needless to say, this technique would not work if the item in bold was changed to e.g. italic. Nevertheless, this was an improvement on techniques such as the one used by W4F (Sahuguet & Azavant, 1999), which used absolute paths to the location of interest. For example, the path to the third column of the second row of the first table would be: HTML.BODY.TABLE[1].TR[2].TD[3]. W4F could not deal with layout changes.

More recent research include: the combination of XML data conversion technologies with Document Object Model -based XPath generation technologies to simplify the extraction of web information from HTML sources (Gao et al., 2007; Hu and Xuan, 2008); the application of XML-based WIE techniques for the improvement of web-based, distance learning systems (Tang et al., 2010) etc. The research presented in this thesis combines XML with Genetic Programming (Xhemali et al., 2010-b; Xhemali et al., 2010-c) to aid the evolution of WIE solutions.

2.4.3 MACHINE LEARNING

The Oxford Dictionary defines 'Learning' as: "The acquisition of knowledge or skills through study, experience, or being taught". Machine Learning (ML) is exactly this; the only exception is that the learning process has to be achieved by machines. Mitchell (1997) describes ML as "the study of algorithms which improve from experience". This ability to

[3] There are other tools that achieve the latter, such as the Tidy package (Raggett, 2007).

make intelligent decisions and to continuously improve makes ML very popular. An additional benefit, particularly for WIE, is the ability of these techniques to eliminate much, if not all of the user involvement from the extraction process.

ML techniques can be categorised based on the amount of available feedback given to the learning process (Russell & Norvig, 2003). These categories include:

- <u>Supervised Learning</u> – The system learns relationships between input(s) and output(s) from user defined training data. The acquired knowledge is then used to extract new information.
- <u>Reinforcement Learning</u> – The system learns not to make the same mistake twice, through reinforcement.
- <u>Unsupervised Learning</u> – This is the most difficult as the system learns relationships between inputs alone without any knowledge of what the outputs should look like.

The above can be further subcategorised into the following two categories:

- <u>Rule-inferential ML</u> – This category includes techniques which learn rules from user-defined training data.
- <u>Statistical ML</u> – The techniques in this category are based on representing the data as statistical models, often passing sets of probabilities as parameters.

The following discusses past and present research in the above categories.

CRYSTAL (Soderland et al., 1995) uses a ML algorithm, however it requires a semantic hierarchy of the data, as well as the training data to be manually annotated by an expert. CRYSTAL learns and creates extraction rules by generating multi-slot concept frames. This allows for related information to be extracted together.

The multi-slot concept is also important to the research in this thesis, because a web page may list many courses with associated prices, locations, dates etc., however, unless all this information is managed as a set, the results would be incomplete.

In 1998, Soderland introduced WHISK, a system which, like CRYSTAL, can handle single as well as multi-slot extractions. However, in contrast from the former, it does not require syntactic analysis for structured or semi-structured data. When combined with a syntactic analyser and a semantic tagger, WHISK outperforms its ancestors in the extraction of free text information such as news stories.

Boosted Wrapper Induction (BWI) (Kauchak et al. 2004) uses an algorithm called AdaBoost to learn two sets of boundary detectors separately. This allows BWI to recognise the beginnings and ends of the areas of interest to be extracted. Then BWI re-checks all the examples, repeatedly for a fixed number of times, in order to focus on areas of the training data that the current rules do not match well. BWI is known to work well with unstructured natural language documents.

OMINI (Buttler, 2001), KNOWITALL (Etzioni et al., 2005) and TEXTRUNNER (Banko et al., 2007) are amongst the latest automated WIE systems, which all claim to be the best in the field. However, they all have limitations.

OMINI (Buttler, 2001) uses five different heuristics to fully automate the object extraction process from the web. It works with static and dynamic websites and achieves high precision and recall values, because it not only uses each heuristic individually, but it also combines them together. However, OMINI cannot manage multiple areas of interest on a page. Furthermore, OMINI assumes that the records to be extracted are always located under the largest HTML tree, which is not always the case.

KNOWITALL (Etzioni et al., 2005) automates WIE by learning to label its own training examples. KNOWITALL depends on existing search engines such as: Google, Alta Vista and Fast, to collect the necessary web pages required and creates extraction rules by using an extensible ontology and a small number of generic rule templates. Banko et al. (2008) argue that their system – TEXTRUNNER – is more efficient than KNOWITALL. Their reasons include the fact that KNOWITALL requires a large number of search engine results and web page downloads to work properly, which means the entire process can take weeks to complete. Furthermore, KNOWITALL uses relation names as input, which means that the system needs to be rerun each time a relation of interest is identified. TEXTRUNNER, on the other hand, achieves a 33% error reduction for a similar number of extractions to KNOWITALL. It does so by utilising three main modules: (1) a Learner, which given a sample of data as input, produces a classifier that categorises the different extractions as trustworthy or not; (2) an Extractor, which generates candidate tuples from each sentence and triggers the Classifier to work on these tuples; (3) an Assessor, which removes redundant tuples and merges similar ones.

The research presented in this thesis and TEXTRUNNER both use the NB approach for the classification process. However, instead of generating candidate tuples from each sentence on a web page to pass to the classifier, this research considers only the top 100 most used keywords from each page, ignoring stopwords, etc. Additionally, this research has developed an extensive vocabulary which helps to reduce the number of keywords considered from a page further. This vocabulary includes keywords that although they do not appear in standard stopword lists, they have been proven to be unnecessary in distinguishing web pages from one another, particularly in the training courses domain.

The extraction process in this research differs greatly from all of the above techniques, as it focuses on optimising WIE solutions by trying to evolve new and better solutions for unknown web pages, rather than simply hoping that the system will find the right answer straight away. Section 6.2.1 discusses the contributions of this research in detail.

2.4.4 EXTRACTION ONTOLOGIES

HTML-based and Machine Learning approaches try to find answers to questions such as: “How to discover patterns in web pages”, “How to create rules to apply to all these patterns”, “How to train a system so it learns where to look for data” etc. However, there are other questions that can be asked about a WIE system, which the above do not consider. One such question is: “How to restructure the content of a web page, so the new structure can be easily extracted”. One answer to this question involves using Ontologies to restructure the web pages into standard models that are independent of the original information sources (Snoussi et al., 2002).

Snoussi et al. (2002) focused on websites which change the content frequently but not the overall structure e.g. stock exchange quotes. The system works by first converting the page content into XML format, then using Ontologies to model the data and assign it semantics before finally carrying out the extraction of the data.

Embley (2004) defined Extraction Ontologies as: “An augmented conceptual-model instance that serves as a wrapper for a narrow domain of interest”. This research concentrated on the car advertisement domain. Differently from Snoussi et al. (2002), Embley incorporated machine-learnt rules over the chosen heuristics, to determine whether or not a web page was applicable for a given Ontology. The research achieved over 90% for both recall and precision ratios. Later research (Embley et al., 2005) focused on retargeting the ‘car advertisement’ domain to domains such as: mobile phones, restaurants, games etc. However, the Ontology required a few dozen person-hours to be updated to fit the description of each new domain.

A main benefit of Extraction Ontologies is that they are competent at reasoning about relations in the data (Embley, 2004). However, they are very complex and there are certain aspects to be considered when dealing with Ontologies such as:

- Reusability – Ontologies tend to be created for a specific domain, in order for them to be successful at properly understanding the relations between the data in that domain. This reduces the reusability factor. Attempts have been made to generalise Ontologies (Gomez et al., 2001; Aubrecht et al., 2005; Toral & Monachini, 2007; Manzano-Macho et al., 2008), also known as Upper or Domain-Independent Ontologies, to describe very general concepts that are well known to most domains, however these would not work as well as detailed Ontologies on specific domains. An alternative idea is to merge different Ontologies together (Bao et al., 2004) in order to expand the number of domains on which the Ontology works, whilst still maintaining a detailed view of each domain. However, this could be an error-prone, time-consuming and expensive process.

- Maintenance – Ontology maintenance deals with updating Ontologies in cases when: attributes for new products need to be added; words that differ in meaning over time need to be updated; slang and colloquial expressions need to be included etc. Ontology maintenance can be a complex and expensive process, especially for large-scale Ontologies, where the information to be analysed is more complex. This is however, inevitable, because, as requirements grow and knowledge about domains evolves, change also becomes inevitable. Whilst some encouraging results have been achieved for the complete automation of Ontology construction and maintenance (Blaschke, 2002; Shamsfard & Barforoush, 2004, Rafi et al., 2009), most Ontologies still rely on the manual influence of experienced knowledge engineers (Khan & Luo, 2002; Labsky et al., 2008). This is, however, not only time consuming but also error prone. Iida et al. (2009) carried out experiments to measure six different metrics for the maintenance of an “evaluation expression ontology”. They discovered that the maintenance quality depends a lot on the skill of the engineer, with more than 10% of inaccuracies being human errors. Furthermore, even with the help of Ontology editors such as Protégé[4] etc., it was calculated that an engineer would need a minimum of 150 hours to perform maintenance of three genres of the “evaluation expression ontology”, which included a total number of 1894 entries.

[4] Protégé: http://protege.stanford.edu

The above shows that although Ontologies would be very beneficial for some domains and organisations, they can easily and quickly turn into very expensive processes. ATM is a small company. There are currently no software developers or Ontology experts in this company, thus it would be very difficult, if not impossible, for them to support an Ontology based system in the premises. Ontologies were therefore considered unsuitable for this research.

2.4.5 NATURAL LANGUAGE PROCESSING

Information on web pages is displayed in many different formats ranging from structured tables to completely unstructured text. Creating rules to extract information from free text is difficult, because, the layout of the data is no longer helpful, and understanding the meaning of the items of interest is more important.

Computers are not capable of ‘understanding’ the data they work with; they need to be directed to it. Natural Language Processing (NLP) techniques try to help computers recognise language structures as an individual would. This is not as straightforward as it sounds, because there are many exceptions to the rules that define a language as well as other irregularities such as:

- Pronouns are used to replace nouns. E.g. “The course was good. It helped everyone”. How would the computer know that ‘it’ represents the ‘course’?
- Some words are spelt the same but mean different things, e.g. river bank vs. financial bank. The system needs to use the context to distinguish between the two.
- Synonyms are also common parts of a language. WIE systems need to realize that all synonyms describe the same situation.
- English has exceptions. Most verbs produce their past tense by adding ‘ed’. However, eat → ate, etc.
- Humans use many vague terms, which are imprecisely defined. WIE systems need to reason about such fuzzy terms.

The above give only a taste of the problems facing NLP techniques. Research works that have concentrated in this field (Zhang et al., 2004; Tedmori, 2008), usually incorporate techniques such as: filtering; lexical analysis of words and phrases to separate free text into text tokens; part-of-speech tagging or otherwise known as grammatical tagging, which tags text tokens as parts of speech e.g. nouns, adjectives, verbs etc.

Riloff and Jones (1999) researched the idea of automating the construction of a domain-specific dictionary, using as input only a set of un-annotated training pieces of text and some ‘seed’ words from the interest domain. The heart of their approach is a technique called ‘mutual bootstrapping’ which learns extraction patterns from the ‘seed’ words, then uses these patterns to extract more words from which to continue learning. This approach is successful in generating a dictionary of extraction patterns in parallel with a semantic lexicon of the interest domain. However this may end up being too general, as many domains use similar terms e.g. this research found that the ‘vehicle’ dictionary created for texts related to terrorism was very similar to the ‘weapons’ dictionary.

As mentioned in previous sections, many research works have used NLP techniques in collaboration with other techniques to improve the accuracy of their WIE systems. For example, Collobert and Weston (2008) used a Neural Network as well as NLP to determine the likelihood that a sentence was grammatically and syntactically correct. Joshi and Liu

(2009) combined NLP with HTML DOM analysis for the web extraction of text content and associated images. Lonsdale et al., (2010) rely on the combination of Ontologies and Natural Language Processing, otherwise known as “Linguistically Grounding Ontologies” for multilingual extraction of web information.

Other uses of NLP techniques include: classifying people‘s opinions over subjects from various web pages (Nasukawa and Yi, 2003), summarising the content of specific websites automatically (Zhang et al., 2004), analysing music lyrics (Mahedero et al., 2005) by concentrating mainly on structure detection and text categorisations, analysing emails for knowledge discovery (Tedmori, 2008) etc.

The research works discussed in this section show that NLP techniques are most appropriate when attempting to analyse and extract multiple sentences e.g. news summaries or email contents, because this way they can truly analyse text based on grammatical rules. This research is interested in extracting specific pieces of information from web pages such as: course titles, locations, dates and prices, thus NLP techniques would be unsuitable.

2.4.6 GENETIC PROGRAMMING

At the most abstract level, Genetic Programming (GP) can be defined as: “*a systematic, domain independent method for getting computers to automatically solve problems starting from a high-level statement of what needs to be done*” (Langdon et al., 2008).

GP is considered to be a special case of Genetic Algorithms, whereby the evolved objects are not directly solutions to a problem; they are programs that, when executed, may potentially be solutions to the problem (Brameier and Banzhaf, 2007). This research fits into the GP category, because it uses the genetic evolution principles to evolve Regular Expressions (REs), which, when executed against web pages, may retrieve relevant training course information to assist ATM.

Genetic evolution in Computer Science was inspired by the biological evolution process and Darwin’s theory of evolution through natural selection. Fogel and Holland were amongst the first pioneers who foresaw how nature’s evolutionary qualities could be applied to Artificial Intelligent (AI) systems. Fogel’s work included predicting prime numbers, predicting symbols (Fogel et al., 1966) – in the latter, the evolutionary process consistently outperformed humans, whilst Holland (1975) managed to demonstrate that a population of fixed length strings could be genetically reproduced.

Since its establishment, GP research has attracted attention in various fields such as: electronics (Long et al., 1997), plant biology (Dyer and Bentley, 2002), military defence (Jackson, 2005), spam filtering (Conrad, 2007), feature extraction from media files (Hsu, 2007; Klank et al., 2008), game strategies (Keaveney and O’Riordan, 2009), railway platform allocation (Clarke et al., 2009), etc., however few researchers have investigated using GP to automate information extraction from the web. Even fewer researchers have investigated evolving REs for the purpose of WIE.

Research works using GP for the automatic extraction of information include: Atkinson-Abutridy et al. (2004) and Gordon et al. (2006) who use GP to evolve association measures for the purpose of text mining and information retrieval, respectively; Snajder et al. (2008) who also use GP to evolve corpus and language –independent association measures for the extraction of collocations; Zhang and Rockett (2006) who concentrate on using GP for feature extraction, specifically for the purpose of image processing; Christy and Thambidurai (2007) who concentrate on using genetic evolution with Natural Language Processing techniques and soft matching rules for the extraction of information from research abstracts. The genetic evolutionary system in their research is used after the extraction process has occurred to aid the classification of the extracted information into the appropriate categories based on their relevancy.

None of the above works attempted the evolution of Regular Expressions (REs) for the automation of WIE. As a matter of fact, to the best of our knowledge, there is only one other research project besides the one presented in this thesis, that focuses on the evolution of REs for WIE (Barrero et al., 2009). Barrero et al. (2009) concentrate on evolving REs for the extraction of web information such as URLs and phone numbers. The research uses a multi-agent system (MAS), where all agents guide the evolutionary process by sharing a training set composed of positive and negative examples. Each agent manages the evolution of a population of fixed-length chromosomes. Different agents can influence each other's populations by means of migration, thus the final genotypes from which REs are created are of variable length. The basic REs created are then combined through subsets of RE operators such as "|, (,), +, ?" to create more complex REs. However, the combination rules are basic, such as "RE1+RE2?", "RE1|RE2", "RE1RE2" and "(RE2)+|(RE1)+" and according to their experiments, the composed REs always end up following the "RE1|RE2" rule, which is an obvious result, making this additional system stage somewhat redundant.

Despite the lack of research in the area of RE evolution for WIE, some research has been achieved in the evolution of REs for other purposes. For example, Heddad et al. (2004) generated RE-based classifiers to locate motifs in DNA sequences that had been manually discovered in labs. Conrad's (2007) involvement with RE evolution was for the purpose of detecting spam emails. Langdon (2008) and Langdon et al. (2008) used GP to evolve REs to predict probe quality after observing that the DNA sequences that form the probes, can indeed indicate poor performance in the probes. Continuing in a similar environment, Langdon et al. (2009) carried on with the evolution of REs, this time for the classification of gene exons. The REs were represented in Backus-Naur form (BNF). This was done in order to verify the validity of the REs to be evolved.

In this research, the syntactic validity of the evolved REs is ensured through the novel idea of having XML rules to guide the process of translating genes to their corresponding RE structures. The modulo function is used to assist the translation/mapping process, as inspired by the work of Withall et al. (2008). Details on this technique can be found in chapter 4 as well as in the attached papers (Xhemali, et al., 2010-b; Xhemali, et al., 2010-c).

Cetinkaya (2007) also used the BNF syntax to verify a subset of POSIX (Portable Operating System Interface for Unix) REs before evolution took place. However, this research used a rather rigid Fitness Test whereby a multiline text file, containing all the possible outcomes of the REs, was specified before the process began. An optimal RE was one that matched all the lines in the file provided.

The above technique would be unsuitable for this research, because the information needed to be extracted may be entirely different from page to page, thus it is impossible to create a file that would have a thorough enough list of all the pieces of information required from the extraction process.

Some researchers (Huang, & Wang, 2006; Braga et al., 2008) have worked towards using GP in combination with other well known AI techniques (in this case Support Vector Machines) to calculate fitness scores. Similarly, the research presented here uses a NB Network to help with the fitness scoring of REs throughout the extraction of training course titles. This combination of approaches represents a novel idea, because differently from standard evolutionary techniques, in this research, additional information is passed from generation to generation, independently of the genetic evolution process. This means that the decision making process after each generation benefits from the newly acquired knowledge from the previous generation. Details on this technique and other detailed information about GP in relation to this research and other related works can be found in chapter 4.

2.5 COMMERCIAL WIE SOLUTIONS

There exist some solutions that have already been marketed as fully operational WIE software systems. Some of these systems were tested and compared to view the extent to which they achieve WIE. The following three were chosen for evaluation, because they were the ones most related to the research presented in this thesis and the ones offering most functionality.

2.5.1 AUTOMATION ANYWHERE

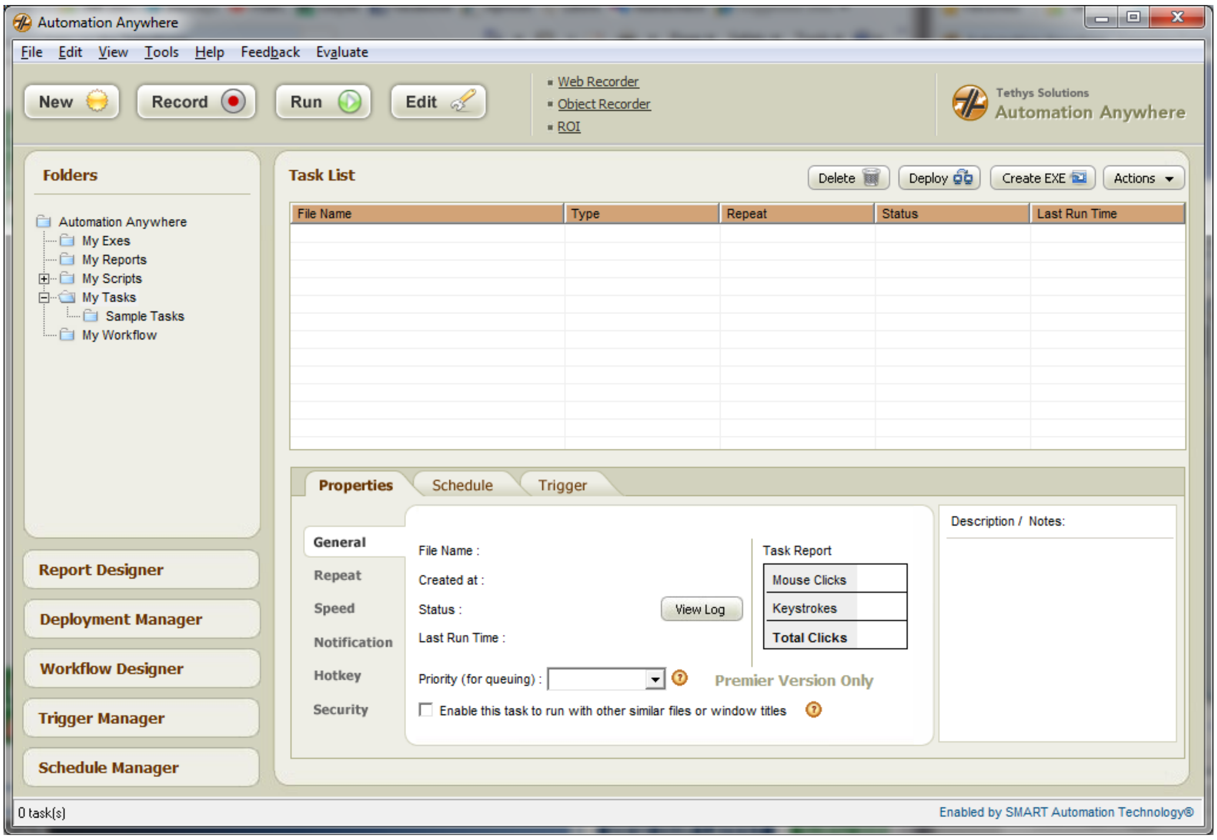
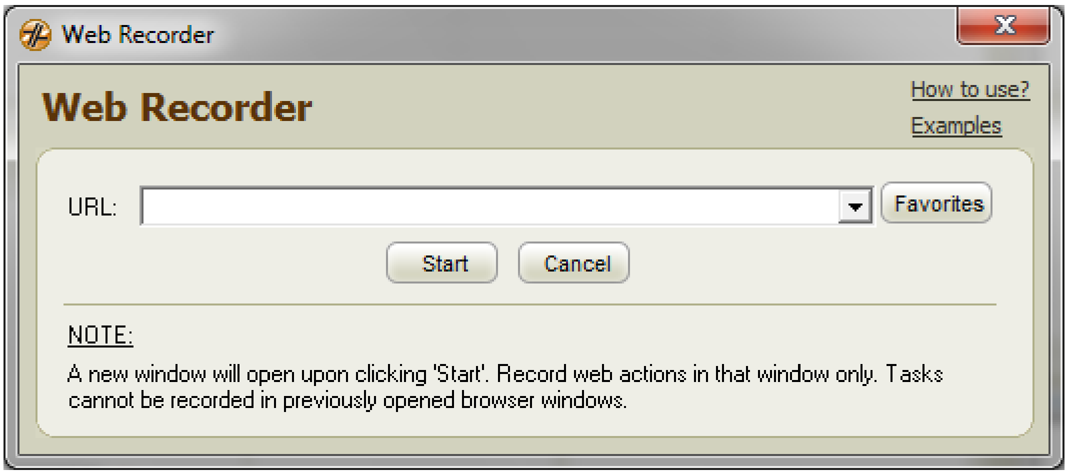


Figure 2.10: Automation Anywhere (Screenshot 1)

Automation Anywhere (Tethys Solutions, 2010) is a software package that helps users to automate the extraction of information of interest from web pages; the scheduling and execution of tasks and other windows applications; the management of tasks within Excel spreadsheets or databases such as: report generation etc. Automation Anywhere was awarded the 2009 Windows IT Pro[5] Editor's Gold award for the "*Best IT Automation Product*". The price of this system ranges from $695.00 to $2,495.00 for standard and premier licences for desktop products, and $7,000.00 for server licences.

Figure 2.10 shows the home page of the system. The 'Web Recorder' link launches the WIE part of the system. The URL of the page under investigation is required before anything else can happen (Figure 2.11). Once the web page is loaded, the user needs to highlight the areas of interest on the page and then click the appropriate button to 'capture' this information. Note that areas of interest only represent single items, not groups of items. Each 'captured' item needs to be given a name. This name will later on serve as the column name for the data field containing the item. The 'capturing' and naming processes need to be repeated for each individual item of interest on the page. The user also needs to tell the system whether or not the data spans over multiple pages and if it maintains a similar format.



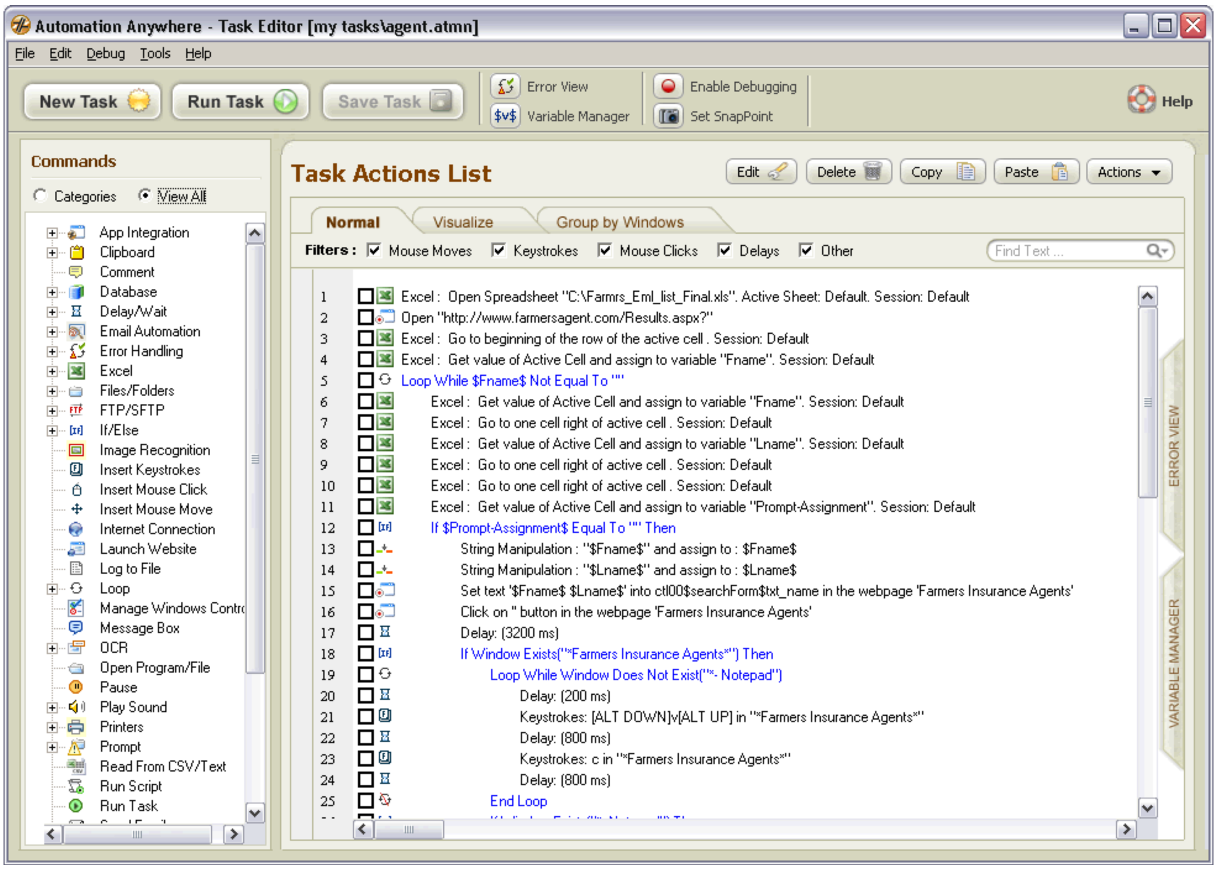
A new window will open upon clicking 'Start'. Record web actions in that window only. Tasks cannot be recorded in previously opened browser windows.

Figure 2.11: Automation Anywhere (Screenshot 2)

Once all of the above preparations have been completed, the system creates a 'Task Actions List', which incorporates all the user's actions into a pseudo code -like design. Figure 2.12 shows an example related to actions concerning an Excel spreadsheet.

The user can then choose to save the 'pseudo code' and execute it on that website whenever they need to do so. The above process needs to be repeated for each new link, from which the user wants to extract information. It is clear that this can easily become a large and time-consuming task.

[5]http://www.windowsitpro.com/article/services/2009-windows-it-pro-editors-best-and-community-choice-awards/9.aspx



Filters : ☑ Mouse Moves ☑ Keystrokes ☑ Mouse Clicks ☑ Delays ☑ Other

```
Excel :  Open Spreadsheet "C:\Farmrs_Eml_list_Final.xls". Active Sheet: Default. Session: Default
Open "http://www.farmersagent.com/Results.aspx?"
Excel :  Go to beginning of the row of the active cell . Session: Default
Excel :  Get value of Active Cell and assign to variable "Fname". Session: Default
Loop While $Fname$ Not Equal To ""
    Excel :  Get value of Active Cell and assign to variable "Fname". Session: Default
    Excel :  Go to one cell right of active cell . Session: Default
    Excel :  Get value of Active Cell and assign to variable "Lname". Session: Default
    Excel :  Go to one cell right of active cell . Session: Default
    Excel :  Go to one cell right of active cell . Session: Default
    Excel :  Get value of Active Cell and assign to variable "Prompt-Assignment". Session: Default
    If $Prompt-Assignment$ Equal To "" Then
        String Manipulation : "$Fname$" and assign to : $Fname$
        String Manipulation : "$Lname$" and assign to : $Lname$
        Set text '$Fname$ $Lname$' into ctl00$searchForm$txt_name in the webpage 'Farmers Insurance Agents'
        Click on '' button in the webpage 'Farmers Insurance Agents'
        Delay: (3200 ms)
        If Window Exists("*Farmers Insurance Agents*") Then
            Loop While Window Does Not Exist("*- Notepad")
                Delay: (200 ms)
                Keystrokes: [ALT DOWN]v[ALT UP] in "*Farmers Insurance Agents*"
                Delay: (800 ms)
                Keystrokes: c in "*Farmers Insurance Agents*"
                Delay: (800 ms)
            End Loop
```

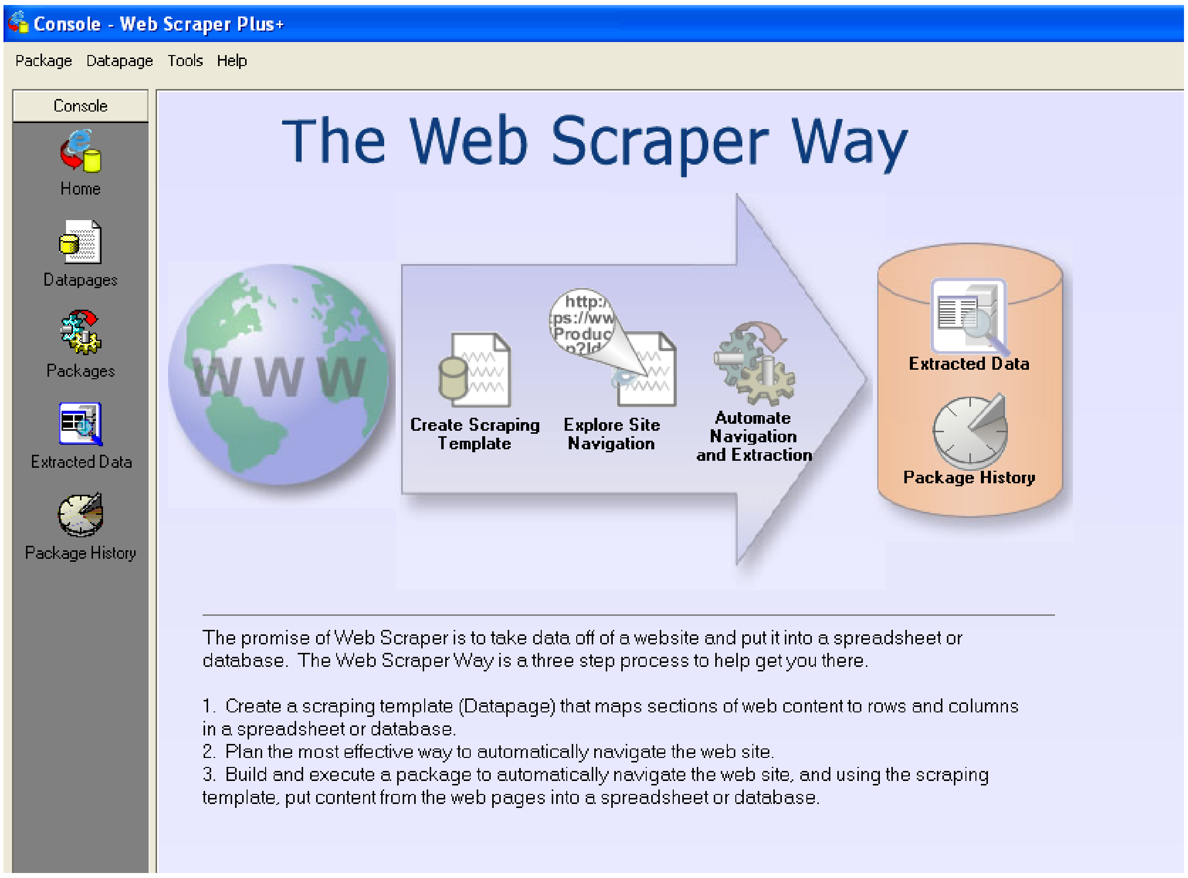
Figure 2.12: Automation Anywhere (Screenshot 3)

2.5.2 WEB SCRAPER PLUS+ (WSP+)

WSP+ (Velocityscape Ltd.) has been marketed since 1999 and it is advertised as "*The most powerful Web Data Extraction platform under $50k*". Similarly to the research presented in this thesis, it aims to extract data from the web and store them into a spreadsheet or database. A single machine licence for this product costs $749.00.

WSP+ allows extraction of data from different structures such as tables, lists, text fields etc. The system also allows for the extraction of information from a combination of the above structures, however in these cases, the data will be mixed up in the Excel spreadsheet or database if the data is not perfectly structured. For example, if there was no Street Name for a particular business, but all the other information such as: City, Post Code etc. was present, then WSP+ would save the City into the Street Name field in the spreadsheet, the Post Code into the City field and so on. This would clearly result in incorrect information.

WSP+ is more complex than Automation Anywhere and users need to have a moderate understanding of web navigation and Excel to successfully operate the basic features of WSP+. In relation to the more complex features of the system, users need to have a solid understanding of web navigation, HTML code, Form Posts and Query Strings, as well as SQL statements and the use of SQL Query Analyser.



The promise of Web Scraper is to take data off of a website and put it into a spreadsheet or database. The Web Scraper Way is a three step process to help get you there.

1. Create a scraping template (Datapage) that maps sections of web content to rows and columns in a spreadsheet or database.
2. Plan the most effective way to automatically navigate the web site.
3. Build and execute a package to automatically navigate the web site, and using the scraping template, put content from the web pages into a spreadsheet or database.

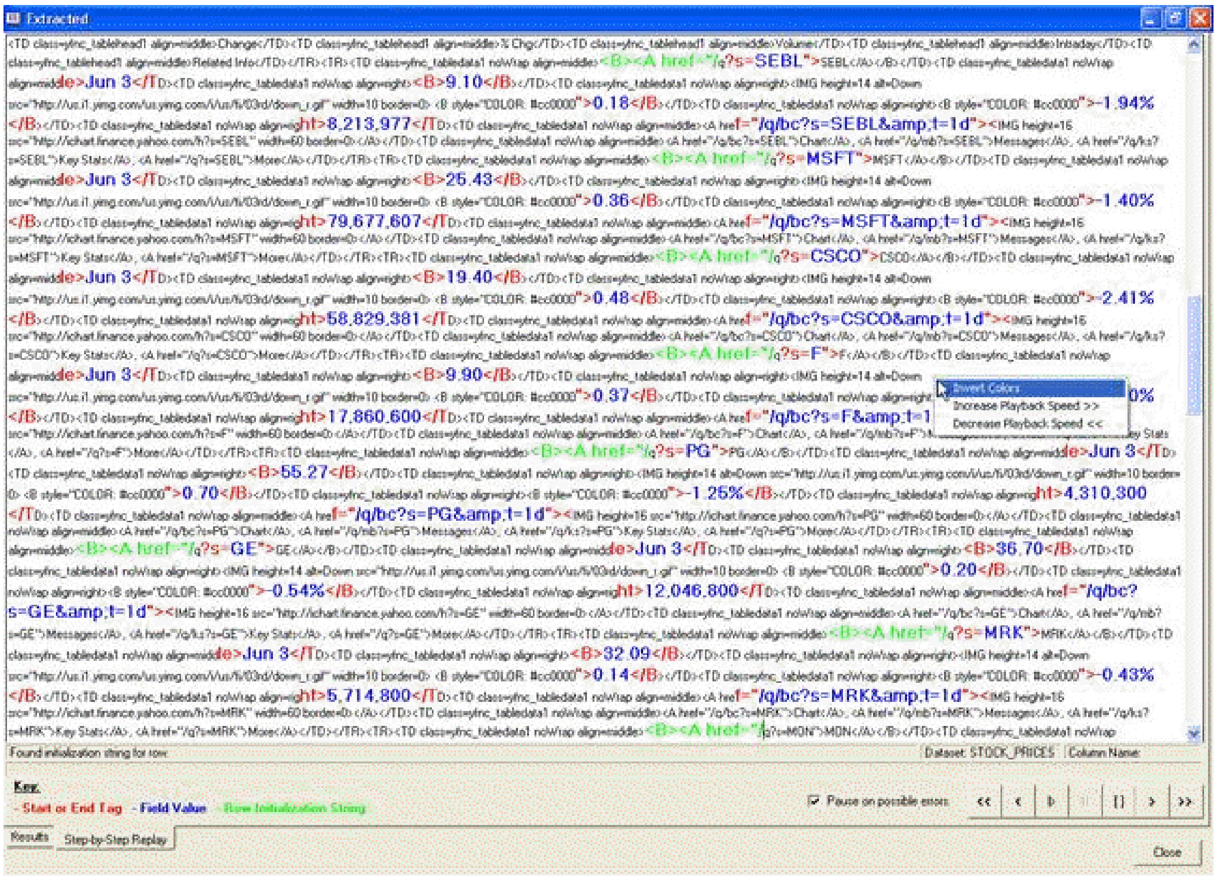
Figure 2.13: WSP+ (Screenshot 1)

To illustrate the complexity, the following list shows the higher-level tasks that need to be carried out by users for the basic extraction of information from a web page:

- Navigate to the template web page (Figure 2.13).
- Create a new Datapage (which is a template that defines the data that the user wants to extract and where they want it stored).
- Start the Datapage properties wizard.
- Create a new Dataset (component of Datapage).
- Create a new Field (component of Dataset).
- Look at the Properties windows and update the information shown if necessary.
- Add additional fields for each of the additional items of interest to be extracted from the page.
- Repeat the steps from ‘Create a new Dataset’ if additional areas of interest on the page are formatted differently from the information included in the previous Dataset.
- Test the Datapage (if unhappy with the results, check the step-by-step replay of the extraction as shown in Figure 2.14 and manually identify the problem).
- Save the Datapage for future use.

Each of the above tasks includes other tedious and time consuming tasks such as: highlighting fields, manually evaluating information, adding/changing field names etc., which are very error prone.

This system is very large and offers a lot of functionality, however it would only be appropriate for users who are prepared to spend a great deal of time on using it and who enjoy being in complete control of what is happening. Users, who need a fast, simple, automated system would not appreciate the way WSP+ manages WIE.



```
<TD class=yfnc_tablehead1 align=middle>Change</TD><TD class=yfnc_tablehead1 align=middle>% Chg</TD><TD class=yfnc_tablehead1 align=middle>Volume</TD><TD class=yfnc_tablehead1 align=middle>Intraday</TD><TD
class=yfnc_tablehead1 align=middle>Related Info</TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=SEBL">SEBL</A></B></TD><TD class=yfnc_tabledata1 noWrap
align=middle>Jun 3</TD><TD class=yfnc_tabledata1 noWrap align=right><B>9.10</B></TD><TD class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down
src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=0> <B style="COLOR: #cc0000">0.18</B></TD><TD class=yfnc_tabledata1 noWrap align=right><B style="COLOR: #cc0000">-1.94%
</B></TD><TD class=yfnc_tabledata1 noWrap align=right>8,213,977</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=SEBL&amp;t=1d"><IMG height=16
src="http://ichart.finance.yahoo.com/h?s=SEBL" width=60 border=0></A></TD><TD class=yfnc_tabledata1 noWrap align=middle> <A href="/q/bc?s=SEBL">Chart</A>, <A href="/q/mb?s=SEBL">Messages</A>, <A href="/q/ks?
s=SEBL">Key Stats</A>, <A href="/q?s=SEBL">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=MSFT">MSFT</A></B></TD><TD class=yfnc_tabledata1 noWrap
align=middle>Jun 3</TD><TD class=yfnc_tabledata1 noWrap align=right><B>25.43</B></TD><TD class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down
src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=0> <B style="COLOR: #cc0000">0.36</B></TD><TD class=yfnc_tabledata1 noWrap align=right><B style="COLOR: #cc0000">-1.40%
</B></TD><TD class=yfnc_tabledata1 noWrap align=right>79,677,607</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=MSFT&amp;t=1d"><IMG height=16
src="http://ichart.finance.yahoo.com/h?s=MSFT" width=60 border=0></A></TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=MSFT">Chart</A>, <A href="/q/mb?s=MSFT">Messages</A>, <A href="/q/ks?
s=MSFT">Key Stats</A>, <A href="/q?s=MSFT">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=CSCO">CSCO</A></B></TD><TD class=yfnc_tabledata1 noWrap
align=middle>Jun 3</TD><TD class=yfnc_tabledata1 noWrap align=right><B>19.40</B></TD><TD class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down
src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=0> <B style="COLOR: #cc0000">0.48</B></TD><TD class=yfnc_tabledata1 noWrap align=right><B style="COLOR: #cc0000">-2.41%
</B></TD><TD class=yfnc_tabledata1 noWrap align=right>58,829,381</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=CSCO&amp;t=1d"><IMG height=16
src="http://ichart.finance.yahoo.com/h?s=CSCO" width=60 border=0></A></TD><TD class=yfnc_tabledata1 noWrap align=middle> <A href="/q/bc?s=CSCO">Chart</A>, <A href="/q/mb?s=CSCO">Messages</A>, <A href="/q/ks?
s=CSCO">Key Stats</A>, <A href="/q?s=CSCO">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=F">F</A></B></TD><TD class=yfnc_tabledata1 noWrap
align=middle>Jun 3</TD><TD class=yfnc_tabledata1 noWrap align=right><B>9.90</B></TD><TD class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down
src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=0> <B style="COLOR: #cc0000">0.37</B></TD><TD class=yfnc_tabledata1 noWrap align=right                    0%
</B></TD><TD class=yfnc_tabledata1 noWrap align=right>17,860,600</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=F&amp;t=1
src="http://ichart.finance.yahoo.com/h?s=F" width=60 border=0></A></TD><TD class=yfnc_tabledata1 noWrap align=middle> <A href="/q/bc?s=F">Chart</A>, <A href="/q/mb?s=F">M                    y Stats
</A>, <A href="/q?s=F">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=PG">PG</A></B></TD><TD class=yfnc_tabledata1 noWrap align=middle>Jun 3</TD>
<TD class=yfnc_tabledata1 noWrap align=right><B>55.27</B></TD><TD class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=
0> <B style="COLOR: #cc0000">0.70</B></TD><TD class=yfnc_tabledata1 noWrap align=right><B style="COLOR: #cc0000">-1.25%</B></TD><TD class=yfnc_tabledata1 noWrap align=right>4,310,300
</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=PG&amp;t=1d"><IMG height=16 src="http://ichart.finance.yahoo.com/h?s=PG" width=60 border=0></A></TD><TD class=yfnc_tabledata1
noWrap align=middle><A href="/q/bc?s=PG">Chart</A>, <A href="/q/mb?s=PG">Messages</A>, <A href="/q/ks?s=PG">Key Stats</A>, <A href="/q?s=PG">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap
align=middle><B><A href="/q?s=GE">GE</A></B></TD><TD class=yfnc_tabledata1 noWrap align=middle>Jun 3</TD><TD class=yfnc_tabledata1 noWrap align=right><B>36.70</B></TD><TD
class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=0> <B style="COLOR: #cc0000">0.20</B></TD><TD class=yfnc_tabledata1
noWrap align=right><B style="COLOR: #cc0000">-0.54%</B></TD><TD class=yfnc_tabledata1 noWrap align=right>12,046,800</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?
s=GE&amp;t=1d"><IMG height=16 src="http://ichart.finance.yahoo.com/h?s=GE" width=60 border=0></A></TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=GE">Chart</A>, <A href="/q/mb?
s=GE">Messages</A>, <A href="/q/ks?s=GE">Key Stats</A>, <A href="/q?s=GE">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=MRK">MRK</A></B></TD><TD
class=yfnc_tabledata1 noWrap align=middle>Jun 3</TD><TD class=yfnc_tabledata1 noWrap align=right><B>32.09</B></TD><TD class=yfnc_tabledata1 noWrap align=right><IMG height=14 alt=Down
src="http://us.i1.yimg.com/us.yimg.com/i/us/fi/03rd/down_r.gif" width=10 border=0> <B style="COLOR: #cc0000">0.14</B></TD><TD class=yfnc_tabledata1 noWrap align=right><B style="COLOR: #cc0000">-0.43%
</B></TD><TD class=yfnc_tabledata1 noWrap align=right>5,714,800</TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=MRK&amp;t=1d"><IMG height=16
src="http://ichart.finance.yahoo.com/h?s=MRK" width=60 border=0></A></TD><TD class=yfnc_tabledata1 noWrap align=middle><A href="/q/bc?s=MRK">Chart</A>, <A href="/q/mb?s=MRK">Messages</A>, <A href="/q/ks?
s=MRK">Key Stats</A>, <A href="/q?s=MRK">More</A></TD></TR><TR><TD class=yfnc_tabledata1 noWrap align=middle><B><A href="/q?s=MON">MON</A></B></TD><TD class=yfnc_tabledata1 noWrap
```

Found initialization string for row

Dataset: STOCK_PRICES | Column Name:

Key:
- Start or End Tag - Field Value - Row Initialization String

[✓] Pause on possible errors

Results | Step-by-Step Replay

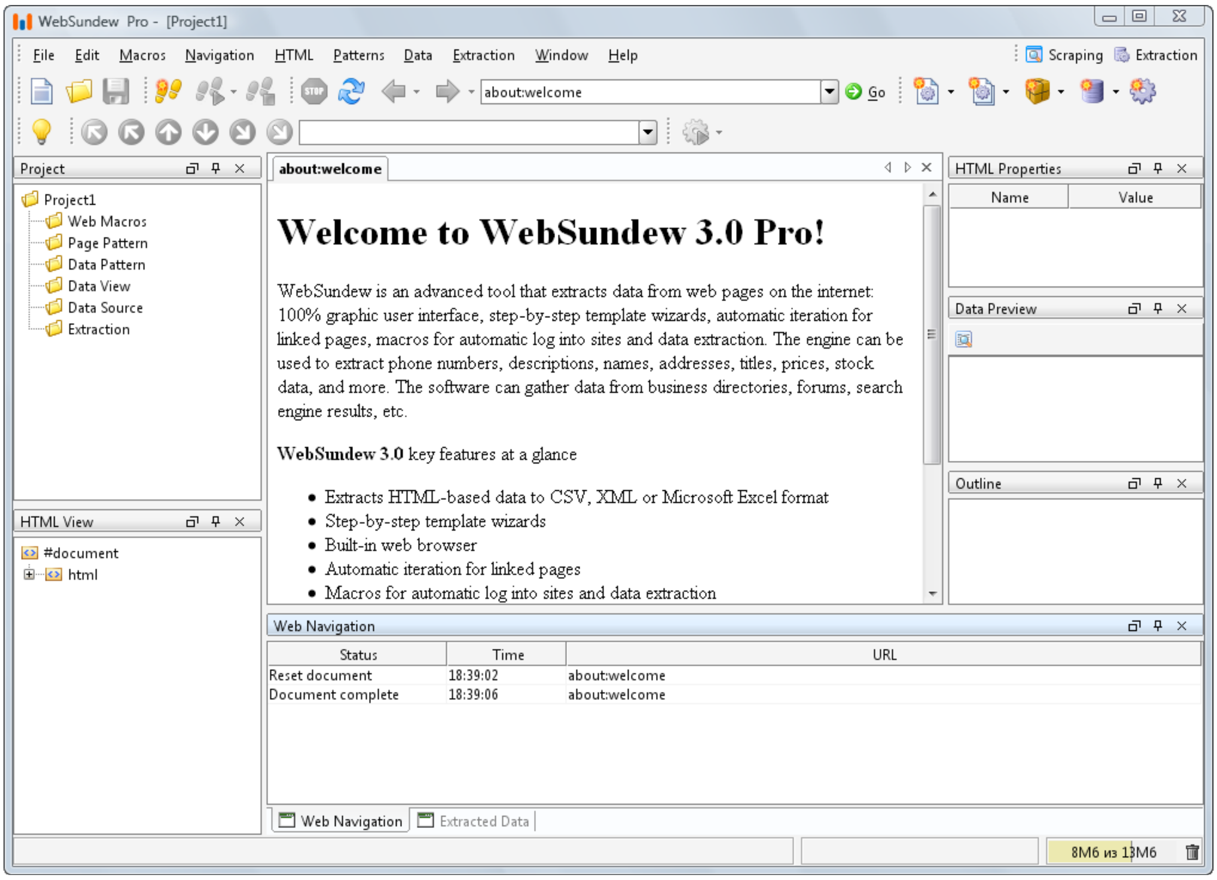
Figure 2.14: WSP+ (Screenshot 2)

2.5.3 WEBSUNDEW

WebSundew (SundewSoft, 2010) is advertised as "*an advanced web scraping tool*". The company prizes itself for the use of intelligent algorithms and fuzzy logic in the development of their products including WebSundew. A professional licence for this system costs $899.90.

Figure 2.15 shows the homepage of the system. WebSundew works by recording a macro of the user's navigation through the different web pages and structures within each page. Users are responsible for the selection of each item of interest on the web page (e.g. name, address, phone number etc.) and manually renaming the field names to which this data will be assigned (Figure 2.16).

A problem with the latter, however, is that users will need to understand the HTML tag path corresponding to each data item chosen, before being able to update the field name, because, as Figure 2.16 shows, it is not clear what information corresponds to e.g. Field_1. In this scenario, the user may remember the order in which the items on the page were selected, however this would become complicated if there were many more items to be extracted from a page and therefore many more fields. Thus, similarly to WSP+, knowledge of HTML tags is required.



Welcome to WebSundew 3.0 Pro!

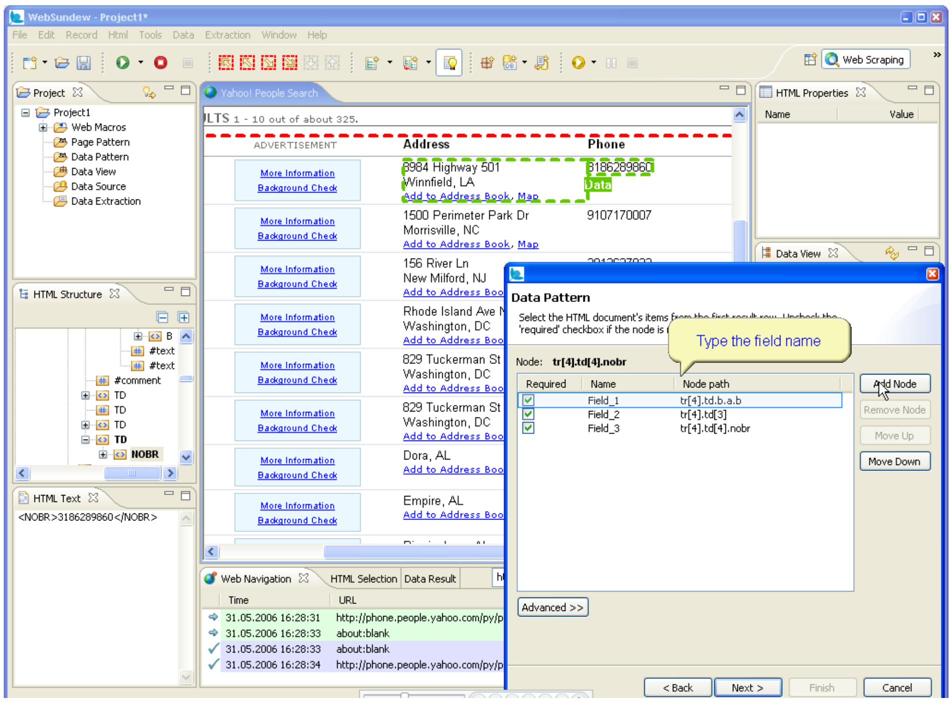
WebSundew is an advanced tool that extracts data from web pages on the internet: 100% graphic user interface, step-by-step template wizards, automatic iteration for linked pages, macros for automatic log into sites and data extraction. The engine can be used to extract phone numbers, descriptions, names, addresses, titles, prices, stock data, and more. The software can gather data from business directories, forums, search engine results, etc.

WebSundew 3.0 key features at a glance

- Extracts HTML-based data to CSV, XML or Microsoft Excel format
- Step-by-step template wizards
- Built-in web browser
- Automatic iteration for linked pages
- Macros for automatic log into sites and data extraction

Status	Time	URL
Reset document	18:39:02	about:welcome
Document complete	18:39:06	about:welcome

Figure 2.15: WebSundew (Screenshot 1)



<NOBR>3186289860</NOBR>

Data Pattern

Select the HTML document's items from the first result row. Uncheck the 'required' checkbox if the node is

Node: tr[4].td[4].nobr

Required	Name	Node path
✓	Field_1	tr[4].td.b.a.b
✓	Field_2	tr[4].td[3]
✓	Field_3	tr[4].td[4].nobr

Time	URL
31.05.2006 16:28:31	http://phone.people.yahoo.com/py/p
31.05.2006 16:28:33	about:blank
31.05.2006 16:28:33	about:blank
31.05.2006 16:28:34	http://phone.people.yahoo.com/py/p

Figure 2.16: WebSundew (Screenshot 2)

Users are also required to choose the elements that link a page to the following page to be extracted and configure the data source into the corresponding data view for the chosen data storage (data can be stored either as an XML file, text file, or Excel spreadsheet). At this stage, the system is still not ready to be executed. The extraction sequence needs to be first setup by choosing all the different system components prepared so far, such as the navigation macro, the data source and the 'next page' link information. Once all of this is completed and saved, WebSundew can be executed.

2.5.4 LIMITATIONS AND DISADVANTAGES

The above three WIE software systems share advantages such as:

- They are not limited to one interest domain, instead they work for websites from different areas e.g. news websites, car advertisements etc.
- They give professional users the ability to control and set up the flow of the extraction process.
- They support several types of data storage.

However, they rely heavily on the users' involvement and although they provide guides through the many extraction steps involved, these systems are far from being straightforward. The users' participation is required in every aspect of the extraction process including:

- Navigating to the correct web page
- Highlighting the area(s) to be extracted
- Evaluating the output from the extraction process etc.

In contrast, the system in this research runs in the background, extracting information from websites with minimum user involvement in the process. As a matter of fact, the users' involvement is only required when training the system. The extraction process itself is fully automated.

Another limitation of these software packages is that users need to have at least basic knowledge of HTML tags and structures to be able to use the packages successfully and efficiently. In this research no technical knowledge is required from users.

The biggest limitation of these systems is that users need to know the websites from which they want to extract information in advance. This is very limiting because it does not give the users the option to discover new websites. Furthermore, all the extraction steps need to be followed for each website under investigation. Taking into consideration the large number of websites available and their ever-changing nature, this could easily become a long, expensive and dreaded undertaking.

2.6 SUMMARY

This chapter discussed past and present research related to the areas considered for the research presented in this thesis. Justification was given for the approaches chosen for the final system. Some commercial WIE systems were also investigated and their disadvantages and limitations were disclosed. The following chapter concentrates on the different methodologies explored for this research and the choices made in support of the aims and objectives of the project.

3 RESEARCH METHODOLOGY

The scope of this chapter is to investigate different methodologies suitable to the project and compare them against each other, highlighting their strengths and weaknesses in relation to the research specification. The choice of the research methodology for this project is then justified as an important step to supporting the aims and objectives outlined in Chapter 1.

3.1 RESEARCH STRATEGY

Research can be categorised as Qualitative or Quantitative in relation to the data types required and the data collection methods used.

3.1.1 QUALITATIVE METHODS

Qualitative methods are concerned with identifying patterns in the data through evaluating not only the data itself, but also the methods used for the data gathering and analysis. The latter are important, because with certain types of data, the collection methods chosen can really affect the results. For example, data on the quality of a particular service would depend on the type of people asked, the number of people asked, the type of questioning method i.e. interviews vs. questionnaires etc. Additionally, the quality of collection methods needs evaluating e.g. the quality of questions in a questionnaire or interview, the variety of questions asked etc.

Qualitative methods are useful in better understanding human behaviour and the reasons behind decision making, which makes them particularly appropriate for social sciences. Their main disadvantage however, is that it is possible for researchers to be influenced by their subjects' opinions and thus not being able to separate their personal beliefs from those of others. An extreme case is what is known as 'go native', where researchers adopt a participant role unknowingly (Lee et al., 1997). Another disadvantage of this approach is that researchers may fail to maintain objective views of the data, by becoming too attached to the problem and by using personal 'hunches' to guide their thinking process (Cormack, 1991).

Qualitative methods are not as common in Software Engineering projects (Glass et al. 1994). Some researchers, however, have attempted to show the importance of such methods in technical projects. For example, Sharp et al. (2005) used Qualitative Methods to uncover any potential, non-technical attributes that could have affected the evolution of software quality management systems in their work. Dubinsky et al (2006) and Hazzan et al. (2006) also published their support for Qualitative Methods, particularly for the purpose of expanding the research scope of a project and broadening their understanding of specific processes and project findings.

The main data collection methods used in qualitative research are: questionnaires, semi-structured interviews, unstructured interviews or conversations, focus groups as well as observations and case studies. The researcher is in charge of analysing and interpreting the data collected. Informal interviews and conversations with participants are discussed in more detail in section 3.4, as they are part of the adopted methodology for this research.

3.1.2 QUANTITATIVE METHODS

Quantitative methods attempt to analyse data in measurable terms, thus they are best suited to assist the development of quantifiable information. These methods use deductive reasoning from existing knowledge to prove a theory. Unlike qualitative methods, where researchers are driven by certain ideas, other peoples' opinions or even instincts, quantitative methods require researchers to focus on the data itself and produce scientific results.

Quantitative methods investigate hard, numerical and reliable data. Surveys, quantitative questionnaires and structured interviews are amongst the techniques used in quantitative research. These, however, like all empirical approaches, consist of entirely quantifiable questions. The exact set of questions is asked to every individual in the population under consideration. "Closed format" questions are a common occurrence in such surveys, whereby a question has a pre-specified number of answers, and as such, it does now allow participants to elaborate on answers using personal experiences and judgement. Experiments are also common quantitative techniques. These are characterised by random assignment of subjects to experimental conditions and the use of experimental controls. Statistical methods are used for the analysis of the data collected.

Unlike qualitative methods, researchers using quantitative methods can easily maintain a detached and objective view of the data under investigation, which increases the study's reliability. Some sources however, see this as a weakness of this approach (Spencer, 1983; Cormack, 1991), stating that quantitative methods treat people like objects, dismissing their experiences. This may be a valid criticism, however it does not apply to this research, as the data under investigation is obtained from the Web, not from human participants.

Qualitative and quantitative methods are very different from each other and they both have strengths and weaknesses, thus no method can be thought of as being superior to the other. Each approach needs to be considered in relation to the research, to which it will be applied. A combination of both approaches has also been proven beneficial in many research works (Dzurec & Abraham, 1993; Tashakkori & Teddlie 2002; Creswell, 2003; Armitage, 2007).

The main methods used in this research were Quantitative Methods, as the project is technical and needs hard, reliable data. Qualitative Methods however, were used in the early stages of the project when capturing the user requirements and at the end of the project, when evaluating whether or not the original requirements were met. Experiments and Quantitative Questionnaires were chosen as part of the adopted methodology for this research and are discussed in more detail in section 3.4.

3.2 METHODOLOGICAL CONSIDERATIONS

The goal of a research project can take different forms depending on what the researcher is trying to achieve. Most research projects can be grouped in the following categories (Avison & Fitzgerald, 2003):

- **Exploratory Research** – Includes research projects, which concentrate on problems that have not been clearly identified before. This type of research may sometimes conclude that the problem does not exist, e.g. the researcher's hypothesis is proven to be false. Exploratory research cannot usually be used to generalise large datasets.

- **Constructive Research** – Includes research projects that develop concrete solutions to clearly defined problems. If previous solutions already exist, then constructive research needs to prove how the new solution is different from or better than the old ones. This type of research cannot conclude that the problem cannot be solved. On the contrary, the accuracy of the solution has to be validated. The solution cannot just be a scientific result; it needs to have a practical relevance.

- **Empirical Research** – Includes research projects where an initial research question is transformed into a theoretical model, based on the analysis of available literature, observations and experimentation. The theoretical model helps the research question to be tested using empirical observations.

This research falls into the Constructive Research category, because it aims to produce an operational prototype, which will be used by a real organisation. The research conducted will help to solve problems that are currently wasting ATM time and money.

In relation to the main objective of this research, i.e. the design and development of an automated web extraction system, various system development methodologies were considered. A system development methodology, or otherwise known as a software development methodology, refers to the framework used to structure and control the process of developing consistent and standard software applications and/or IT systems (CMS, 2005). A variety of system development methodologies have evolved through the years (Scacchi, 2001), each with its own strengths and weaknesses. These include methodologies like: Procedural Programming, Rapid Application Development, Object-Oriented, System Development Life Cycle (SDLC), Rational Unified Process, Prototyping etc.

Research has shown that system development methodologies are most efficient when tailored to each particular project (Touil & Ahmed-Nacer, 2006; Karlsson & Hedstrom, 2009). This means that instead of following a specific methodology entirely, it is advised to revise an existing methodology or to combine two or more methodologies together to benefit from their joint attributes. The requirement specification for this research, as well as a review of the strengths and weaknesses of the available methodologies, aided the decision to select Prototyping and Object-Oriented for this research. The following sections provide details on the reasons behind the decision to choose these methodologies:

3.2.1 PROTOTYPING DEVELOPMENT METHODOLOGY

Prototyping is a system development methodology, in which a prototype (a version of the final system) is developed, tested and evaluated and then re-developed if and as necessary until an acceptable prototype is created, from which the final system can be developed (Worth & Greenough, 2005).

The stages included in this methodology are shown in Figure 3.1. The Prototyping methodology is popular amongst researchers (CMS, 2005; Tedmori, 2008) because it is particularly fitting when creating fundamentally new software. In these cases, not all of the system requirements are known in advance, thus, this methodology allows developers to experiment with different versions of the system in order to improve the requirements and create a complete software package. Prototyping also allows for a closer involvement of the users. Refining requirements throughout the different versions of the system and involving the sponsoring company throughout the process were fundamental to this research, thus prototyping was appropriate for this project.

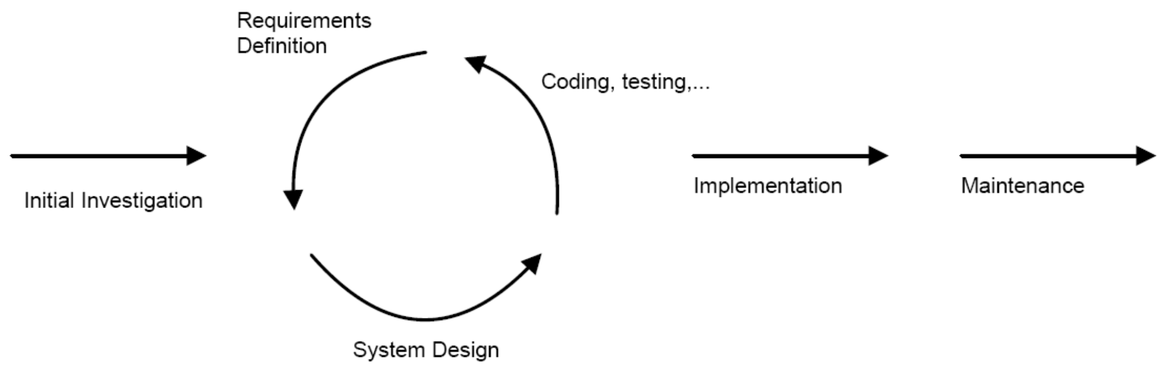


Figure 3.1: Prototyping (CMS, 2005)

3.2.2 OBJECT-ORIENTED DEVELOPMENT METHODOLOGY

Object-oriented (OO) development is "a partial-lifecycle software development methodology, in which the decomposition of a system is based upon the concept of an object" (Booch, 1986). Indeed, an OO program can be viewed as a collection of interacting objects, as opposed to the conventional model – Procedural Programming – in which a program is viewed as a sequence of tasks to perform. In OO development, an object is an instance of an entity, known as a class. The class defines the abstract characteristics of all its objects, including their properties and behaviour. Objects are capable of receiving from, processing and sending messages to other objects. Other fundamental concepts in OO development, useful to this research, include:

- **Inheritance** – inheriting classes can make use of all the properties and behaviours declared in the parent class, without the need to re-declare them.
- **Encapsulation** – otherwise known as 'Information Hiding', it is the principle of segregation of design decisions in a computer program that are most likely to change, thus protecting other, more stable parts of the program from changes.
- **Polymorphism** – if the implementation of a method in the parent class is not suitable or it is too general, then the declaration of this method can still be kept in the parent class, however the inheriting class is allowed to override the method by providing a different implementation.
- **Abstraction** – this concept simplifies complex scenarios by modelling only the classes that are most appropriate for the problem at hand, whilst ignoring the remaining classes.

The above concepts formed an important part of the development of the prototype in this research. Particularly, Inheritance and Encapsulation were greatly used. Inheritance was necessary as it helps to program more efficient code, whereas Encapsulation was especially useful for the WIE part of the system, where the more stable functionalities of the genetic program, such as parent selection, mutation, crossover etc., were 'encapsulated' from the rest of the functionalities, such as: the genotype to phenotype mapping, in order to protect them from any potential changes to the latter in the future.

The powerful ability of the OO development methodology to model complex, real-life situations, has transformed this methodology into a very sought after approach to programming (Hodgett, 2003; Ramsin, 2006; Booch et al. 2007; Kamandi & Habibi, 2008; Ramsin & Paige, 2008).

3.3 ADOPTED RESEARCH METHODOLOGY

The adopted research methodology for this research incorporates the three main stages of system development: Concept Development, System Development and System Evaluation (Burstein, 2002). The three stages do not occur in sequence. Instead, they are more dynamic and interactive in nature and as such they may be carried out concurrently and/or may be revisited whenever necessary. Each stage has clear objectives and outcomes, however there are also overlaps amongst them, which means that it was possible to work towards different stages of this project simultaneously.

Table 3.1: Research Stages and Processes

Stage	**Action/Process**
Concept Development	Investigation ▪ Investigate research area ▪ Investigate business processes at sponsoring company ▪ Define business/user requirements ▪ Produce Literature Review Analysis ▪ Analyse requirements ▪ Analyse domain area Design ▪ Translate business/user requirements to system requirements
System Development	Investigation ▪ Investigate programming languages and data storage solutions ▪ Investigate possible existing solutions Programming ▪ Develop physical database ▪ Develop system functionality Testing ▪ Perform Procedure and Module testing ▪ Perform Integrated Modules testing
System Evaluation	Evaluation ▪ Evaluate aims and objectives ▪ Evaluate data samples used in experiments ▪ Evaluate approach and results ▪ Evaluate user acceptance ▪ Determine research contributions ▪ Determine the research implications to the sponsoring company and the wider industry ▪ Evaluate overall solution

Each stage involved various actions/processes such as: investigation, analysis, design, programming, testing and evaluation. Table 3.1 shows a breakdown of the processes involved in each stage of this research. The main outcomes of the Concept Development stage included: outlining the project's aims and objectives and producing a literature review of the research area (discussed in Chapters 1 and 2). This assisted the choice of Methodology, which has been the subject of this chapter. The following chapters discuss the work involved and achievements obtained from the System Development and System Evaluation stages.

3.4 TECHNIQUES/TOOLS USED

This section concentrates on the tools and techniques used in this research. The choice of research techniques (i.e. Literature Review (LR), Quantitative Questionnaires (QQ), Experiments (EX), Semi-structured Interviews (SSI) and Unstructured Interviews (UI)) was made in relation to the research objectives they support, the actions/processes affected (from Table 3.1) and the specific associated tasks carried out to satisfy each objective (Table 3.2). Table 3.2 also shows the development methodologies used (i.e. Prototyping (P) and Object-Oriented (OO)) and the research outputs, identified in the form of publications and the EngD thesis.

Table 3.2: Research Road Map

Objectives	**Associated Tasks**	**Actions/Processes**	**Method**	**Output**
1 – Review related work in the field of WIR and WIE.	1 – Review past and present research in the areas of WIR and WIE. 2 – Review any existing, operational solutions and their limitations.	- Investigate research area - Investigate possible existing solutions - Produce Literature Review - Define business/user requirements	LR	Paper 1
2 – Analyse ATM's business processes and project requirements.	3 – Determine the issues encountered at ATM. 4 – Investigate existing processes for finding training course information at ATM.	- Investigate business processes at sponsoring company - Define business/user requirements	SSI UI QQ	Paper 1
3- Model database to store the results from each stage of the system 4 – Automate the **Retrieval** of web pages from the Web.	5 – Decide on the programming language and database system to be used. 6 – Design and build the database 7 – Design and develop a crawler to retrieve and clean links from the web.	- Investigate programming languages and data storage solutions - Translate business/user requirements to system requirements - Develop physical database - Develop system functionality - Perform Procedure and Module testing	LR EX P OO	Paper 2

	8 – Test system.	- Perform Integrated Modules testing		
5 – Automate the **Classification** of the previously retrieved pages into *relevant* and *irrelevant* categories.	9 – Design and develop a Classification system.	... Same as objectives 3 & 4	LR EX P OO	Paper 2
	10 – Update database structure.			
	11 – Test system.			
	12 – Evaluate results and compare classifier with other classifiers.			
6 – Automate the **Extraction** of training course attributes from the relevant web pages into ATM's database.	13 – Update database structure.	... Same as objectives 3 & 4	LR EX P OO	Papers 3, 4 & 5
	14 – Implement system that extracts training course details from web pages.			
	15 – Test system.			
	16 – Collect and evaluate results.			
7 – Automate system execution to run at scheduled times.	17 – Investigate scheduling systems.	- Investigate possible existing solutions - Perform Integrated Modules testing	LR P OO EX	Thesis
	18 – Integrate all three systems developed in objectives 3, 4 and 5 and implement system scheduling.			
8 – Critically evaluate the project.	19 – Evaluate the overall system.	- Evaluate aims and objectives - Evaluate data samples used in experiments - Evaluate approach and results - Evaluate user acceptance - Determine research contributions -Determine the research implications to ATM and the wider industry - Evaluate overall solution	UI SSI EX	Paper 5 & Thesis
	20 – Explore user satisfaction.			
	21 – Recommend further improvements.			

Note that the Associated Tasks in Table 3.2 represent a general list of the tasks involved in this research. Each of these tasks was broken down to further subtasks during the different stages of the project. Particularly, the tasks related to the implementation of the three different systems (tasks 3, 4 and 5) involved other subtasks concerning the different programming decisions and challenges that needed to be achieved. These additional details will become apparent in the following chapters.

The following sections in this chapter give a small introduction to the different data collection methods chosen for this research.

3.4.1 Literature Review

As the name suggests, a literature review surveys past and present scholarly articles, books and other sources relevant to a particular research area or issue, giving a critical evaluation of each work (Cooper, 1998). The aim of the review is to set the current research project within a conceptual and theoretical context and identify relationships between this and past work in order to clearly demonstrate any contributions to the research community and prevent duplication of effort (Wisconsin-Madison, 2009). Furthermore, literature reviews aid the process of choosing the correct methodology for the project.

The literature review for this research was carried out throughout the research period, using both academic and industrial literature. Technical areas under scrutiny included the different approaches to Web Classification and Web Extraction, including NB Networks and Genetic Programming principles, as discussed in chapter 2. Technologies behind search engines as well as some existing WIE solutions were also examined. All this enabled the identification of key weaknesses in the existing retrieval and extraction approaches and assisted, therefore, in the implementation of the prototype developed in this research.

3.4.2 Quantitative Questionnaires

Quantitative questionnaires are the same in structure and format as qualitative questionnaires. The difference rests on the type of questions asked. Quantitative questionnaires only include questions that have quantifiable answers. For example, words “easy” and “difficult” may mean different things to different people, thus are considered ambiguous. Quantitative questionnaires are carefully created to avoid such ambiguity (Parr et al. 2007).

Quantitative questionnaires are mainly constructed with ‘closed format’ questions. These usually take the form of multiple-choice questions, which are easier for the participants to answer and for the researcher to analyse. This means that quantitative questionnaires are less costly to administer. ‘Closed format’ questions also prevent participants from elaborating on answers using personal experiences and judgement. In certain cases, this may be considered as a negative view (Harris, 2006) as it decreases the likelihood of receiving unexpected and insightful suggestions. However, in other cases (Shaeffer et al., 2004), personal experiences and judgements may confuse the situation and result in less definitive answers.

In this research, quantitative questionnaires were used during the requirements gathering stage of the project. The questionnaire targeted the advisors at ATM, in order to collect their views regarding the existing difficulties and problems at ATM in relation to finding and collecting information from the Web, and their requirements for various aspects of the final system, such as different functionalities and usability. ‘Closed format’ questions were favoured to ‘open format’ ones, because it was believed that these types of questions focused the participants’ attention on the questions better and required less time to complete. Example questions included: choosing the most appropriate solution(s) (from a given, finite list) to the existing problems related to course searching; identifying the amount of time each advisor would like to ideally spend on searching for courses; identifying the amount of user involvement each advisor thinks appropriate for obtaining course information, etc. Additional methods were also used at this stage, such as: semi-structured and unstructured interviews (discussed in section 3.4.4) to put the advisors at ease.

Note that the number of advisors at ATM, from the start of the EngD research in July, 2006 to present, has reduced from eight to just two. For this reason, questionnaires were only used to help the gathering of the initial requirements, rather than evaluate results, as that would have been too small a sample to be of much evidence.

3.4.3 EXPERIMENTS

Experiments form an important tool in research projects. They are used to test existing theories, new hypotheses or other important research questions. They need to be formulated correctly, however, and control any possible confounding factors in order to ensure the validity of the results (Fletcher, 1995).

In this research, experiments were used during the System Development and System Evaluation stages, as these were responsible for the design, development and evaluation of the final prototype. Experiments were set up not only to test and support the ideas discussed in this thesis, but also to compare them to existing or similar research techniques. For example, experiments were conducted to test the NB classifier not only against existing, standard NB classifiers, but also against other types of classifiers such as NNs and DTs. These experiments and others are described in chapter 5 and the included papers.

3.4.4 SEMI-STRUCTURED & UNSTRUCTURED INTERVIEWS

Informal interviews, or otherwise known as Unstructured Interviews, take the form of controlled conversations between the researcher and the participants. There is not a strictly fixed list of questions that is asked during such interviews. The researcher may choose to add or remove questions from their list of pre-prepared questions based on the participants' answers to previous questions. The researcher is always in control, however, of the direction of the interview.

Conversations with Participants, also referred to as chats, are even more relaxed. These aim to put participants at ease, in order to truly discover their thoughts and opinions on the matter at hand. Some people have the tendency to become timid and withdrawn under interview conditions and this may prevent them from revealing exactly how they feel about a certain product or service. Conversations are ideal for these types of people.

Semi-structured and Unstructured Interviews and Conversations with Participants were chosen for both the initial stages of the research, especially for the gathering of the requirements, and the evaluation of the final prototype. They were chosen, because it was important to get the undivided opinions of the advisors and the Managing Director at ATM, regarding the problems with their existing methods of managing training course information, as well as their thoughts on the functionalities and results obtained from the final prototype. Each type of interview was carried out with one participant at a time, in order to get everyone's personal view without them being influenced by others at the company.

3.4.5 DEVELOPMENT TOOLS

A major part of this research involved programming, thus one of the very first tasks was to select a programming language, as well as appropriate data storage for the results.

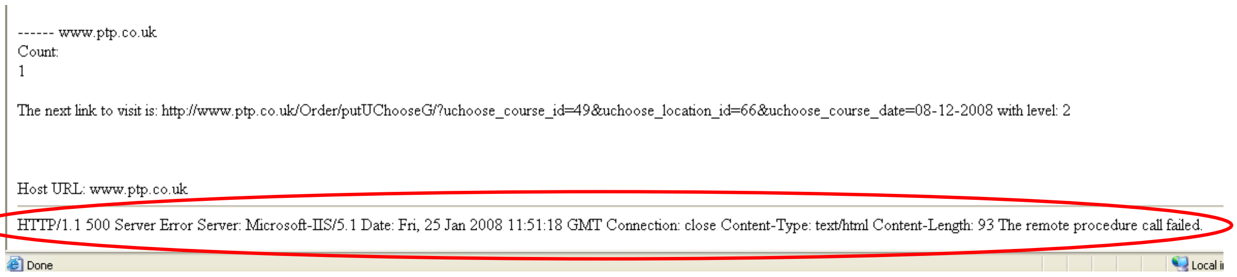
The decision on data storage was made easier by the fact that one of the project requirements was to integrate the results of the new system with the database behind the CRM package used, which is a Microsoft (MS) SQL Server 2000 database.

In relation to programming languages, out of the many languages available, it was decided that the final choice was between PHP, the supervisors' recommendation, and the .NET framework, with which the author was most familiar. PHP was initially chosen for the development of the search engine, because of the following reasons:

- PHP is free and open-source, which means that new releases and patches come out regularly. Languages that are not open-source, such as VB.NET, follow rigid processes in fixing bugs, thus roll-outs or patch releases can be considerably delayed.
- PHP was designed with the Web in mind, therefore it is very efficient and it contains a rich selection of inbuilt commands to deal with various aspects of the Web. Some of these commands, e.g. *get_meta_tags()*, do not exist in .NET and need to be implemented through many lines of code.

Both the search engine and the classification system were developed in PHP, however, despite both systems being efficient at the beginning, limitations of PHP started to become apparent, particularly when handling connections and communications with MS SQL Server.

An error message (Figure 3.2) occurred during the crawling of a large site, which comprises over 4800 web pages. The initial concern for the error was a potential memory leak that was consuming the overall Random Access Memory (RAM) and causing the system to resort to using the Virtual Memory to store the temporary data during execution. However, after many tests and many error catching procedures, the idea of a memory leak was rejected.



```
------ www.ptp.co.uk
Count:
1

The next link to visit is: http://www.ptp.co.uk/Order/putUChooseG/?uchoose_course_id=49&uchoose_location_id=66&uchoose_course_date=08-12-2008 with level: 2


Host URL: www.ptp.co.uk
HTTP/1.1 500 Server Error Server: Microsoft-IIS/5.1 Date: Fri, 25 Jan 2008 11:51:18 GMT Connection: close Content-Type: text/html Content-Length: 93 The remote procedure call failed.
```

Figure 3.2: Server Error Message

Further research into the problem showed that many other programmers had experienced similar problems and that it could be an MS SQL Server – PHP connectivity issue, where PHP does not close connections well, resulting in memory not being released after use. Some of the solutions posted on Internet forums were unsuccessful. It was noticed however that after the error occurred, a simple browser refresh resumed the program and because the program was written to record all visited sites, the refresh restarted the program exactly where it was interrupted. It was also observed that the error message was repeated after the visiting of around 700 links. An automatic refresh was, therefore, built into the program, which refreshed the web page at a rate determined in the database settings (e.g. every 700 links). Extensive testing of the new functionality resulted in no more errors and the uninterrupted running of the program.

The above does not work out the cause of the problem, however it does resolve the issue. This solution also strengthens the program, as it means that whatever the number of web pages being crawled, the system will always complete the crawling process, even for machines with limited RAM, as each refresh cleans all the previous memory allocations.

The final stage of the project – the development of the automated extraction system – asked for a revisit of the strengths and weaknesses of PHP and the .NET framework. The decision to enhance genetic programming principles to optimise web extraction required a language that was powerful enough to manage thousands of population generations and large volumes of data. The above PHP problem and other reasons listed below, justified the need to change programming languages and opt for the .NET framework, specifically VB.NET, for the last part of the project:

- VB.NET is fully Object Oriented, and as such it supports inheritance, encapsulation, polymorphism etc. PHP is a scripting language, thus despite it being powerful, it is not a pure Object-Oriented language.
- The .NET Class library (of Base Class Library) is available to all the .NET languages, resulting in a consistent model regardless of the programming language used.
- Debugging is very efficient as .NET supports runtime diagnostics, which help to track down bugs and determine how well an application is performing. VisualStudio.Net also provides excellent features for debugging applications such as: breakpoints, tracing of sections in the code etc., making programming easier and faster. PHP offers none of these, so debugging is time consuming.
- PHP, unlike VB.NET, is less consistent and lacks standardised structures for catching exceptions or for error handling, leaving programmers to have to code error handling techniques themselves. (PHP 5 has touched upon this issue and it now includes the ability to use some form of the Try…Catch and Throw structure).
- .NET offers straightforward Application Deployment and Maintenance. The installation process requires only that the application and its components are copied into a directory in the target machine.
- .NET has improved the way that code is shared between applications, introducing the concept of assembly, which replaces the traditional DLL. Assemblies are the .NET unit of deployment, versioning and security and different versions of assemblies can exist side by side.

The development of the WIE part of the project in VB.NET, specifically the part of the system dealing with the extraction of training course names, required the reuse of the NB approach, developed previously, for the fitness evaluation of the Regular Expressions evolved (see section 4.4.2.5). This made it necessary to convert the NB system to VB.NET to be compatible with the GP system. Results from testing both the PHP and the VB.NET versions of the classification part of the prototype are compared in chapter 5.

3.5 SUMMARY

This chapter investigated the different methodologies currently available and justified the decision made on the methodologies most appropriate for this project. The specific research methods and development tools adopted in this research were also discussed in this chapter. The main aspects of the project are explained in detail in the following chapter.

4 RESEARCH AND DEVELOPMENT

4.1 INTRODUCTION

This chapter concentrates on the design and development work, focusing on both the back end (database structure) and front end (program) of the prototype. Technical details of the two main approaches chosen, Naïve Bayes Networks and Genetic Programming, are also presented. Diagrams are used better to illustrate the system's functionality.

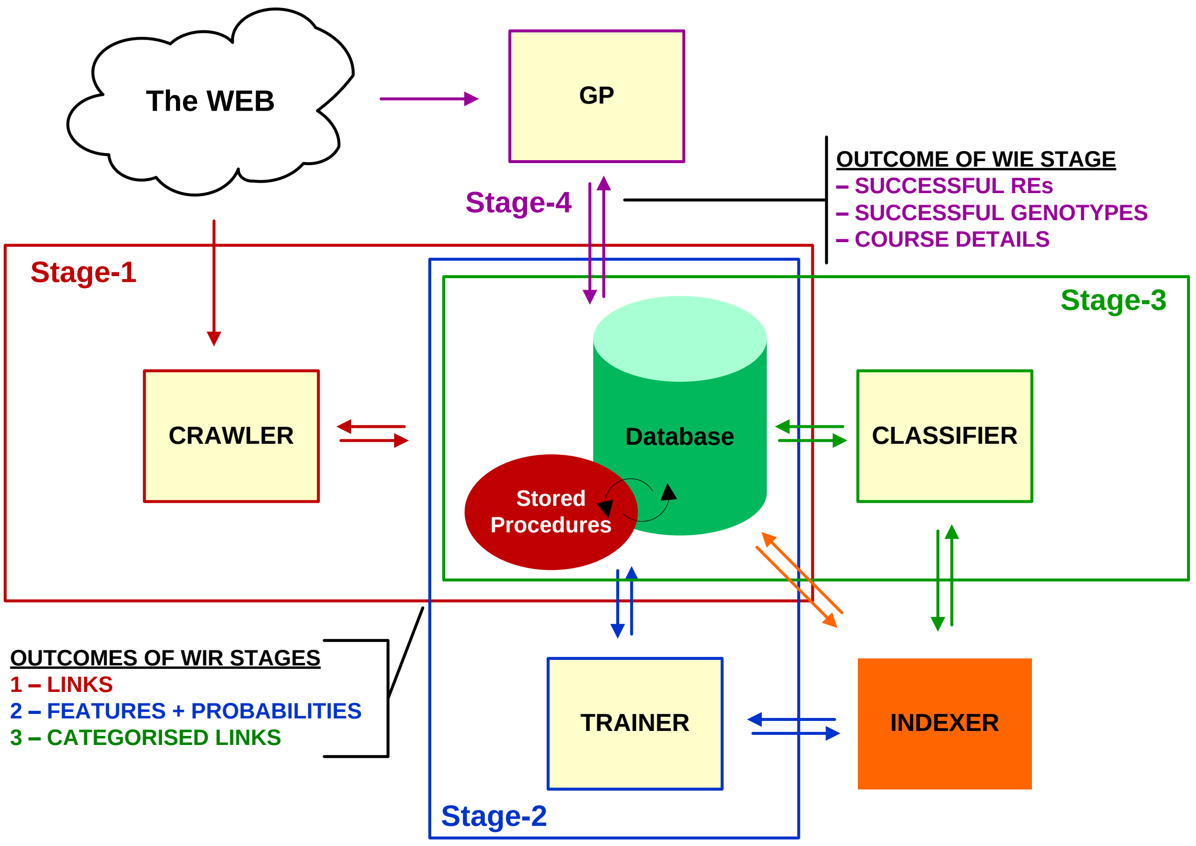


Figure 4.1: Overall System

ATM currently uses its CRM package as the front-end to all details about their clients including their needs and behaviours as well as the different courses available. Therefore, the prototype developed as part of this research serves as a mediator between the Web and CRM, collecting course information from training websites and feeding them to CRM through ATM's database, whilst guaranteeing the accuracy of the courses already stored in the database. Figure 4.1 shows the different stages involved in the prototype together with the outcomes from each stage.

4.2 DATABASE STRUCTURE

The database plays a significant part in the system's functionality. It is used not only to store the final training course information, but also to store intermediate data that is important to the success of the different parts of the system. For example, information about the web pages retrieved and the main keywords extracted from these pages is crucial for the calculations performed by the Naïve Bayes (NB) classifier developed during this research. Thus, the design and development of the database structure was one of the main tasks of this project.

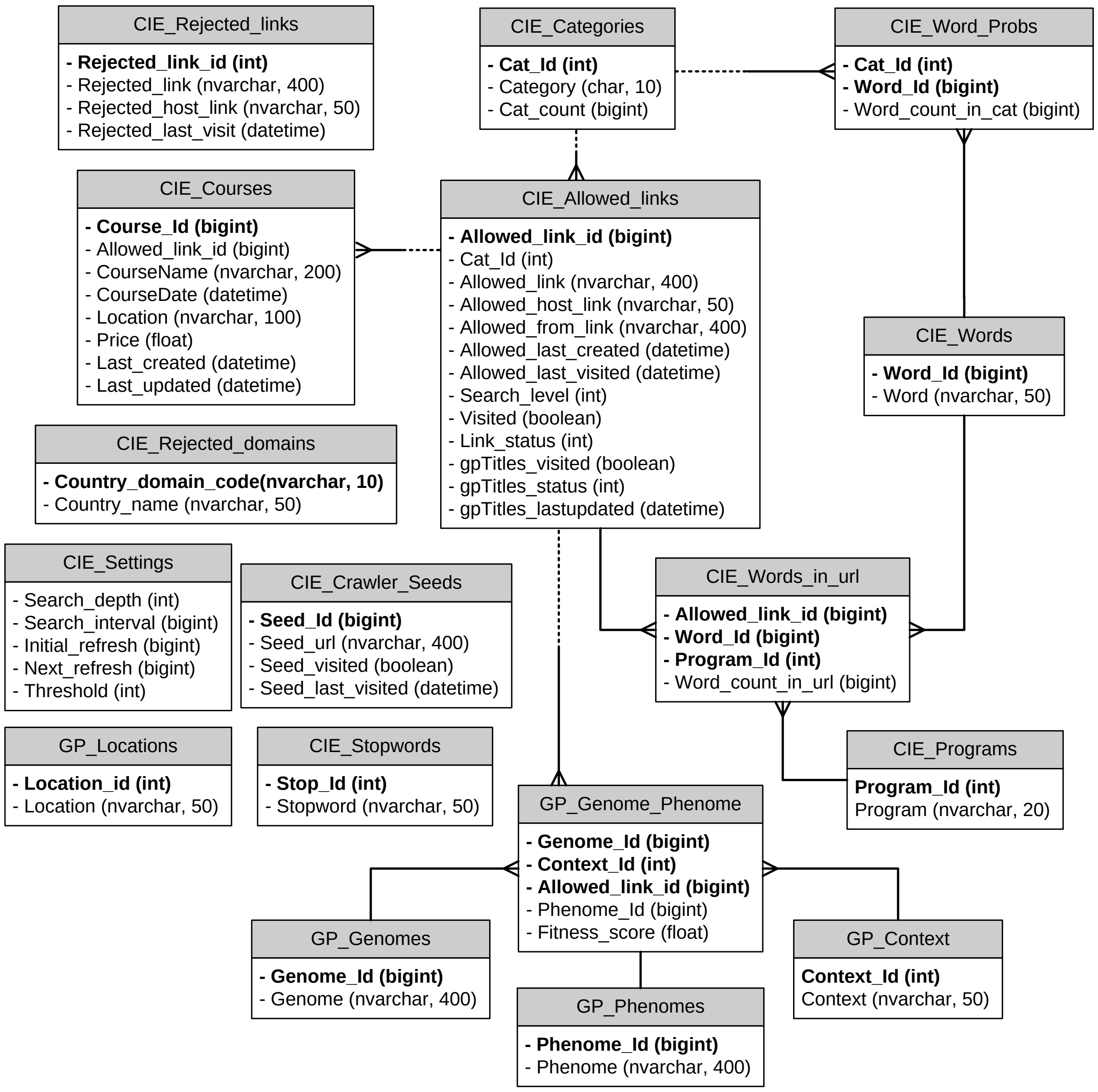


Figure 4.2: Database Structure

The database consists of 17 tables which store information about all the different parts of the system. Figure 4.2 shows the tables and the relationships between them. Two other database structures that were considered for the WIR part of the system are discussed in Appendix B.

A brief description of each database table and their roles is given in Table 4.1. Tables CIE_Settings, CIE_Rejected_domains, CIE_Rejected_links, CIE_Stopwords, CIE_Crawler_Seeds and GP_Locations hold unique information; there are no relationships to connect the data stored in these tables, as none of them rely on each other's attributes for information. All tables have been normalised to Third Normal Form (3NF) (Dawson, 1997).

Fields “Word_count_in_url” and “Word_count_in_cat” are calculated fields, and as such they should not be part of a normalised database. They could be calculated each time they are needed by the system, however this would have a great impact on the system’s execution speed and performance, thus it was decided to store them in the database.

Fields “Initial_refresh” and “Next_refresh” in table CIE_Settings were created to overcome the server related error discussed in section 3.4.5.

Table 4.1: Database Tables

Table	Description
CIE_Settings	This table stores settings needed by the system, so they are not hard-coded in the program. Administrators can change the settings to alter the program as required. E.g. *Search_depth* stores the depth in websites, to which the crawler is allowed to continue.
CIE_Rejected_Domains	ATM is interested in courses within the UK alone, thus websites from foreign domains are ignored from the search.
CIE_Courses	This table stores the final results of the system, i.e. training course information such as course title, price(s), date(s) and location(s). This information is then displayed to ATM advisors through CRM.
CIE_Rejected_links	Links found to be entirely irrelevant are stored here, so they are excluded from future searches. However, in order to cope with system errors, these links are rechecked at regular intervals.
CIE_Stopwords	This table stores words that are too common or too insignificant to distinguish web pages from one another.
CIE_Crawler_Seeds	As the name suggests, this table stores the initial seed URLs for the crawling process (see section 4.3.1 for details).
CIE_Allowed_links	This table stores details about the links found by the Crawler, after ignoring those found in CIE_Rejected_links and CIE_Rejcted_Domains. The *Search_level* shows the depth of each web page in that website. *Link_status* records the status of each link, e.g. status: 2 means that the link has been used to train the Classifier. *Cat_Id* shows the category, in which the web page is finally classified. *Visited* helps the system to avoid crawling the same website twice. The last three attributes relate to the WIE part of the system, specifically to the extraction of course titles. This is due to this part of the system also using Naïve Bayes to evaluate title REs (see section 4.4.2.5 for details).
CIE_Categories	This table stores the different categories in which web pages are classified. It also stores the total number of links classified in each different category.

Table	Description
CIE_Words	All unique, stemmed words extracted from the INDEXER are stored in here, for future analysis.
CIE_Words_in_url	This table stores the words found in each web page and their frequency rate. This table sorts out the many-to-many relationship between tables CIE_Allowed_links and CIE_Words. *Program_Id* is needed to distinguish between the features related to the classification process and those related to the title extraction process.
CIE_Word_Probs	This is very useful for the classification process, as it stores the number of times each word has appeared in each category.
CIE_Programs	As previously mentioned, this table aids with the distinction between the classification process and that of course title extraction.
GP_Genomes	This table stores all the successful genotypes evolved by the system.
GP_Phenomes	This table stores all the successful REs (phenotypes) evolved by the system.
GP_Genome_Phenome	This table sorts out the many-to-many relationship between tables GP_Genomes and GP_Phenomes. Different REs can be applied to the same link, thus, this table stores the corresponding REs and their fitness scores for each genotype-link-context combination.
GP_Context	The context allows the system to know the types of course attributes, for which each evolved RE was successful e.g. price, title, date etc.
GP_Locations	This table is useful to the GP part of the system, particularly during the fitness test (see section 4.4.2.5 for details).

4.3 SYSTEM FUNCTIONALITY – WIR

This section focuses on the stages of the WIR part of the system, specifically on the Crawler and the components of the Classifier (Figure 4.3). More details of these components and the results from this part of the system can be found in Xhemali et al. (2009).

In Stage-1, the CRAWLER was developed to find and retrieve web pages after being seeded with an initial five training course web pages. The results of this stage – list of URLs – are stored in table CIE_Allowed_links. In Stage-2, a subsystem – the TRAINER – was developed to train the classifying process. The TRAINER uses the results of the first stage as input and stores feature-category associations for each link analysed in table CIE_Word_Probs. The training results are then used in Stage-3 by the CLASSIFIER, which assesses the ‘knowledge’ accumulated during training and uses this to make intelligent decisions when classifying new links. The CLASSIFIER expands the system's 'knowledge' with each new classification.

The second and third stages have a very important sub-stage in common – the INDEXER. This is responsible for identifying all relevant keywords/features used in web pages. The INDEXER also applies rules to reject HTML formatting and features that are ineffective in distinguishing web pages from one-another. A detailed description of each of the above stages and sub-stages is given below.

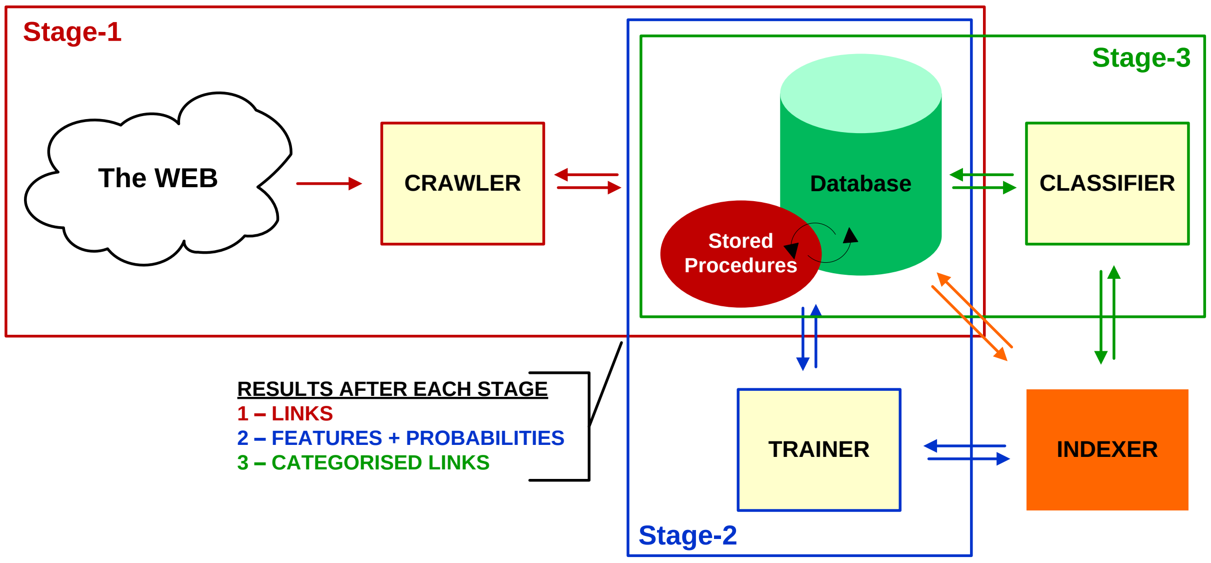


Figure 4.3: WIR System Stages

4.3.1 CRAWLER

The CRAWLER is seeded with the homepages of five training websites. The CRAWLER is responsible for going through the web pages provided and finding all other pages within each given page, without causing duplications. A list of important requirements for crawlers was included in section 2.2.1.

As shown in Figure 4.4, the CRAWLER starts by obtaining the content of the first seed URL. Exception handling statements are used to deal with links leading to dead ends. The content retrieved is a mixture of HTML code, relevant text, punctuation and irrelevant words. The HTML code can tolerate inconsistencies, e.g. missing HTML tags, however these irregularities affect CRAWLERS, which are based on the analysis of such HTML tags.

For this research, the HTML code is cleaned and tidied up before anything else occurs, e.g. missing HTML closing tags are added, links are repaired etc. The CRAWLER also discards potential *irrelevant* web pages. The latter was achieved after the development of the CLASSIFIER, as it involved the use of the NB approach to analyse page URLs and determine a list of URL keywords most likely to be associated with *irrelevant* web pages. The CRAWLER then rejects all URLs that contain any of these keywords. Table 4.2 shows the list of keywords identified through this approach. Note that, in order to avoid rejecting potentially *relevant* web pages, only keywords with an estimated probability of over 97% were considered.

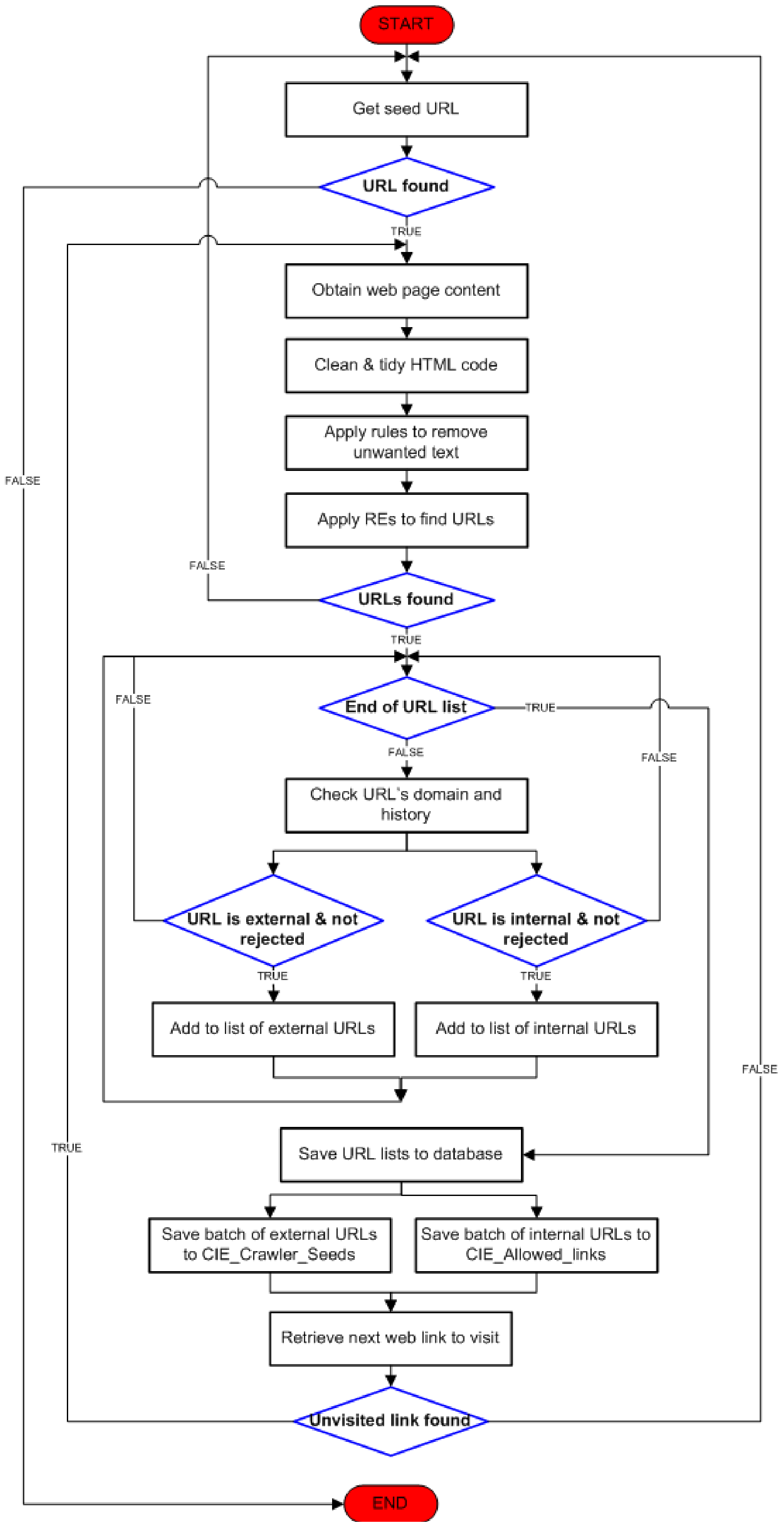

START
Get seed URL
URL found
TRUE
FALSE
Obtain web page content
Clean & tidy HTML code
Apply rules to remove unwanted text
Apply REs to find URLs
FALSE
URLs found
TRUE
End of URL list
TRUE
FALSE
FALSE
FALSE
Check URL's domain and history
URL is external & not rejected
URL is internal & not rejected
TRUE
TRUE
Add to list of external URLs
Add to list of internal URLs
FALSE
TRUE
Save URL lists to database
Save batch of external URLs to CIE_Crawler_Seeds
Save batch of internal URLs to CIE_Allowed_links
Retrieve next web link to visit
Unvisited link found
END

Figure 4.4: Crawling for New Links

Table 4.2: URL Keywords Identified as Pointing to Irrelevant Web Pages

about	anchor	article	basket
biography	book	brochure	career
claim	client	club	company
contact	disclaimer	download	faq
find	image	item	javascript
job	join	login	mailto
map	news	offer	opportunities
order	password	popup	press
privacy	register	registration	report
sample	shopping	sign	staticpage
subscribe	testimonial	ttg	username
vacancies	venue	workwithus	

Section 5.4.1 shows the number of web pages eliminated by utilising the keywords identified in Table 4.2 and discusses how this affects the results of the CLASSIFIER.

The web pages, which are not identified as *irrelevant* at this stage, are searched for further links. All links found are checked to see if they are direct children of the host web page (internal), or if they belong to external websites. Internal web pages are added to table CIE_Allowed_links for further processing, whereas external websites are added to the list of initial seeds, to be checked in a FIFO (First In First Out) manner. The links identified during the crawling process are checked against the list of links that have been rejected in previous crawling and/or classification sessions as well as the list of rejected domains stored in the database. The links that do not exist in either of these lists are repaired to include all parts of a URL (Figure 4.5). Ports (e.g.:80) and fragments (after #) are also considered, although not shown in the figure.

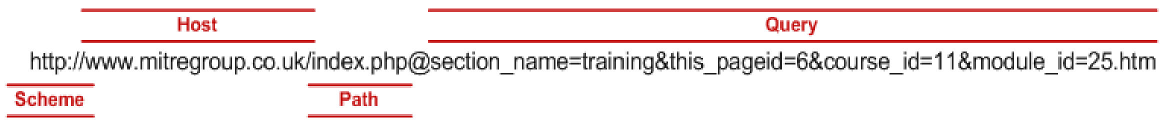

Host
Query
http://www.mitregroup.co.uk/index.php@section_name=training&this_pageid=6&course_id=11&module_id=25.htm
Scheme
Path

Figure 4.5: URL Parts

The batch of links found in each web page (both internal and external) is saved into the database, before the process is repeated for the next unvisited link. The SQL stored procedure responsible for selecting the next link to visit is very important, as it considers various parameters such as the search depth and refresh settings specified in the CIE_Settings table, as well as the file type of the next link to visit. In this prototype, files with extensions .PDF, .DOC and .XLS are not visited (see section 6.3).

The search algorithm used for crawling through each web page is the Breadth-First Search (BFS) algorithm (Figure 4.6). BFS is an uninformed search, which starts at the root node and systematically explores all the neighbouring nodes of each of the nearest nodes until the goal has been found, or in this case, when the search-depth specified in table CIE_Settings has been reached or the website has no more levels to be explored. The search depth is currently set to five levels, because no training websites were found that required more than five levels.

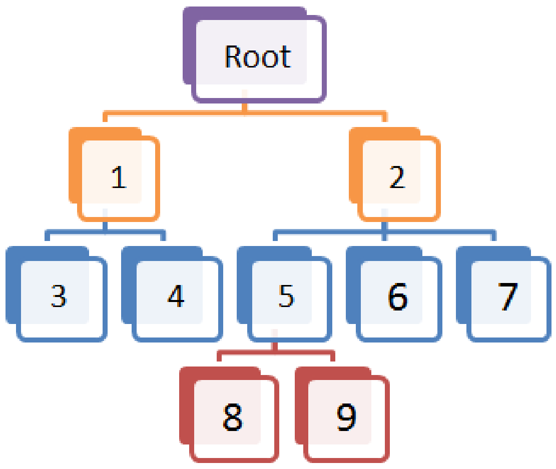


Figure 4.6: Breadth-First Search

4.3.2 TRAINER

Like all supervised methods, NB Classifiers are trained through examples. The TRAINER stage is responsible for providing such training to the system, using collections of web pages. The TRAINER uses both positive (*relevant*) and negative (*irrelevant*) examples (web pages) in order to learn which type of web pages contain relevant training course information and which do not. As mentioned in chapter 2, finding negative data samples for some domains is difficult; however this is not an issue for the training course domain.

Figure 4.7 shows the series of actions undertaken by the TRAINER.

The TRAINER uses the training URLs and their corresponding categories as inputs. Initially, each web page is indexed, which refers to the extraction of the most suitable features associated with each page (see section 4.3.3). All the features extracted from a web page are assigned to the same category as the web page itself (Table 4.5, page 63).

Table 4.3: Text and Categories Training Sample

Sample Text	Category
Our clients range from international 'blue chip' companies to small businesses.	Irrelevant
We have delivered work in the UK, mainland Europe and North America.	Irrelevant
1-day courses include: Management qualification available: ILM	Relevant
Level 3 Introductory Award in First Line Management in Organisations	Relevant
The company was established in 1992.	Irrelevant
Management qualification available: ILM Level 2 Introductory Award in Team Leading	Relevant

Table 4.3 shows a sample training set with sentences selected from six different web pages, three *relevant* and three *irrelevant* ones. The Classifier is trained by receiving examples one by one. During each training session, extracted features are saved in the database, together with the categories assigned to them. The number of features belonging to each and all categories and the number of times a specific feature is found in each category are updated after each training example. Features become strongly associated with the different categories through each training session.

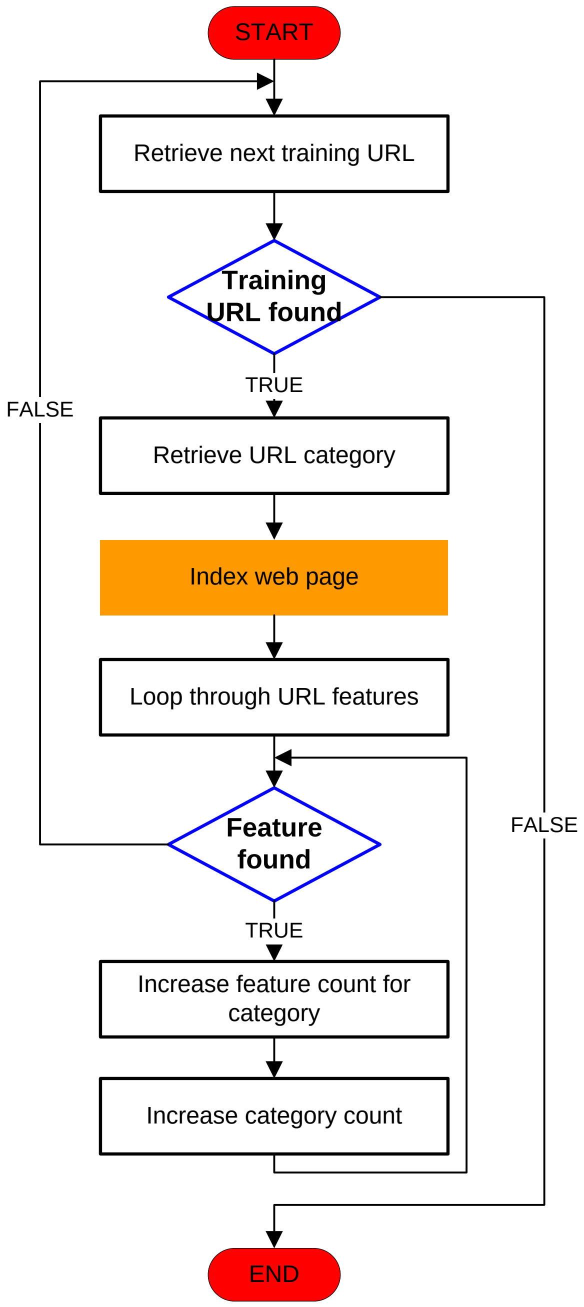


Figure 4.7: System Training

4.3.3 INDEXER

The INDEXER is responsible for extracting the one hundred most frequent text tokens from each web page, to be later used in the analysis process. The original text obtained from web pages contains HTML code such as tags, properties etc. These are not relevant to the CLASSIFIER, thus need to be removed prior to the classification process. Care is taken however, to preserve the information contained in certain web structures such as: the page title, links and META tags. This information is extracted at the beginning of the indexing process and kept safe until the "Remove stopwords" step (Figure 4.8).

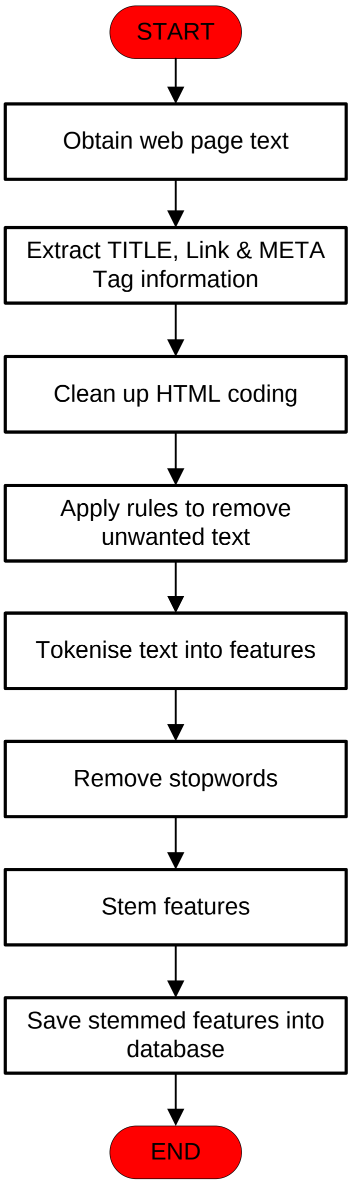

START
Obtain web page text
Extract TITLE, Link & META Tag information
Clean up HTML coding
Apply rules to remove unwanted text
Tokenise text into features
Remove stopwords
Stem features
Save stemmed features into database
END

Figure 4.8: Indexing a Web Page

As shown in Figure 4.8, the INDEXER strips off HTML tags together with their attributes and other properties. There is a PHP command to help with this process – strip_tags() – however this only gives partial results. This command not only ignores page scripts, styles, embedded objects and other unwanted code, but it also joins words found on either side of the tag being removed, which is incorrect. Figure 4.9 illustrates that, joins occurring when *inline tags* are removed (e.g. <strong>, <b>, <i> etc.) are correct, however joins occurring when *block-level tags* are removed (e.g. <p>, <div>, <h1> etc.) are incorrect.

This problem was resolved by forcing the INDEXER to add line breaks before and after block-level tags prior to the tag stripping process. The INDEXER was also designed to be able to remove invisible text such as scripts; remove punctuation, one letter words, HTML entities such as &nbsp;, &amp;, &quot; etc.

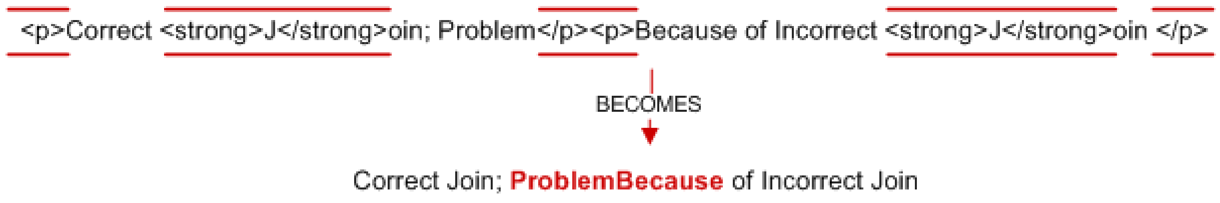

<p>Correct <strong>J</strong>oin; Problem</p><p>Because of Incorrect <strong>J</strong>oin </p>
BECOMES
Correct Join; ProblemBecause of Incorrect Join

Figure 4.9: PHP *strip_tags()* Joining Problem

Steps 3 and 4 (Figure 4.8) ensure that the page content is ‘clean’ enough to be taken through to the following step – tokenisation. Tokenisation separates the text into individual features, which are used by the TRAINER and the CLASSIFIER. Features that are believed to be too common or too insignificant in distinguishing web pages from one another are removed. Such features are known as Stopwords and the standard list available for computer science related projects can be found in Appendix C. An additional list of Stopwords was also created to apply to the training courses domain more closely (Table 4.4).

Table 4.4: Sample of Additional Stopwords

e.g.	group	home	I
ie	i.e.	right	left
mitre	ten	var	europe
asia	africa	america	north
south	east	west	blue
culture	accordia	horizon	king
unusual	intro	mitre	group

Another step undertaken by the INDEXER is the stemming of all the features resulting from the above steps. Stemming is the process of reducing words to their root form, otherwise known as the stem form. This form is the primary lexical unit of a word. However, the stems produced by the stemming process do not need to be identical to the real root of the word, as long as the related words are all stemmed to the same root form. This is because, the purpose of stemming is to avoid the analysis of related words as independent ones, as this can cause misinterpretation of the data. For example, words manage, managed, manages are all stemmed to *manag*; to the system this means that there are three instances of one word, rather than three words with one instance each.

The stemming algorithm used in this research is the Porter Stemming Algorithm (Porter, 1980). All the stemmed features are stored in table CIE_Words, ready to be used by the CLASSIFIER.

Using the same data sample as in Table 4.3, Table 4.5 shows the stemmed features extracted from the text provided, after the stopwords have been removed. Each stemmed feature is assigned to the same category as the web page in which it is found.

Table 4.5: Stemmed Features and Categories Training Sample

Web Page Text	Features Extracted	Category
Our clients range from international 'blue chip' companies to small businesses.	client, range, international, chip, compani, small, busi	Irrelevant
We have delivered work in the UK, mainland Europe and North America.	deliver, work, uk, mainland	Irrelevant
1-day courses include: Management qualification available: ILM	cours, includ, manag, qualif, avail, ilm	Relevant
Level 3 Introductory Award in First Line Management in Organisations	level, introduct, award, line, manag, organisatio	Relevant
The company was established in 1992.	compani, establish	Irrelevant
Management qualification: ILM Level 2 Introductory Award in Team Leading	manag, qualif, ilm, level, introductori, award, team, lead	Relevant

4.3.4 CLASSIFIER

The CLASSIFIER is responsible for classifying previously unseen web pages into appropriate categories. In its very first instance, the CLASSIFIER relies heavily on the ‘knowledge’ obtained from the training process. Inadequate training could result in questionable results at this stage. The CLASSIFIER then goes through different steps to analyse the information available (Figure 4.10) and classify new web pages. Its ‘knowledge’ is also updated after each decision made, thus the CLASSIFIER becomes more accurate, the more classifications it performs.

As previously mentioned, the CLASSIFIER needs to make use of the INDEXER to obtain all the appropriate keywords/features from each web page analysed. Once the keywords are extracted, the classification algorithm is applied to each category stored in CIE_Categories. There are only two categories used in this research – *relevant* and *irrelevant* – however the system is designed to work with any number of categories. This is to allow for potential future changes at ATM. The standard Bayes Theorem is defined as follows:

(Eq. 1)

$$\arg\max_{n}\{P(C_n|W)\} = \frac{P(W|C_n) \times P(C_n)}{P(W)}$$

where:

$P(C_n|W)$ = the posterior probability that, given a web page W, the page belongs to category C_n
$P(W|C_n)$ = the probability that, given a category C_n, web page W appears in this category
$P(C_n)$ = the prior probability of the given category C_n, i.e. the probability of a web page being in category C_n given no evidence.
$P(W)$ = the probability of this particular web page occurring

The $P(W)$ can be disregarded, because it has the same value regardless of the category for which the calculation is carried out, and as such it will scale the end probabilities by the exact same amount, thus making no difference to the overall calculation. Also, the results of this calculation are going to be used in comparison with each other, rather than as stand-alone probabilities, thus calculating $P(W)$ would be unnecessary effort. Bayes Theorem is therefore simplified to:

(Eq. 2)

$$\arg\max_{n}\{P(C_n|W)\} \propto P(W|C_n) \times P(C_n)$$

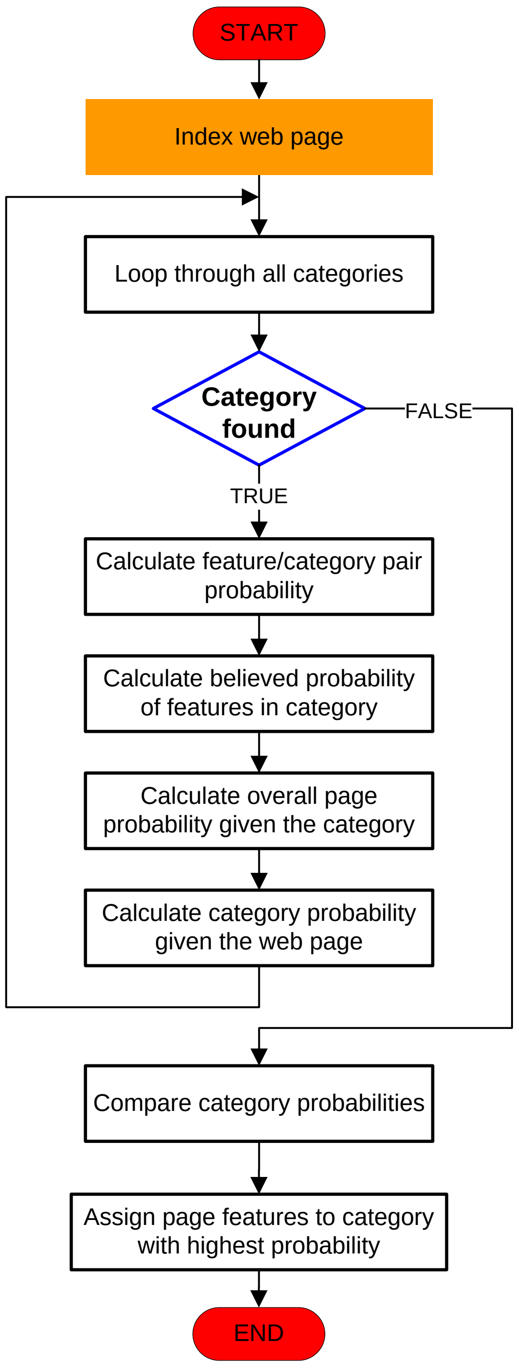

START
Index web page
Loop through all categories
Category found
FALSE
TRUE
Calculate feature/category pair probability
Calculate believed probability of features in category
Calculate overall page probability given the category
Calculate category probability given the web page
Compare category probabilities
Assign page features to category with highest probability
END

Figure 4.10: Classifying New Web Page

Each web page is composed of a set of features $\{f_1, f_2 \ldots f_i, \ldots, f_n\}$ (extracted by the INDEXER as explained in section 4.3.3), thus calculating *P(W|C$_i$)* involves calculating and combining the probabilities of each individual feature. The NB theorem assumes that all the different features in a web page can be considered independently of one another, thus the probability of a page belonging to a particular category is calculated through the simple product of the individual probabilities for each feature (Eq. 3).

(Eq. 3)

$$\arg\max_{n}\{P(C_n|W)\} \propto \frac{1}{z}\,P(C_n)\;\times\prod_{i=1}^{n} P(f_i|C_n)$$

where z is a scaling factor dependent on the feature variables, which helps to normalise the data. This scaling factor was added, because it was found that, when experimenting with smaller datasets, the feature probabilities got very small, which made the page probabilities get close to zero. This made the system unusable.

NB classifiers are very efficient in managing attribute data (Zhang, 2004), however they are not as capable with attributes that have rarely been investigated before. This is particularly the case at the early stages of the classification process, when there is little information about attributes, if any at all. For example, considering that there is only one instance of the feature ‘leadership’ in the classifier’s knowledge base and this instance belongs to category *irrelevant*, then the probability of this feature being assigned to category *relevant* would be calculated as zero. This is an extreme calculation however, because ‘leadership’ may be a perfectly relevant word to characterise a training course web page, but just so happened to be found in an irrelevant web page first.

This research deals with the above limitation by adding an additional step to the traditional NB approach. This additional step involves the calculation of a Believed Probability for all features, which helps to calculate more gradual probability values for the data.

An initial probability is decided for each feature, which, in this research, is equal to the probability of the page given no evidence (*P(C_n)*). The Believed Probability represents the weighted average of this initial probability and the probability of the feature in the given category calculated in the ordinary manner. The calculation is as follows:

(Eq. 4)

$$Believed\ Probability = \frac{\big(bpw \times P(C_n)\big) + \big(\sum_{i=1}^{C_n} nf_i \times P(f_i|C_n)\big)}{bpw + \sum_{i=1}^{C_n} nf_i}$$

where: nf_i is the count of feature f_i in all categories and bpw is the Believed Probability Weight. In this research bpw = 1, which means that the Believed Probability is weighted the same as one feature.

Once the probabilities for each category have been calculated, the probability values are compared to each other. Web page W is classified into category C_i if and only if:

(Eq. 5)

$$P(C_i|W) > P\big(C_j|W\big)\ \ for\ all\ \ j \neq i$$

The top one hundred, most frequent features, extracted from the classified page, are associated with the resulting category and the corresponding information in the database is updated, thus expanding the system’s knowledge. Experiments carried out on this part of the system and the achieved results are discussed in section 5.4.2.

4.4 SYSTEM FUNCTIONALITY – WIE

The WIE part of the system deals with the extraction of specific training course information such as course title, price(s), location(s) and date(s), from the previously retrieved web pages. Figure 4.1 showed that the EXTRACTOR uses the list of relevant links as well as an initial set of genotype seeds to evolve extraction solutions, specifically REs. The combination of GP principles and REs was chosen for this research, because although each field separately is popular and well researched, few researchers have focused their attention on the evolution of REs for the purpose of WIE (section 2.4.6).

The following sections discuss GP and REs and the steps involved in the development of this part of the system.

4.4.1 REGULAR EXPRESSIONS (REs)

REs are powerful tools used to detect patterns in data (Thompson, 1968; Friedl, 1997; Sells, 2002; Friedl, 2006; Sun Microsystems, 2008). They can range from basic to very complex, matching from just literal text[6] to very specific instances of text based on certain criteria. For example: *^[A-Z][a-z]+* matches all instances that begin (^) with an uppercase letter (*[A-Z]*), followed by one or more (+) lowercase letters (*[a-z]*), such as: "Regular" or "Expression" but not: "regular", "RE" or "REs".

REs are very well known, particularly in the UNIX community and they feature largely in some programming languages such as Perl, PHP and AWK. However, the manual generation of REs can be a difficult, error-prone and time consuming undertaking, especially for complex patterns. This is due to the fact that although REs are built up from small building blocks, where each block is fairly simple, all the available blocks can be combined in an infinite number of ways (Friedl, 2006), which may result in a highly complex RE.

Tools have been developed to evaluate the validity of REs (RegexDesigner.NET – Sells, 2002; RegexBuddy – Goyvaerts, 2009 etc.), however, very little, if anything, has been done towards the automatic generation of REs.

GP principles allow for the automation of RE generation, whilst aiming to evolve the optimal RE for each particular web page under investigation. The following section details the steps involved in the genetic evolution of REs in this research.

4.4.2 EXTRACTOR

Section 2.4.6 discussed past and present research work in the field of GP, specifically in relation to WIE and the evolution of REs. This section gives details on the steps involved in developing a genetic system for the extraction of training course information through evolved REs. Much of this information has been published (Xhemali et al., 2010-a; Xhemali et al., 2010-b; Xhemali et al., 2010-c), thus references to the specific papers may be given instead of repeating information. The GP steps involved in this research follow a straightforward algorithm as shown below:

[6] / all this is literal text and will be captured by the RE /

```
Pseudo-code: GP Algorithm                                                    (1)
  P = Initial population of REs (with corresponding genotypes)
  F = Evaluate Fitness of P
  Loop through to the (predefined) maximum number of generations.
      Loop through P until offspring population is the same size as the parent population.
          Select two parents from P for reproduction.
          O = Produce two offspring through Uniform Crossover.
          M = Mutate one gene from each offspring.
      End Loop
      PH = Map offspring genotypes to the corresponding phenotypes (REs).
      F = Apply Fitness Test to PH.
      If perfect solution is found
          Stop and exit the evolutionary system
      Else
          Replace the weakest individual in the parent population with offspring, in order to
          preserve the best parents and offspring throughout the evolutionary process.
        Repeat Loop for new population P.
  End Loop
```

The following subsections explain each of the above steps in comparison to other available approaches, in order to justify the decisions made in this research.

4.4.2.1 Representation

There are two main representations in GP: the Tree-Based GP and the Linear GP (Poli et al., 2008; Withall et al., 2008). The Tree-Based GP represents programs as syntax trees (Figure 4.11), where program variables and constants make up the tree leaves (2.2, X, 11, 7, Y), whereas the program operators (+, –, /, *, cos) are the internal nodes. The tree leaves are also known as the terminals, whereas the internal nodes are known as the functions. Figure 4.11 shows the tree representation of program: $(2.2 - (X/11)) + (7 * \cos(Y))$. For more complex programs, the main tree can contain many sub trees.

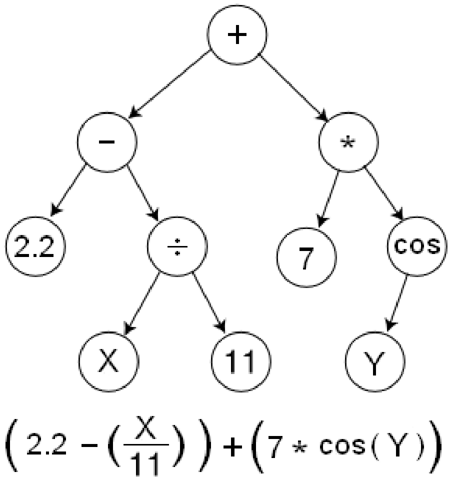

+
−
*
2.2
÷
7
cos
X
11
Y
$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$

Figure 4.11: Tree-based GP Representation

Linear GP represents programs as a linear sequence of instructions (Figure 4.12). Based on biological evolution, the sets of instructions are known as Genotypes, each set of instructions is called a chromosome and each individual instruction is a gene. Linear GP representations may have either fixed length Genotypes i.e. the same number of genes for every instruction, or variable ones. This depends on the problem to be solved.

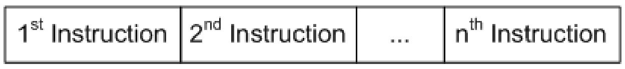


Figure 4.12: Linear GP Representation

The GP representation can have a great impact on the performance of the genetic system. Generally, the larger the search space allowed by the representation, the longer it will take to achieve a solution (Koza, 1992).

The main difference between the Tree-Based and Linear representations relates to the way individuals are manipulated in the search space and solution space. In Tree-Based GP (Koza, 1992; Whigham, 1995; Conrad, 2007; Snajder et al., 2008 etc.) individuals remain the same throughout the process, thus the search space and the solution space are identical. Reproduction is achieved by swapping different branches of the trees of parent individuals and mutation is achieved by adding/updating or deleting branches from offspring trees. Linear GP, on the other hand, separates the search space from the solution space. This separation involves the encoding of the individuals to a form known as the genotype, which is later on decoded back to the corresponding individual referred to as the phenotype. Banzhaf (1994) was the first one to suggest this separation. Since then, many other researchers have embraced the separation into genotypes and phenotypes (Keller & Banzhaf, 1996; Withall et al., 2008; Clarke et al., 2009 etc.). This separation simplifies and increases the efficiency of certain genetic operations such as: Crossover and Mutation, because these would no longer be constrained by the parameters used in the program being evolved. In genotype-phenotype based GP, operations such as Crossover and Mutation would be performed on the genotype, whereas other processes, such as the Fitness scoring, would be performed on the phenotype.

On the downside, however, this adds an additional step to the genetic evolution process – the translation or mapping of the genotypes to their corresponding phenotypes. This step occurs after the genetic reproduction stage (i.e. the crossover and mutation) and before the Fitness test can take place. There are researchers who criticise the separation into genotypes and phenotypes (Moore, 2000). The main concern expressed is that the conversion process of a mutated genotype into the phenotype may result in anomalies that could potentially lead to invalid solutions. A direct mapping between the encoded program (genotype) and the actual program (phenotype) is therefore vital to ensure the validity of the solutions (Rothlauf, 2006; Withall et al., 2008).

The linear approach was chosen for this research, because it is more appropriate for the representation of non-standard models like REs. Section 4.4.2.4 explains this concept further.

4.4.2.2 Parent Selection

Selecting fit parents for reproduction will increase the chances of the offspring also having good qualities. There are three main selection techniques: Ranking, Roulette Wheel and Tournament Selection.

Ranking: This method picks parents at random from the list of available genotypes. With this technique, all genotypes have an equal chance of being selected, regardless of their qualities.

Roulette Wheel: This method gives priority to the genotypes with high fitness scores. It is based on the idea behind a roulette wheel, where the slots are of different sizes and thus affect where the roulette stops. In parent selection, the slots are assigned proportionally to the genotypes' fitness scores. E.g. the genotype with the highest score would get the biggest slot and the one with the lowest score would get the smallest. This technique is popular, however it has a tendency to achieve premature convergence, which can cause local optima, or in other words little genetic diversity

Tournament Selection: This is the parent selection method chosen for this research. Tournament Selection is successful in giving all genotypes a chance to be selected, slightly leaning towards the ones with better fitness scores. It is also immune from any constant offset in the fitness evaluation. All this avoids premature convergence, whilst still selecting good quality parents. The process is as follows:

```
Pseudo-code: Tournament Selection Process                                    (2)
  ▪ TS = Tournament-Size (40% of the population size)
  ▪ Randomly select TS individuals from population
  ▪ Select the genotype with the highest fitness score from the TS individuals chosen
    previously.
  ▪ Repeat steps 2 and 3 once more to choose the second parent genotype.
```

4.4.2.3 Genetic Operators

Reproduction is significant in creating new generations of solutions, which would ideally inherit the useful characteristics from the parent population and discard the ineffective ones. In this research, the offspring created may be of variable length, depending on the lengths of their parent genotypes. Reproduction is aided by two main genetic operators: Crossover and Mutation.

Crossover: Crossover involves combing genes from two or more parents (normally two – Ward, 1999) to create one or more offspring. Depending on the number of crossover points selected in both parent genotypes, the different types of crossover operators include: Single-point Crossover, Two-point Crossover, Multiple-point Crossover or Uniform Crossover. More details on Crossover techniques can be found in the work of Syswerda (1989), who maintains that Uniform Crossover is the best choice.

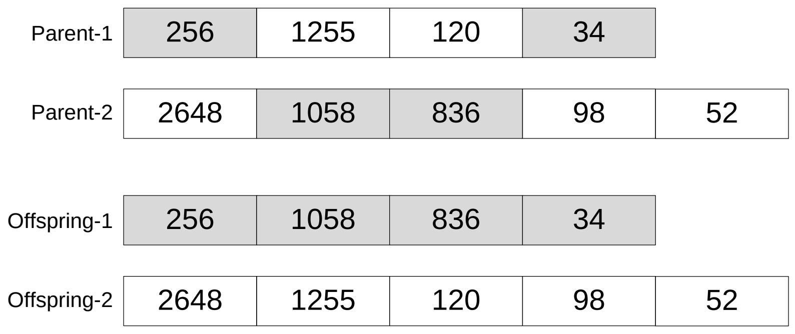


Figure 4.13: Uniform Crossover

This research has chosen the Uniform Crossover, which involves all the genes in both parent genotypes getting swapped based on 50% probability. Two offspring are generated each time two parents reproduce. The second offspring is the exact opposite of the first one, created with the ‘left over’ genes (Figure 4.13).

Mutation: Mutation usually follows the Crossover process, although one may choose to perform mutation prior to crossover. The goal of mutation is to help to break out of local optima. There is no guarantee, however, that mutation will create a better genotype, due to mutation occurring at random.

In this research, each offspring is mutated after the Crossover process. Mutation is rare however, with only one gene in each offspring genotype being manipulated. The offspring are mutated instead of the parents, because this way, the parent population does not need to be reset, which simplifies the process.

4.4.2.4 Genotype-Phenotype Mapping

This section continues from section 4.4.2.1. This part is a significant component of the overall evolutionary system, thus, some past and present research works are discussed in comparison to the novel genotype-phenotype mapping idea presented in this thesis. This idea is then explained in detail through examples.

Banzhaf (1994; 2006) represented his genotypes as linear binary strings. The mapping stage then processed these genotypes from left to right in 5-bit sections, where each 5-bit code mapped to a pre-specified symbol. For example: 00000 mapped to PLUS (+), 00100 mapped to POW, 11000 mapped to variable X etc. The first bit indicated whether the code represented a function (PLUS, POW etc.) or a terminal (X, Y etc.). The research also discussed their concern about generating constant numbers. Koza (1992) had solved this problem by defining “random ephemeral constants” where constants are only generated once for a particular program and then reused wherever they are needed within that program.

Keller (1996) continued in the footsteps of Banzhaf, concentrating on providing experimental evidence for choosing the genotype-phenotype approach instead of the normal GP approach. Keller used the LALR (Look Ahead LR) parser for the repairing stage of the genotype-phenotype mapping process. LALR parsers scan the input from left to right and construct a rightmost derivation in reverse (Aho and Ullman, 1979).

There was a certain amount of redundancy in the genetic coding in both Banzhaf’s and Keller’s works. They both admitted that, in their works, different binary strings could correspond to the same symbol, which could lead to inconsistencies e.g. 000 and 100 both mapping to ‘a’.

A slightly different genotype representation is seen in the work of Withall et al. (2008). In here genotypes are represented as linear blocks of integers. Each block consists of exactly four integers, each integer representing a different gene. Although both research works used fixed-length genotypes, in the work of Banzhaf (1994; 2006) the resulting phenotypes could vary in length, whereas in (Withall et al., 2008) they remained fixed. However, Withall allowed for variable-length genotypes through padding, whereby shorter program structures or statements ignored outstanding genes. The first integer in Withall’s genotype always determines the type of function that follows.

Grosan and Abraham (2009) worked with multi-chromosome genotypes. The number of chromosomes was varied. However, the number of genes per chromosome was fixed. In this research, each gene corresponded to either a terminal: T = {a, b, c, d} or a function: F = {+, *}. A function gene also included pointers towards the function parameters to tell the system which terminals were to be manipulated by the function. Also, the first gene of the chromosome was always a terminal. This was to ensure that only syntactically correct programs were evolved. Very differently from above, Yosif et al. (2010) introduced the novel approach of adapting a support vector machine to predict phenotypes from genotype data.

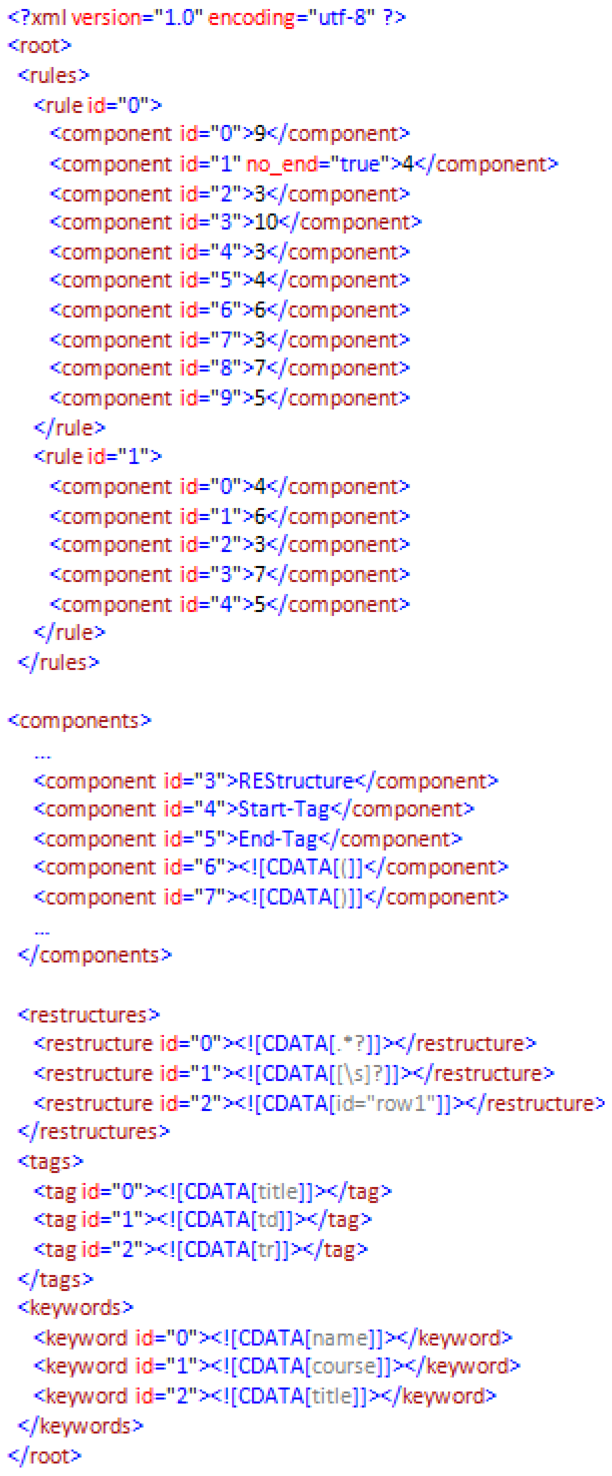
Similarly to Withall et al. (2008), the genotype in the research presented in this thesis is represented as a string of integers. There are no fixed length genotypes determined however; instead the genotype can contain any number of genes. This is because different Regular Expressions (REs) may contain a varying number of components (i.e. tags, RE structures, keywords etc.). One can create regular expressions with as few or as many components as one wishes. Withall et al., however, dealt with the evolution of programming statements and structures (discussed in detail in Xhemali et al., 2010-c), which are more rigid when it comes to the number of components they require.

The genotype-phenotype mapping in this research is achieved through an innovative approach involving carefully written XML rules being fed to the system as a separate file. This technique has several advantages. The existence of rules as a separate file means that if this file is replaced with one that works on a different domain, then the rest of the GP system will be able to carry on working for the new domain with minimal interruptions. Specifically, the two areas that would need to be changed to work with the new domain are the Fitness Test and the Repairing function discussed below. Furthermore, the use of XML denotes improved readability, compatibility with many programming languages, portability and extendibility (XML is not restricted to a limited set of keywords defined by the proprietary vendors, which aids the process of creating rules of different levels of complexity).

The initial XML rules were created manually after an extensive analysis of a number of web pages. However, in the future, new rules will be able to be added to the XML file automatically (Siau, Hinde and Stone (Siau et al., 2010) are currently working towards implementing this functionality). The following explains the mapping of genotypes to phenotypes in detail.

Note that this is not a character by character evolution, because this would increase the search space and dramatically increase the execution time. Instead REs are divided into three collections: HTML tags (e.g. “title”, “tr” etc.), keywords (e.g. “course”, “title” etc.) and RE substructures (e.g. “.*?” or “[\s]?” etc.). Each evolved gene is translated to an element of one of these collections. Each collection is further divided into a number of components e.g. the Start-Tag component determines an opening tag, the End-Tag determines a closing tag, the Start-REStructure determines an opening RE structure, an RE-Structure determines a normal structure that does not need closing etc. These components are important for ensuring consistency and accuracy in the phenotypes created. For example once a Start-Tag has been determined, the system automatically knows that it needs to reuse this tag as an End-Tag when told so by the rules.

There are also some additional components that refer to elements that do not need to be evolved and as such do not require the use of any extra genes from the genotype. For example, the Start-Capture component translates to symbol ‘(‘ and indicates the beginning of a capturing group i.e. the part of the RE, which captures the part of the results needed; the End-Capture component indicates the end of the capturing group etc. These additional components were introduced to help the mapping process and the Repairing function to translate genotypes into syntactically correct REs.



```
<?xml version="1.0" encoding="utf-8" ?>
<root>
 <rules>
   <rule id="0">
     <component id="0">9</component>
     <component id="1" no_end="true">4</component>
     <component id="2">3</component>
     <component id="3">10</component>
     <component id="4">3</component>
     <component id="5">4</component>
     <component id="6">6</component>
     <component id="7">3</component>
     <component id="8">7</component>
     <component id="9">5</component>
   </rule>
   <rule id="1">
     <component id="0">4</component>
     <component id="1">6</component>
     <component id="2">3</component>
     <component id="3">7</component>
     <component id="4">5</component>
   </rule>
 </rules>

<components>
   ...
   <component id="3">REStructure</component>
   <component id="4">Start-Tag</component>
   <component id="5">End-Tag</component>
   <component id="6"><![CDATA[(]]</component>
   <component id="7"><![CDATA[)]]</component>
   ...
 </components>

 <restructures>
   <restructure id="0"><![CDATA[.*?]]></restructure>
   <restructure id="1"><![CDATA[[\s]?]]></restructure>
   <restructure id="2"><![CDATA[id="row1"]]></restructure>
 </restructures>
 <tags>
   <tag id="0"><![CDATA[title]]></tag>
   <tag id="1"><![CDATA[td]]></tag>
   <tag id="2"><![CDATA[tr]]></tag>
 </tags>
 <keywords>
   <keyword id="0"><![CDATA[name]]></keyword>
   <keyword id="1"><![CDATA[course]]></keyword>
   <keyword id="2"><![CDATA[title]]></keyword>
 </keywords>
</root>
```

Figure 4.14: Sample of XML Rules

The Repairing function is responsible for making sure that all the different components are combined correctly to create a syntactically correct RE. Figure 4.14 shows a sample of the XML rules and components needed to guide the genotype-phenotype stage of the genetic evolution of REs for the extraction of course titles. The components are shown in the order in which they are to be chosen by the mapping stage in order to create a valid and efficient RE. The following gives an example to illustrate this method. Table 4.6 shows a sample genotype to be translated using the information in Figure 4.14. Note that the first gene in the genotype is always associated with the RE rule choice. The remaining string of integers in the genotype maps to the different components within that rule.

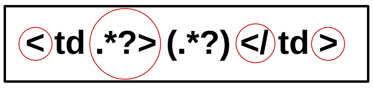
The modulo function is used for this purpose. In this example (Table 4.6), the value of the first gene is 5. This represents the XML rule to be followed. In this case, there are two different rules in the XML file, so 5 mod 2 = 1 means that the second rule (id = 1) is chosen. This rule contains five different components (Figure 4.14). The first component (no. 4) corresponds to the Start-Tag component, which means the next gene in the genotype (gene 7) needs to be mapped to one of the elements in the 'tags' collection. In this case, the collection has three elements, thus 7 mod 3 = 1 gives the 'td' tag. The following component (no. 6) corresponds to the Start-Capture component, which maps to symbol "(". This is a static component, which does not need to be evolved and as such does not require the use of any genes from the genotype. The remaining components are dealt with in the same manner (see Table 4.7). Note that in this example, only the first three genes of the genotype were needed; the remaining two are simply ignored.

Table 4.6: Genotype

5	7	15	22	43

Table 4.7: Genotype to Phenotype (RE) Mapping

Component No.	*Component*	*Gene Used*	*Modulo*	*Translation*
-	-	5	1	Rule 2
4	Start-Tag	7	1	td
6	Start-Capture	-	-	(
3	RE-Structure	15	0	.*?
7	End-Capture	-	-	)
5	End-Tag	-	-	td



```
<td.*?>(.*?)</td>
```

Figure 4.15: Phenotype

Once all the required genes have been decoded, the resulting RE is repaired to ensure its validity and efficiency. Figure 4.15 shows the complete RE (phenotype) for the above example. The additional symbols added by the Repairing function are shown circled.

The Repairing function is an independent function, which scrutinises the phenotype created in order to guarantee the syntactic validity of the RE. This function is in charge of tasks like: closing opening tags based on the opening tag evolved; preserving tag nesting in the RE hierarchy[7]; closing opening parentheses; adding/removing RE structures in cases when there are fewer genes in the genotype than required by the XML rule; adding variable declarations at the beginning of the program; adding 'footer' information where necessary e.g. when returning the RE to a calling function, etc. All this is achieved through the use of a STACK programming structure, which works in a LIFO (Last In First Out) manner.

The XML rules can also determine whether or not a closing tag is actually needed. This is useful in cases where the inclusion of a closing tag results in different results being extracted to the ones needed. For example, the RE: <tr[\s]?id="row1".*?><td.*?>.*?</td> contains two opening tags ('tr' and 'td'), however only the 'td' tag needs to be closed for this to work as intended. This is achieved by including no_end="true" in the rule (Figure 4.14).

Xhemali et al. (2010-a) and Xhemali et al. (2010-c) concentrated on applying the above technique to different domains. Inspired by the work of Withall et al. (2008), the domains chosen were the Software Statements and Structures and Complete Software Programs. The changes that were required for the above approach to work for such different domains are explained in detail in the above papers. Experimental results from the complete evolution of software programs, particularly the evolution of 'Sorting' programs can be found in Xhemali et al. (2010-c).

4.4.2.5 Fitness Test

The fitness test is a very important part of GP. It is responsible for validating the efficiency of each individual from a population at solving the given problem. Normally, a higher score means better fitness. These fitness scores are used to aid the selection of parents before reproduction can take place (as explained in section 4.4.2.2). The highest score can also terminate the evolutionary system, as it means that the perfect individual has been generated.

In this research the fitness test is concerned with the validation of Regular Expressions (REs), which attempt to extract the course name/title, price(s), location(s) and date(s) from web pages. Initially, it was believed that all the different attributes of the course, i.e. the title, price(s), location(s) and date(s), could be investigated independently of one another, whilst still retaining the consistency and accuracy of the information. The initial idea was to therefore separate the GP system into four separate subsystems each dealing with a separate course attribute. For example, one of the subsystems would be responsible for the evolution of REs to extract the course location from the specified web page, another one would be responsible for the evolution of REs to extract the course price from that same web page and so on. All the subsystems would work in exactly the same manner. The only differences would be the XML rules and the fitness test, as these are too closely related to each RE type generated.

This was an exciting idea for this research, because it simplified the whole process. The rules could be specific enough to concentrate on one area alone and the fitness test would only have to be concerned with the evaluation of one type of RE.

However, a more thorough research and investigation into training course websites showed that the above idea can only be applied to situations where there is never any dependency

[7] E.g. Tag1 ... Tag2 ... REStructure ... *Closing-REStructure* ... *Closing-Tag2* ... *Closing-Tag1* (The *Italic* part shows the part of the RE added by the Repairing function)

amongst attributes to be extracted. For example, if every web page advertised single courses that only had one price, one location and one date, or multiple values for only one of the attributes, then the above solution is ideal. However, this is not always the case. Some websites advertise courses in multiple locations and multiple dates and some even have different prices associated with the different dates and locations. The course title, however, remains independent from the rest of the course attributes, thus, adhering to the above idea, a separate subsystem was designed to evolve REs for extracting course titles. Course details such as price(s), location(s) and date(s) are at times dependent on one-another, thus needed to be managed as a group.

In relation to the fitness test, course titles are pure text thus the general formatting is not helpful. Instead, it was found that the analysis of the content itself provided better answers (section 2.3.6.1). For this reason, the NB approach, developed for the WIR part of the system, was reused to predict the validity of the evolved title REs. Similarly to before, the NB system is trained with positive and negative examples of training course titles, which help to create a knowledge base for future decisions.

The NB solution was unsuitable for the fitness evaluation of the remaining attributes, i.e. date(s), price(s) and location(s), due to the partial dependency in this data. The main approach followed for the validation of these attributes was based on utilising the training course information already stored in the database. Specifically, table CIE_Courses stores all the course details extracted from each web page. Comparing the RE output against the data in this table not only validates the output from the evolved REs, but also guarantees that the data stored in this table is always kept up-to-date, which was part of the main aim for this research.

Evaluating information that does not exist in the database is more difficult. This is because, the RE domain is too flexible and there are many unknowns. For example, the system may evolve REs that extract entirely different information (not necessarily related to the course). Also, REs may extract incorrect information that is very similar to the real information that needs to be extracted, e.g. some web pages advertise books to accompany each course, which means that the pages include book prices as well as course prices. These would be challenging for a program to distinguish. Furthermore, the evolution process may generate REs that extract information regarding prices, dates and locations separately. The difficulty in this case is that the system needs to determine how to correlate this data accurately i.e. associate the correct price with the correct date with the correct location. This is extremely challenging.

In order to manage the above scenarios and others that may arise, REs are evaluated against different criteria. Each criterion adds points to the overall score for each RE evolved. The main criteria evaluated include:

- **Formatting**: Formatting is used to check if the attributes extracted are indeed what they are supposed to be, e.g. if the extracted attribute is a valid date (dd/mm/yy or dd/mm/yyyy etc.), or a valid price (£decimal.number) etc.

- **GP_Locations:** This database table includes all the towns, cities and counties in the UK. The locations extracted by the evolved REs are checked against this list to ensure that a valid location is extracted.

- **Length of extracted attributes:** Prices, dates and locations tend not to expand over a large number of characters, thus, large pieces of extractions are scored lower.

- **Single vs. Set of Attributes**: The extracted data is analysed to determine whether the information represents a separate attribute or a collection of different attributes. This step also checks if the attribute(s) extracted contain(s) single or multiple values.

- **Page position:** This is important when an attribute is extracted separately from the other attributes. In this case, once the attribute is verified as valid, its position in the web page is identified. The fitness test then attempts to locate the remaining attributes within a specified, close proximity from the position of the extracted attribute in the web page (e.g. ± 20 characters).

- **RE length**: Shorter REs are favoured to longer, more complex REs.

- **RE duplications**: REs are checked for any duplications or redundancies, e.g. RE component ".*?.*.*?" achieves the same result as ".*?", thus it is scored lower.

Once an evolved RE has been evaluated as a fitter candidate by the fitness test, the parent replacement takes place. The process of selecting the better individuals is known as Elitism.

In this research, only the weakest individuals from the parent population are substituted with the fitter offspring. If the fitness test proves that the offspring is weaker than any of the parents, then the offspring is rejected and the parents are taken forward to the next generation.

4.4.2.6 Initial Population

The initial population can either be created randomly or be seeded with known solutions. The random approach is the most common approach. This is because generating individuals randomly for the initial population can provide variation amongst individuals, which is an advantage. This approach is straightforward for Linear GP, which involves the simple generation of random integers, however Tree-based GP must ensure that only syntactically correct programs are generated.

Seeding the initial population with known solutions means that the evolutionary process can improve them and arrive at an optimal solution quickly. However, this approach can also limit the areas of the search space that the evolutionary system has initial access to (Withall et al., 2008), which may make it less likely for the system to find solutions in other areas.

In this research, a novel combination of the two approaches is used. As discussed in section 4.4.2.4, the first gene in the genotype represents the type of rule to be followed. Therefore, the first gene is seeded from existing solutions in that category (e.g. if the system needs to evolve REs to extract course titles, then the first genes of the top ten RE genotypes, that were successfully evolved to extract titles, are used as the first genes of the initial population). The remaining genes of each genotype are generated randomly. This approach allows for some initial knowledge to be injected in the evolutionary process, without compromising the search space too much.

4.4.2.7 Termination

The conditions that terminate the GP process are:

- The maximum number of predefined generations, over which the evolutionary system is run, is reached. The maximum number of generations specified for this research is 50,000.
- A fit solution for the problem is found, i.e. an RE has been evolved that successfully extracts training course information from the web page under investigation.
- The solution has not improved for several generations (100 generations in this research).

4.5 SUMMARY

This chapter discussed the research and development undertaken to meet the aims and objectives specified in Chapter 1. A thorough description of the database structure and the different components of the system – CRAWLER, TRAINER, INDEXER, CLASSIFIER and EXTRACTOR – were presented in this chapter. Details regarding the programming languages used and other development tools were given in section 3.4.5. The key findings of the research, as well as a detailed evaluation of various aspects of the research are presented in chapter 5.

5 FINDINGS AND EVALUATION

5.1 INTRODUCTION

This chapter presents the key findings of the research. Several tests have been set up to validate each part of the system from different angles. The final results achieved are presented and compared against existing similar technologies. Furthermore, the results are evaluated in relation to the aims and objectives set out at the beginning of the project, the approaches used for this research and the data corpus analysed.

The various tests set up in this research and discussed in this chapter include:

Test 1:	Compare CRAWLER against two existing crawling technologies.
Test 2:	Evaluate CRAWLER with and without the functionality to reject web pages deemed *irrelevant* (discussed in section 4.3.1).
Test 3:	Compare CRAWLER against Google.
Test 4:	Test for feature independence to evaluate the NB approach.
Test 5:	Evaluate CLASSIFIER against data corpuses of different volumes.
Test 6:	Compare CLASSIFIER against Neural Network and Decision Tree classifiers.
Test 7:	Compare the improved CLASSIFIER against standard NB classifiers.
Test 8:	Compare PHP CLASSIFIER against VB.NET CLASSIFIER
Test 9:	Evaluate the GP parameters established for the EXTRACTOR.
Test 10:	Evaluate EXTRACTOR for the extraction of course titles.
Test 11:	Evaluate the benefit of passing the knowledge learnt from generation to generation in the genetic evolution of Regular Expressions (REs) for the extraction of course titles.
Test 12:	Evaluate EXTRACTOR for the extraction of course dates, prices and locations.
Test 13:	Compare the Genotype-Phenotype Mapping for REs against the Genotype-Phenotype Mapping for 'Software Statements and Structures'.
Test 14:	Evaluate EXTRACTOR against the entirely different domain of 'Complete Software Programs'.
Test 15:	Evaluate each part of the system against two different operating systems: Windows XP Pro and Windows 7 (32-bit).

5.2 PERFORMANCE MEASURES

The approach chosen to evaluate the experiments related to the CLASSIFIER utilises the standard metrics of Accuracy, Precision, Recall and F-Measure. These were calculated using the predictive classification table, known as Confusion Matrix (Provost et al., 1998):

Table 5.1: Confusion Matrix

		PREDICTED	
		IRRELEVANT	RELEVANT
ACTUAL	IRRELEVANT	**TN**	**FP**
	RELEVANT	**FN**	**TP**

Considering Table 5.1:

TN (True-Negative) → Number of correct predictions that an instance is IRRELEVANT
FP (False-Positive) → Number of incorrect predictions that an instance is RELEVANT
FN (False-Negative) → Number of incorrect predictions that an instance is IRRELEVANT
TP (True-Positive) → Number of correct predictions that an instance is RELEVANT

Accuracy – The proportion of the total number of predictions that were correct.

(Eq. 6)

$$\boldsymbol{Accuracy}\ (\%) = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{FP} + \mathrm{FN} + \mathrm{TP}}$$

Precision – The proportion of the predicted RELEVANT pages that were correct (i.e. the number of RELEVANT pages retrieved over the total number of pages retrieved).

(Eq. 7)

$$\boldsymbol{Precision}\ (\%) = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{TP}}$$

Recall – The proportion of the RELEVANT pages that were correctly identified (i.e. the number of RELEVANT pages retrieved over the total number of RELEVANT pages).

(Eq. 8)

$$\boldsymbol{Recall}\ (\%) = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}}$$

F-Measure – A widely used performance measure that derives from precision and recall values:

(Eq. 9)

$$\boldsymbol{F - Meaure}\ (\%) = \frac{2 \times \ Recall \times Precision}{Recall + Precision}$$

The F-Measure is used, because despite Precision and Recall being valid metrics in their own right, one can be optimised at the expense of the other (Turney, 1999). For example, if all web pages are classified as *relevant*, then recall is 100%, but precision will be close to 0%. Similarly, if one web page is classified as the only *relevant* web page, then precision might be 100%, but recall would be close to 0%. To avoid such situations, a performance measure is required that only produces a high result when Precision and Recall are both balanced. The F-Measure is such a measure.

Another measure evaluated for this part of the system involves the analysis of the Receiver Operating Characteristic (ROC) space (Figure 5.1), as this shows the sensitivity (FN classifications) and specificity (FP classifications) of a test. The ROC curve is a comparison of two characteristics: TPR (true positive rate) and FPR (false positive rate) as follows:

The TPR measures the number of RELEVANT pages that were correctly identified:

(Eq. 10)

$$\boldsymbol{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

The FPR measures the number of incorrect classifications of RELEVANT pages out of all IRRELEVANT test pages:

(Eq. 11)

$$\boldsymbol{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

In the ROC space graph, FPR and TPR values form the x and y axes respectively. Each prediction (FPR, TPR) represents one point in the ROC space.

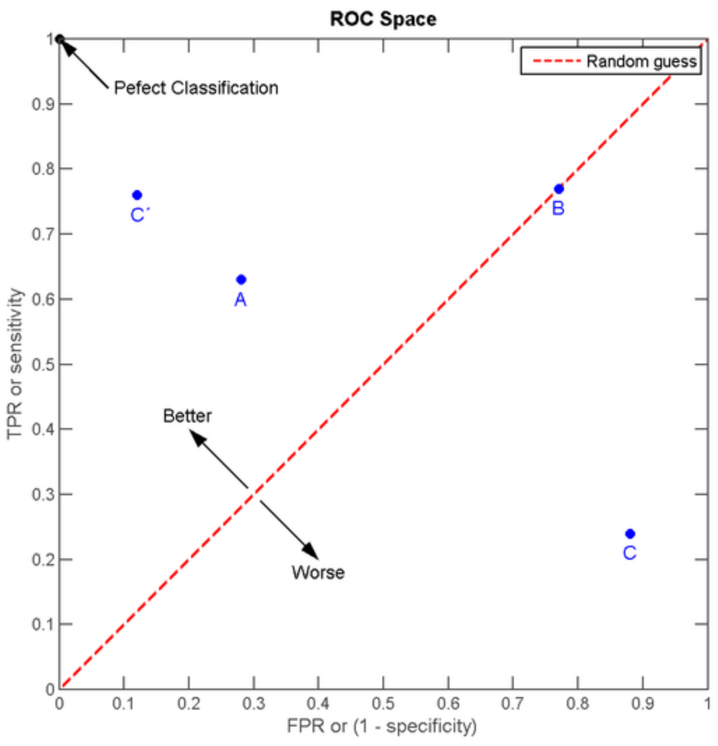


Figure 5.1: ROC Space (http://en.wikipedia.org)

There is a diagonal line that connects points with coordinates (0, 0) and (1, 1). This is called the “line of no-discriminations’ and all the points along this line (point B - Figure 5.1) are considered to be entirely random guesses. The points above the diagonal line (points A and C’ - Figure 5.1) indicate good classification results, whereas the points below the line (point C - Figure 5.1) indicate wrong results. The best prediction (i.e. 100% sensitivity and 100% specificity), also known as ‘Perfect Classification’, would be at point (0, 1). Points closer to this coordinate show better classification results than points elsewhere in the ROC space.

The CRAWLER cannot be evaluated using the above measures, because, the exact, overall number of web pages on the Web (needed for the calculations) is unknown. Similarly for the EXTRACTOR, it is impossible to find the total number of training course attributes available on the Web. Therefore, the CRAWLER is evaluated in relation to the number of web pages retrieved in comparison with other crawling systems. The EXTRACTOR is evaluated in relation to the number of generations required to produce successful REs (which means the successful extraction of course attributes) and the accuracy of the training course information extracted from each page (Accuracy, Precision, Recall and F-Measure can be partially used for this latter evaluation).

Note that, although the execution time for each of the prototype's components (i.e. CRAWLER, CLASSIFIER and EXTRACTOR) is recorded, this is not very important for this research, because the prototype will not be executed on demand. Instead, it will run undisturbed in the background and collect/update results in its own time.

5.3 DATA CORPUS

The experiments in this research were based on three main test collections. The first collection was created for the comparison of the CRAWLER with other existing crawlers. This collection consists of data obtained from 24 websites. Each website contains a number of internal web pages, ranging from a few tens to thousands. In this research, each website is referred to as the sample and each web page is referred to as the sampling unit. Therefore, the first collection consists of 24 samples and 163,340 sampling units. Table 5.2 shows the breakdown of six of the samples into sampling units, to show the variation in size of each sample.

Table 5.2: Sample Breakdown

Sample	Description	Sampling Units
www.mitregroup.co.uk	This is one of the smallest websites tested. It was recommended by previous advisors at ATM.	**19**
www.trainanddevelop.co.uk	This website shows a great variety of courses and was also recommended by previous advisors at ATM.	**348**
www.accordia.org.uk	This was chosen as it contains many .PDF documents as well as HTML pages, which need analysing.	**301**
www.ptp.co.uk	This website includes a large list of courses in a variety of areas.	**3,582**
www.dncc.co.uk	The original website recommended by ATM was www.nottschamber.co.uk. However, this website has now merged with DNCC.	**17,095**
www.underoak.co.uk	This website was one of the largest sites discovered and crawled by the CRAWLER. This site presents training course attributes differently from all the above websites (see Appendix E).	**36,216**

The above samples were chosen because they comprise various features that the CRAWLER needs to be able to manage, such as dynamic pages, large volume of data, different formats and page types etc.

The second test collection was created for the CLASSIFIER and it consists of data obtained from the sampling units in the first test collection (after the filtering of the initial set of *irrelevant* web pages, as discussed in section 4.3.1). In order to evaluate the efficiency of the CLASSIFIER on different volumes of data (Test 3 – section 5.1), the CLASSIFIER was evaluated against three differently-sized data corpuses (Table 5.3). Note that each sampling unit comprises a number of features, which range from a few to a maximum of one hundred features per sampling unit (discussed in section 4.3.3). The total number of unique features analysed is shown for each data corpus. This is to show that, although the number of unique features increases rapidly at the beginning of the classification process (when the initial knowledge base is being created), this slows down later in the process.

Table 5.3: Data Corpus for CLASSIFIER

Data Corpus	Sampling Units	Total Unique Features
#1	636	3900
#2	9436	5217
#3	105,941	10340

The third test collection was created for the EXTRACTOR. This test collection consists of the unit samples that were classified as *relevant* by the CLASSIFIER in the previous stage (the unit samples classified as *irrelevant* are removed to table CIE_Rejected_Links as discussed in chapter 4, and as such are not considered by the EXTRACTOR). Section 5.4.2 will show that the number of sampling units classified as relevant by the CLASSIFIER is 76,930. Therefore, the third test collection consists of 76,930 sampling units.

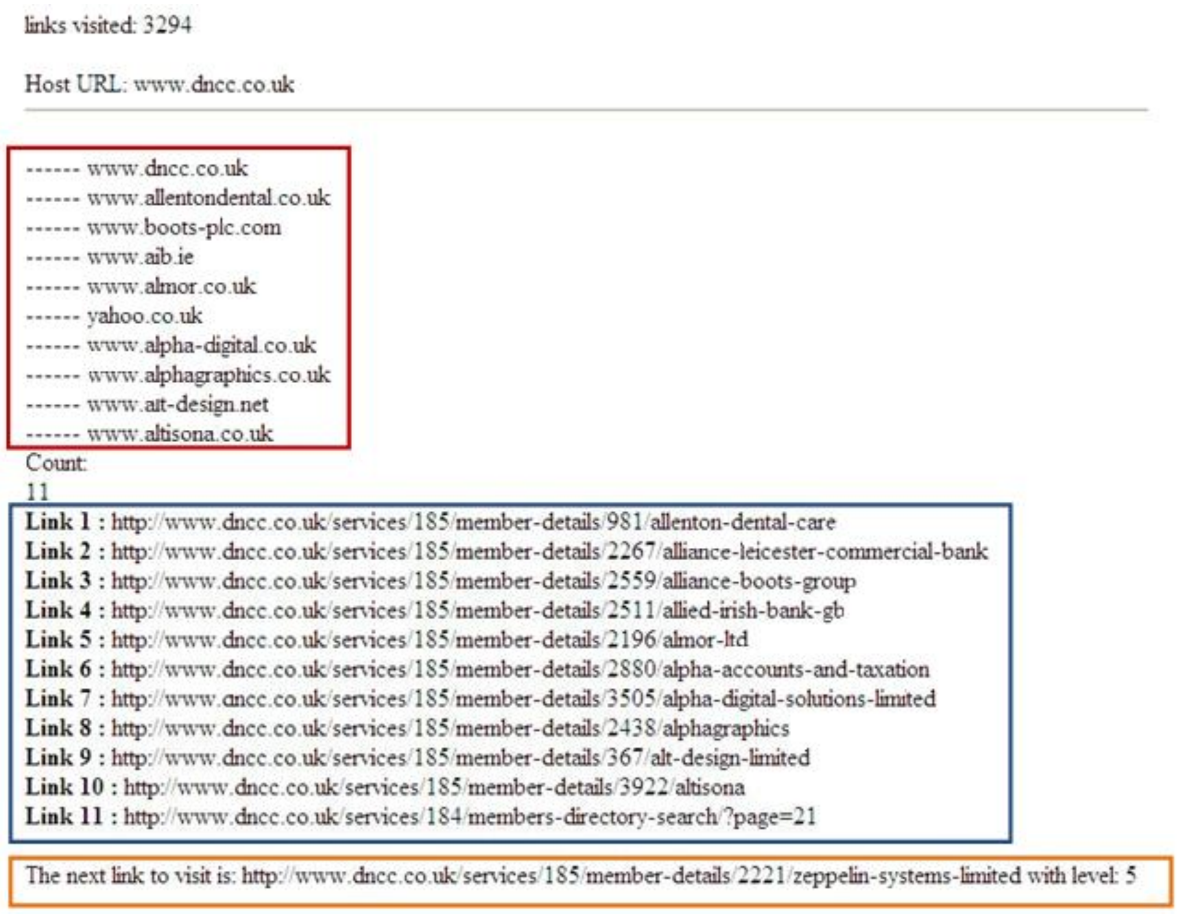
5.4 THE KEY FINDINGS OF THE RESEARCH

5.4.1 CRAWLER RESULTS

As previously mentioned, the CRAWLER was tested against 24 samples (websites). The aim was to crawl through and discover all pages within each given website, and compare the results against those of two existing crawlers (*Test 1, section 5.1*) - Web Link Validator (WLV, REL Software) and Link Checker Pro (LCP). The latter specialise in analysing websites and detecting broken links. They were chosen, as they provide similar functionalities to the CRAWLER and they are advertised and reviewed as leading solutions for website analysis and link evaluation. All three crawlers allow for the search depth to be specified, the search to focus on internal links, external links or both, and to ignore pages related to images, audio, script etc. For this experiment, the crawlers were instructed to search five levels of pages within each website, for reasons explained in section 4.3.1.

There is no user interface to the CRAWLER, as the results from the crawling process are obtained only to be further analysed by the CLASSIFIER and the EXTRACTOR. However, in order to aid testing, the program was updated to display the CRAWLER’s progress during execution (Figure 5.2).

External links (red boundary), which are not located in the rejected lists, are added to table CIE_Crawler_Seeds to be analysed in a FIFO manner. Internal links (blue boundary) are saved in table CIE_Allowed_links after they are discovered, validated and repaired and the process is then repeated for the next link to be visited (orange boundary).



```
links visited: 3294

Host URL: www.dncc.co.uk

------ www.dncc.co.uk
------ www.allentondental.co.uk
------ www.boots-plc.com
------ www.aib.ie
------ www.almor.co.uk
------ yahoo.co.uk
------ www.alpha-digital.co.uk
------ www.alphagraphics.co.uk
------ www.ait-design.net
------ www.altisona.co.uk
Count:
11
Link 1 : http://www.dncc.co.uk/services/185/member-details/981/allenton-dental-care
Link 2 : http://www.dncc.co.uk/services/185/member-details/2267/alliance-leicester-commercial-bank
Link 3 : http://www.dncc.co.uk/services/185/member-details/2559/alliance-boots-group
Link 4 : http://www.dncc.co.uk/services/185/member-details/2511/allied-irish-bank-gb
Link 5 : http://www.dncc.co.uk/services/185/member-details/2196/almor-ltd
Link 6 : http://www.dncc.co.uk/services/185/member-details/2880/alpha-accounts-and-taxation
Link 7 : http://www.dncc.co.uk/services/185/member-details/3505/alpha-digital-solutions-limited
Link 8 : http://www.dncc.co.uk/services/185/member-details/2438/alphagraphics
Link 9 : http://www.dncc.co.uk/services/185/member-details/367/alt-design-limited
Link 10 : http://www.dncc.co.uk/services/185/member-details/3922/altisona
Link 11 : http://www.dncc.co.uk/services/184/members-directory-search/?page=21
The next link to visit is: http://www.dncc.co.uk/services/185/member-details/2221/zeppelin-systems-limited with level: 5
```

Figure 5.2: Crawler Execution

Unlike the CRAWLER, LCP offers the ability to represent a website graphically. For example, Figure 5.3 shows the graphical structures of www.mitregroup.co.uk and www.accordia.org.uk.

The graphical website structures represent all possible links found in a website i.e. links from text, e-mails, images, scripts, internal and external websites etc. The user however, cannot select which links to display, which is a weakness especially for large websites. WLV also provides additional small features to help webmasters analyse websites quickly. Such features include: grouping web pages by their age, revealing orphaned, slow-loading and miniscule pages etc. All these extra features are helpful if the only purpose of the software package is the crawling of websites, however this is not the main aim of this research, thus these features would give no added benefit to the current project, however they could prove useful if requirements at ATM change in the future.

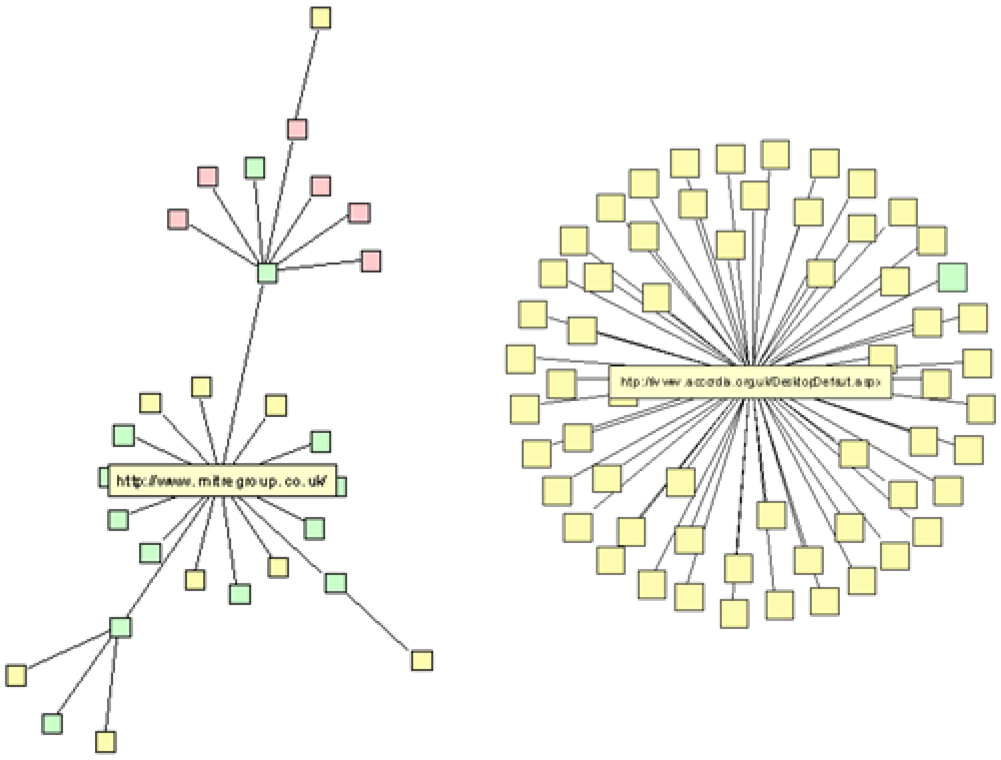

http://www.mitregroup.co.uk/
http://www.accordia.org.uk/DesktopDefault.aspx

Figure 5.3: Graphical Representation of Websites

Table 5.4 shows a comparison of the web pages crawled by each system. The results are based on internal web pages, up to search depth five. Links pointing to image and video files, scripts and style sheets, email and contact pages are not considered for this experiment. External web pages are located and stored but not crawled. The 'Total internal pages found' field includes 'Dead links' and other internal web pages.

Table 5.4: Crawler Results Comparison

#	WEBSITES	CRAWLER			WLV			LCP		
		Total internal pages found	Total external pages found	Dead links	Total internal pages found	Total external pages found	Dead links	Total internal pages found	Total external pages found	Dead links
1	mitregroup.co.uk	18	2	0	19	2	0	17	1	0
2	accordia.org.uk	298	6	1	301	6	1	296	6	1
3	trainanddevelop.co.uk	348	7	1	230	7	1	20	4	0
4	ptp.co.uk	3582	11	16	3576	11	16	-	-	-
5	dncc.co.uk	17095	23	113	17031	22	112	4562	19	88

#	WEBSITES	CRAWLER			WLV			LCP		
		Total internal pages found	Total external pages found	Dead links	Total internal pages found	Total external pages found	Dead links	Total internal pages found	Total external pages found	Dead links
6	underoak.co.uk	36216	153	1	33263	153	1	7440	191	1
7	challengeconsulting.co.uk	54	18	2	54	18	2	54	17	2
8	nec.ac.uk	2417	9	0	2417	8	0	1	177	0
9	cegos.co.uk	106	50	1	108	48	2	83	48	1
10	associatedtraining.co.uk	395	29	2	395	29	1	203	26	1
11	ontargetlearning.co.uk	113	19	0	56	19	1	13	19	0
12	chesterfield.ac.uk	1723	13	0	1723	14	0	76	5	0
13	coursesplus.co.uk	54630	52	10	37400	49	8	9890	25	5
14	eastmidlandsnti.co.uk	502	53	2	502	53	2	116	53	2
15	yourfuture-eastmidlands.co.uk	1062	75	0	1020	70	1	691	66	0
16	reedlearning.co.uk	1035	54	16	1032	54	19	440	42	16
17	loucoll.ac.uk	793	7	10	-	-	-	-	-	-
18	smart-training.com	18	7	0	18	7	0	1	7	0
19	diamondtraininggroup.co.uk	62	15	4	62	15	4	62	15	4
20	ableskills.co.uk	268	24	9	298	23	9	268	23	10
21	gbscorporate.com	290	9	3	289	10	6	80	4	6
22	barleythorpe.com	25	4	0	25	5	0	-	-	-
23	hotcourses.com	42500	17	12	39638	17	11	6	99	0
24	managementtrainingcoursesuk.co.uk	60	16	0	60	16	0	59	16	1

Figure 5.4 illustrates the results in Table 5.4 as a line chart in order to show the results from each crawling system side by side.

The figures show that the CRAWLER performed better than LCP for all websites and better than or as well as WLV for 20 out of the 24 websites. The CRAWLER, however, slightly underperformed for mitregroup.co.uk, accordia.org.uk and cegos.co.uk, where a total of six standard links were overlooked. A further 30 links were missed for ableskills.co.uk, because one of the links was evaluated as an external link, which meant that this link was not crawled further. Note that the difference in the number of dead links identified by all three crawlers, may have been influenced by the fact that the three crawlers were tested on different dates (within the same two weeks).

LCP retrieved noticeably fewer web pages from some websites than the other two crawlers. An analysis of these results showed that LCP is inconsistent in the analysis of different file types. For example, it was observed that for hotcourses.com, only 17 out of the 99 web pages categorised as external links, were in fact external. This means that LCP missed the opportunity to crawl 82 web pages. For yourfuture-eastmidlands.co.uk, LCP rejected 268 links as being script pages, however only 10 of these were script pages and the remaining 258 were normal web pages that could have been crawled. Websites ptp.co.uk, loucoll.ac.uk and barleythorpe.com have no results as LCP crashed each time it attempted to crawl these sites. WLV also crashed whilst attempting to crawl loucoll.ac.uk.

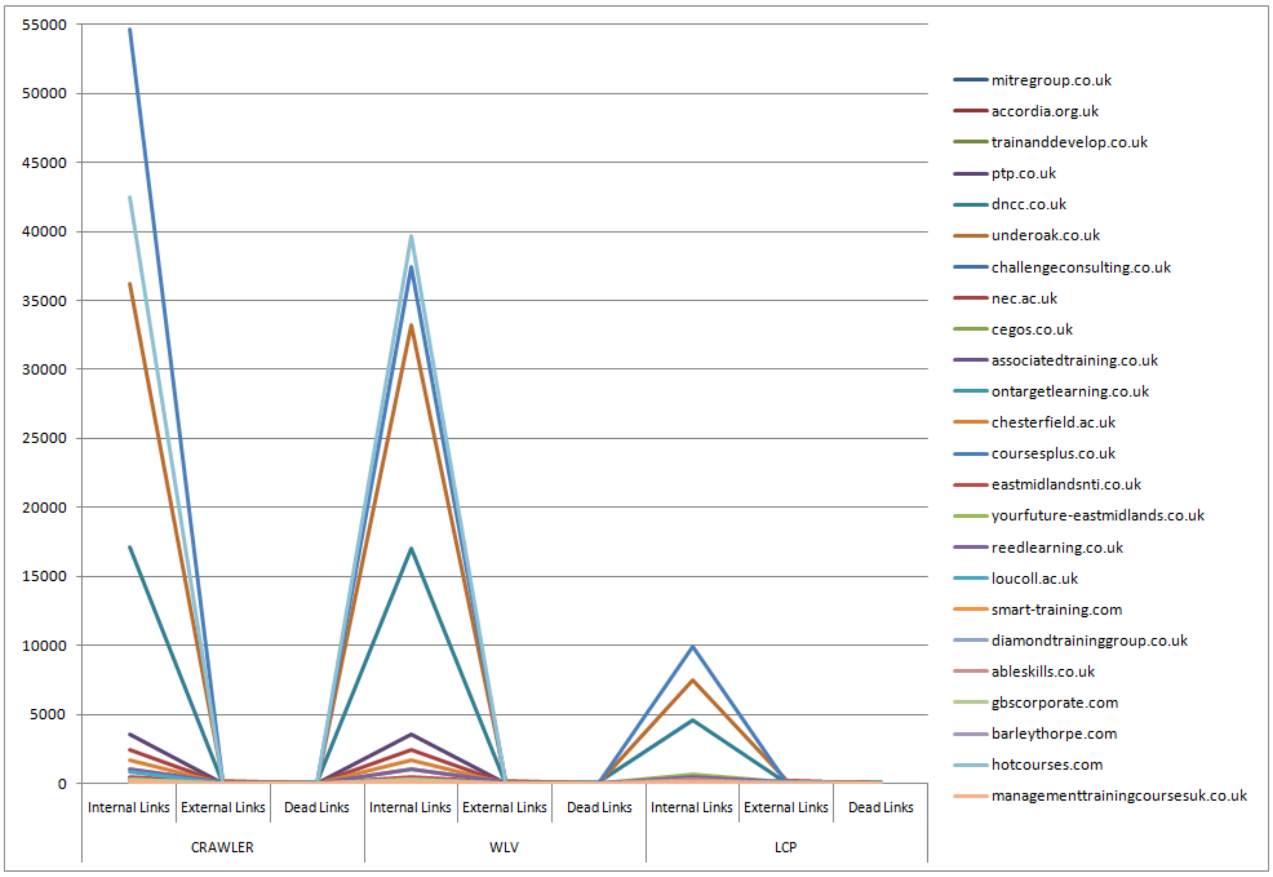


Figure 5.4: Crawler Results Comparison

Section 4.3.1 discussed the additional functionality that allows the CRAWLER to reject a number of web pages as being *irrelevant*, before the CLASSIFIER is even involved. This was achieved through a list of URL keywords (listed in Table 4.2) identified by reusing the NB approach. Table 5.5 shows the number of internal web pages retrieved by the CRAWLER from the same websites as in Table 5.4, after rejecting the URLs containing any of these keywords.

Table 5.5: Total Web Pages Retrieved after Rejected URLs

#	WEBSITES	Total Internal Pages	Decrease in web pages (%)
1	mitregroup.co.uk	8	55.6
2	accordia.org.uk	285	4.4
3	trainanddevelop.co.uk	112	67.8
4	ptp.co.uk	3084	13.9
5	dncc.co.uk	4832	71.7
6	underoak.co.uk	35983	0.6
7	challengeconsulting.co.uk	47	13.0
8	nec.ac.uk	1796	25.7
9	cegos.co.uk	73	31.1
10	associatedtraining.co.uk	118	70.1
11	ontargetlearning.co.uk	71	37.2
12	chesterfield.ac.uk	1369	20.5
13	coursesplus.co.uk	53483	2.1
14	eastmidlandsnti.co.uk	447	11.0
15	yourfuture-eastmidlands.co.uk	594	44.1
16	reedlearning.co.uk	832	19.6
17	loucoll.ac.uk	660	16.8
18	smart-training.com	15	16.7
19	diamondtraininggroup.co.uk	11	82.3
20	ableskills.co.uk	149	44.4
21	gbscorporate.com	265	8.6
22	barleythorpe.com	8	68.0
23	hotcourses.com	1650	96.1
24	managementtrainingcoursesuk.co.uk	49	18.3

The experimental results in Table 5.4 and Table 5.5 (*Test 2, section 5.1*) clearly illustrate the large number of *irrelevant* web pages that get filtered out before even being stored in the database. Specifically, 57,399 *irrelevant* links were rejected at this stage. The largest number of web pages was removed from websites: dncc.co.uk and hotcourses.com. This is because these websites do not offer just training courses, they also offer a lot of information related to: career opportunities in various sectors, scholarship opportunities for postgraduate degrees, membership services, events organised in the local area such as seminars, workshops etc.

Rejecting irrelevant links at this stage is an achievement for the following reasons:

- The *irrelevant* links are rejected before they are stored in the database, which means that the database is kept smaller in size, which improves the execution speed of the CLASSIFIER and the EXTRACTOR.
- As discussed in section 4.3.4, the links classified by the CLASSIFIER as *irrelevant* are moved to table CIE_Rejected_links, which means that this table could get very large, very quickly. However, the fact that a large number of links is rejected by the CRAWLER before they are stored in the database, means that fewer links are moved by the CLASSIFIER to table CIE_Rejected_links, which frees up valuable space in this table.
- With both the CRAWLER and the CLASSIFIER filtering out *irrelevant* web pages, there is a cleaner data corpus for the EXTRACTOR to analyse, which is very important considering the complexity of this stage.

The final test performed on the CRAWLER involved comparing the results from the CRAWLER with those of Google *(Test 3, section 5.1*). Google is used by billions of people around the world and it is also the search engine of choice at ATM, thus comparing the CRAWLER to this search engine, instead of others, was considered the obvious choice. The experiment carried out for this reason was set up to emulate a scenario common to the one advisors at ATM go through when searching for courses.

Table 5.6: Google Rank for CRAWLER Websites

#	WEBSITES	Google Page
1	hotcourses.com	4
2	managementtrainingcoursesuk.co.uk	14
3	cipd.co.uk	2
4	i-l-m.com	21
5	cegos.co.uk	7
6	ptp.co.uk	3
7	reedlearning.co.uk	16
8	ldl.co.uk	19
9	associatedtraining.co.uk	21
10	underoak.co.uk	5

Specifically, the query ‘leadership courses UK’ was used to compare the results from both crawling solutions. This query was chosen as it is one of the courses for which ATM gets many requests, but which the advisors normally decline, as the options for this course are extensive and as such it takes too long to find the most appropriate websites on the Web that offer relevant information.

For this experiment, the CRAWLER was seeded with an initial five leadership and management websites (provided by ATM) to focus the search in this domain. The program was left running until thirty websites (excluding the seed URLs provided) were discovered. The results from the CRAWLER were evaluated by the advisors at ATM and the top ten websites, as ranked by the advisors according to relevance, were searched for amongst Google’s results for the same query. Table 5.6 shows the page position of each of these websites on Google (on the 3rd of July, 2010).

Table 5.6 shows that none of these websites were found on the first page of Google. Furthermore, all of the websites listed in Google’s first results’ page (Table 5.7) were considered less *relevant* by the advisors at ATM then all thirty websites discovered by the CRAWLER in this experiment.

Table 5.7: Google’s 1st Page Results for Query: ‘Leadership Courses UK’

#	Google – Page 1 Results
1	www.impactfactory.com
2	www3.open.ac.uk
3	www.vitae.ac.uk
4	www.totalsuccess.co.uk
5	www.le.ac.uk
6	www.medicalinterviewsuk.co.uk
7	www.ksl-training.co.uk
8	www.spearhead-training.co.uk

This shows that a properly seeded CRAWLER achieves the wanted results faster than Google.

5.4.2 CLASSIFIER RESULTS

Test 4, section 5.1

As previously mentioned, the NB approach was chosen for the classification process, as an initial manual analysis of websites showed conditional independence amongst the features extracted from each web page. The NB Classifier was designed and developed (as explained in section 4.3.4) and successfully tested on a number of websites, as will be shown below. It was after all this that it was decided to run tests to formally prove the feature conditional independence. This was considered a formality due to the impressive results of the NB

CLASSIFIER. However, the feature independence tests showed a conditional dependence amongst some of the features, such as: price, date and location, as shown below:

Two features F_1 and F_2 are considered conditionally independent, given condition C (i.e. web page is *relevant*), if and only if:

(Eq. 12)

$$P(F_2|C) = P(F_2|F_1 \wedge C) \text{ where } P(F_2|F_1 \wedge C) = \frac{P(F_2 \wedge F_1 \wedge C)}{P(F_1 \wedge C)}$$

Considering:

F_1 = 'price'
F_2 = 'date'

No. of relevant pages containing F_1	= 1278
No. of relevant pages containing F_2	= 1460
No. of relevant pages containing F_1 and F_2	= 876
Total no. of relevant pages	= 16822

$$\left.\begin{array}{l} P(F_2|F_1 \wedge C) = \dfrac{876}{1278} = 68.5\% \\ \\ P(F_2|C) = \dfrac{1460}{16822} = 8.7\% \end{array}\right\} P(F_2|C) \neq P(F_2|F_1 \wedge C) \Rightarrow \mathit{Conditional\ Dependence}$$

Similar figures were calculated for feature pairs: <price, location> and <date, location>.

The experiment was also carried out on features, less obviously related to each other, such as: <price, appoint>, <action, construct> etc. The calculations showed a much smaller difference in probability values. For example, for F_1 = 'price' and F_2 = 'appoint':

$$P(F_2|F_1 \wedge C) = 1.56\% \quad \mathit{and} \quad P(F_2|C) = 0.81\%$$

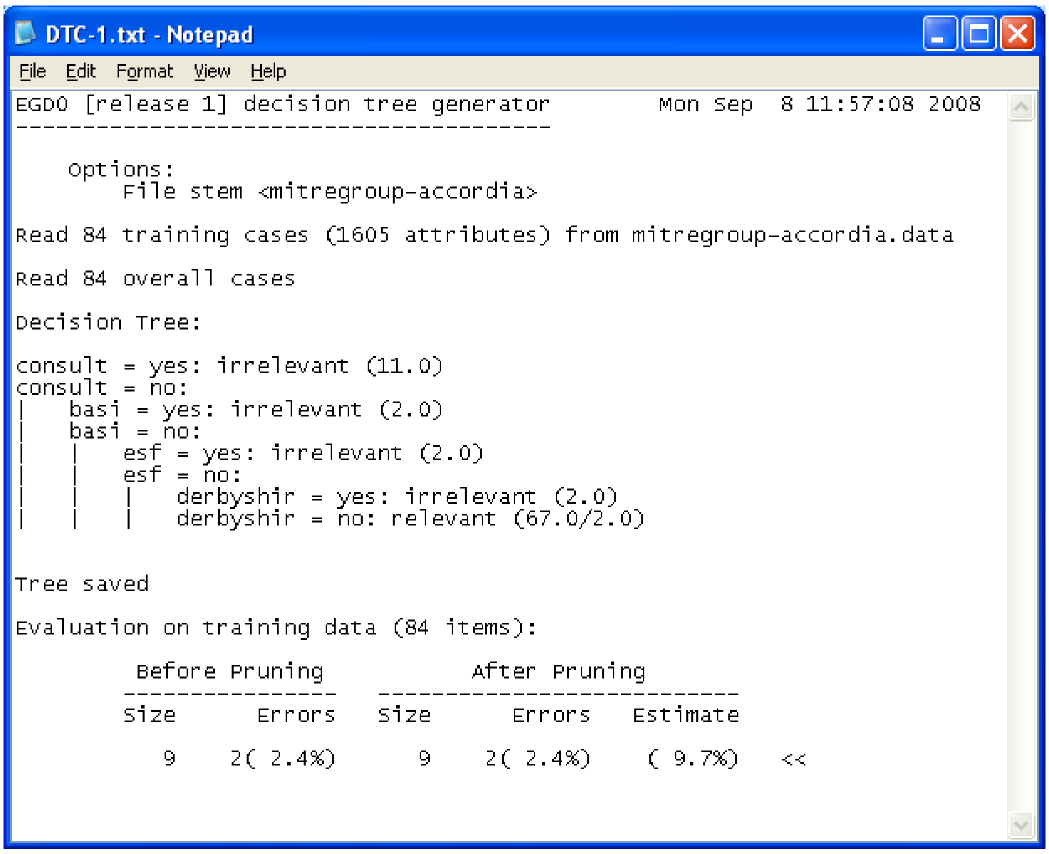
Despite the conditional dependency calculated for some of the features, the NB approach was tested on a large number of web pages and proved very successful (as discussed below). Additionally, the NB approach was appropriate for this research as it was suitable for the hardware specification available at ATM at the time.

Tests 5 & 6, section 5.1

As mentioned in section 5.3, the CLASSIFIER was tested against three data corpuses. Each corpus changes considerably in size. This was deliberate in order to test the CLASSIFIER on different volumes of data. Corpus #1 and #2 were also examined by a Decision Tree Classifier (DTC) and a Neural Network Classifier (NNC). The results were compared to determine which classifier is better at analysing attribute data from training web pages.

The DTC is a ‘C’ program, based on the C4.5 algorithm (Quinlan, 1996), written to evaluate data samples and find the main pattern(s) emerging from the data. For example, the DTC may conclude that all the web pages containing a specific word are all *irrelevant* (Figure 5.5). Figure 5.5 shows that the DTC manages to identify a pattern in the data (from a small data corpus of 84 sampling units) with an accuracy of 97.6%. The pattern identified is that if the stemmed feature ‘derbyshir’ is found in a web page, then that web page is *irrelevant*.

More complex data samples however, may result in more complex configurations found. The DTC can examine the entire data sample and discover all the potential patterns within the data, or it can be instructed to split the sample into a training set and a testing set. This way, the DTC will only train itself on the training data and then test its ‘acquired knowledge’ on the testing set.



```
EGDO [release 1] decision tree generator        Mon Sep  8 11:57:08 2008
----------------------------------------

    Options:
        File stem <mitregroup-accordia>

Read 84 training cases (1605 attributes) from mitregroup-accordia.data

Read 84 overall cases

Decision Tree:

consult = yes: irrelevant (11.0)
consult = no:
|   basi = yes: irrelevant (2.0)
|   basi = no:
|   |   esf = yes: irrelevant (2.0)
|   |   esf = no:
|   |   |   derbyshir = yes: irrelevant (2.0)
|   |   |   derbyshir = no: relevant (67.0/2.0)


Tree saved

Evaluation on training data (84 items):

         Before Pruning           After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate

           9    2( 2.4%)      9    2( 2.4%)    ( 9.7%)   <<
```

Figure 5.5: DTC – Examination of Entire Data Sample

The NNC used is also a ‘C’ program, based on the work published by Fletcher and Hinde (1994, 1995, 1996), and Hinde et al. (1997). MATLAB’s Neural Network toolbox[8] could have also been used, however, in past experiments, MATLAB managed approximately two training epochs compared to the 60,000 epochs achieved by NNC in the same timeframe.

The first corpus consisted of 636 sampling units. In this test, all three classifiers were trained with 105 units and tested with the remaining 521 sampling units. The second corpus consisted of 9436 sampling units, from which 711 units formed the training set, whereas the remaining

[8] TheMathsWork: http://www.mathworks.com/products/neuralnet

8725 sampling units formed the test set. The third corpus consisted of 105,941 sampling units. The training set included 21,100 units, whereas the test set was created with the remaining 84,841 sampling units. Note that the training sets for each data corpus were created by randomly selecting positive (relevant) and negative (irrelevant) examples (sampling units) from the available list of sampling units.

Table 5.8 shows a comparison of the results obtained by the three classifiers using the first and second data corpus. A breakdown of how these figures were achieved, using the calculations in section 5.2, is explained in Xhemali et al. (2009).

Table 5.8: Classification Result Comparison for Data Corpus #1 & #2

Data Corpus	Classifier	Accuracy	Precision	Recall	F-Measure
#1	CLASSIFIER	**97.9%**	**99.2%**	**98.6%**	**98.9%**
	DTC	91.9%	97.3%	94.3%	95.8%
	NNC	89.5%	95.3%	93.7%	94.5%
#2	CLASSIFIER	**95.2%**	**99.4%**	95.2%	**97.3%**
	DTC	94.9%	98.3%	**95.9%**	97.1%
	NNC	-	-	-	-

In relation to the results of the NNC for the second data corpus, these are missing, because it was observed during these experiments that the Neural Network (NN) approach did not manage well with the increasing number of attributes. The vocabulary used in data corpus #2, consisting of 5217 unique features, was initially mapped onto a NN with 5217 inputs, one hidden layer with 5217 nodes and one output, in keeping with the standard configuration of a NN, where the number of midnodes is the same as the number of inputs. A fully connected network of this size would have over 27 million connections, with each connection involving weight parameters to be learnt. Attempting to create such a network resulted in the NN program failing to allocate the required memory and crashing.

After reconsidering the function's complexity, it was decided to change the number of midnodes to reflect this complexity. Thus, a NN with 5217 inputs, one output and only 200 midnodes was created. This worked well and the resulting NN successfully established all connections. However, it was realised that the NN would need to be extended (more nodes and midnodes created) to model any additional new features, each time they are extracted from future web pages. This would potentially take the NN back to the situation where it fails to make all the required connections. This would be an unacceptable result for ATM. Technology exists for growing nodes, however this would be complex and expensive. Furthermore, the NNC took 200 minutes to train, which is much longer than the other classifiers, which took few minutes for the same training sample. Therefore, it was decided not to proceed with NNs any further, as they would be unsuitable for this project.

Table 5.8 shows that both the CLASSIFIER and the DTC achieve impressive results in the classification of attribute data in the training courses domain. The DTC outperforms the CLASSIFIER in execution speed and Recall value for data corpus #2, however the CLASSIFIER achieves higher Accuracy, Precision and most importantly, overall F-Measure value, which is a great result. This success is further confirmed by the comparison of the two classifiers on the ROC space (Figure 5.6), where it is clear that for both data corpuses, the result set from the CLASSIFIER falls closer to the 'perfect classification' point than the result set from the DTC. (The ROC space was created using the values in Table 5.9, following the calculations in (Eq. 10 and (Eq. 11.)

Table 5.9: ROC Space – TPR and FPR Values

Data Corpus	Classifier	FPR	TPR
#1	CLASSIFIER	0.235	0.986
	DTC	0.765	0.942
#2	CLASSIFIER	0.0509	0.9523
	DTC	0.1397	0.9589

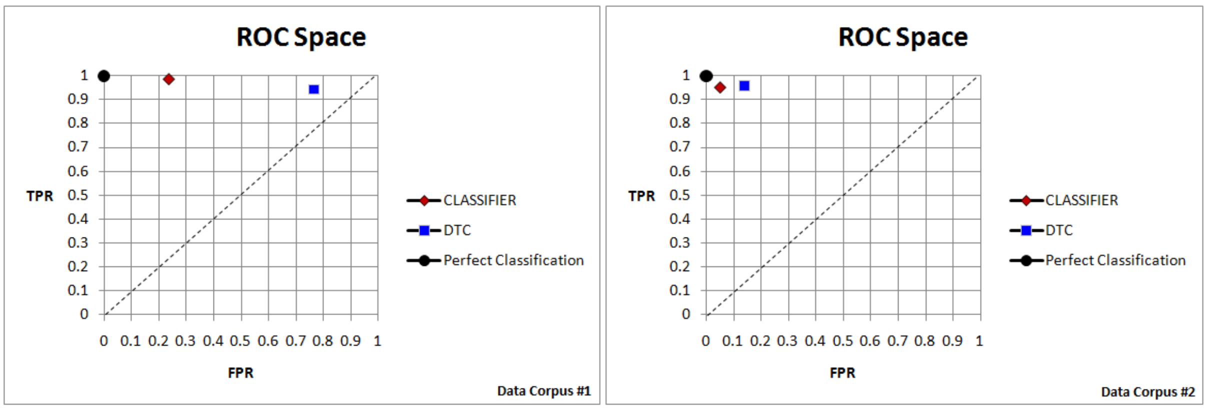


Figure 5.6: ROC Space – Data Corpus #1 & #2

The third data corpus was applied only to the CLASSIFIER, due to the complexity of preparing the data sample for the DTC and the limited timeframe. Unlike the CLASSIFIER, for which the data sample is prepared through simple SQL queries, the data sample for the DTC requires much more preparation time. The sample consists of two files: filename.data and filename.names (Table 5.10).

Table 5.10: DTC Training Data Sample

filename.data	filename.names
	relevant, irrelevant.
yes, no, no, no, no, no, no, irrelevant	account: yes, no.
no, yes, no, no, no, no, no, irrelevant	analysi: yes, no.
no, no, no, no, yes, no, no, irrelevant	agenc: yes, no.
yes, no, no, no, no, no, yes, relevant	advanc: yes, no.
yes, no, yes, no, yes, no, yes, relevant	abil: yes, no.
no, no, no, no, yes, no, yes, relevant	achiev: yes, no.
no, no, no, no, no, no, yes, irrelevant	apprais: yes, no.

The filename.names contains a list of all the unique features used to train the system. The filename.data acknowledges these features and establishes whether a page contains each feature (yes) or not (no). This file also informs the system of the collection of features that make a page either *relevant* or *irrelevant*.

The format of the files cannot be changed; even the absence of a comma or the addition of a blank line would cause the program to fail. The example in Table 5.10 shows only seven features being used by the DTC. For the second data corpus in this research, thousands of features were manually analysed and counted, before a final 5217 attributes were used by the DTC. Despite the use of Microsoft Excel to automate the analysis of certain parts of the sample, this data sample was completed in 15 working days.

Table 5.11 shows the results obtained by the CLASSIFIER for the classification of the data in corpus #3. The results show an improvement from those of data corpus #2, which shows that the CLASSIFIER gets better as the ‘knowledge base’ grows.

Table 5.11: Classification Results for Data Corpus #3

Data Corpus	Classifier	Accuracy	Precision	Recall	F-Measure
#3	CLASSIFIER	96.1%	99.5%	96.9%	98.2%

Test 7, section 5.1

For this experiment, the CLASSIFIER was stripped of the additional steps given to it in this research such as: the calculation of the believed probability and the exploitation of important web structures such as the page TITLE, Link and META Tag information when classifying web pages. The CLASSIFIER was then tested again, against the data in all three corpuses. Each test revealed a decrease in accuracy as shown in Table 5.12.

Table 5.12: Test 5 Results

	Data Corpus #1	Data Corpus #2	Data Corpus #3
Accuracy	93.2%	88.2%	88.7%
Decrease	4.7%	7.1%	7.4%

Table 5.12 shows that the average increase in accuracy for the modified CLASSIFIER is 6.4%. The results also show that the changes made to the CLASSIFIER have a bigger impact on the larger data samples; the biggest increase (7.4%) occurred for Data Corpus #3, whilst the smallest increase (4.7%) occurred for the smallest data corpus (#1).

Test 8, section 5.1

As previously mentioned, the original CLASSIFIER was developed in PHP (version 5.2.6) and it was converted to VB.NET after the decision was made to also use the NB approach for the evaluation of the genetically evolved REs for the extraction of course names.

Section 3.4.5 discussed the advantages and limitations of each programming language, with VB.NET winning for better memory management, error debugging and general consistency and PHP winning for simplicity, particularly in relation to the abundance of in-built commands related to web applications.

Both, the PHP and the VB.NET CLASSIFIER were tested on the data from Data Corpus #1 and #2. The experiments resulted in very similar Accuracy, Precision, Recall and F-Measure values (as shown in Table 5.8). A big difference, however, relates to the execution speed recorded for each classifier, with the VB.NET CLASSIFIER achieving a 35% faster execution speed than the PHP CLASSIFIER. This improvement could be related to the fact that the REFRESH functionality (explained in section 3.4.5) was no longer required in the VB.NET version, as VB.NET is very efficient in managing application connections to the SQL Server Database and memory allocations (through the Garbage Collection mechanism, which releases memory from unused objects and components in an application).

VB.NET is also a better choice for this project, because it is better compatible with Windows operating systems (the only operating system available at ATM). PHP is strong in Linux or Unix environments, however it has proven to be inconsistent in Windows. Furthermore, the process of installing/updating PHP for a Windows machine is complex and error prone and with new versions of PHP being released almost monthly, ATM would have to be involved in this process, which is far from ideal.

For these reasons, it is advised that the CRAWLER is also translated to VB.NET in the future.

5.4.3 EXTRACTOR RESULTS

The experimental results for this part of the system have been collected from a number of tests (Test 9 – Test 14, section 5.1). Each of these tests is discussed in detail below:

Test 9, section 5.1

This stage of the experiments involved a number of tests to establish the optimal GP parameters necessary for the EXTRACTOR. Some of these results have been discussed throughout the thesis, thus, they are listed here without further repetition.

- **Population size**: It was observed that a population of ten individuals is a good compromise between convergence speed and computational resources.
- **Tournaments**: Based on this population size, experiments showed that the best results were achieved with a tournament size of 40% i.e. 4 genotypes, when selecting parents for reproduction.
- **Fitness Target**: The fitness score that terminates the execution of the evolutionary system is one (1).
- **Crossover & Mutation Rates**: Whilst the probability rate (50%) chosen for the Uniform Crossover was never in question, it was discovered that the best choice for mutation was the mutation of one gene per genotype, regardless of the number of genes in that genotype.

Test 10, section 5.1

The analysis of a number of training web pages showed that the course names/titles are displayed in at least three different areas on the page:

- As part of the overall web page title (<title> ... </title>)
- As a large heading on the page (<H...>...</H...>)
- In a table (<table...>...</table>)

As previously mentioned, the NB approach (discussed in section 4.3.4) was reused to evaluate the REs evolved for the extraction of training course names, by calculating fitness scores for each evolved RE. For this experiment, the EXTRACTOR was trained with positive (course titles) and negative examples from 16,930 mixed web pages. An additional 60,000 web pages were then used to test this part of the prototype. As mentioned in section 4.4.2.5, the fitness scores were also influenced by other determining factors such as: the length of the evolved RE, the length of the extracted data and the efficiency of the RE's components. Note, that the main score for REs in this category comes from the probability value computed by the NB network. If the additional points (from the remaining criteria) take the score to above 1, then the final score is rounded to one and the GP process is terminated.

Table 5.13 shows a sample of the Regular Expressions (REs) evolved, the areas of the web pages for which the REs were aiming and the fitness scores assigned to them.

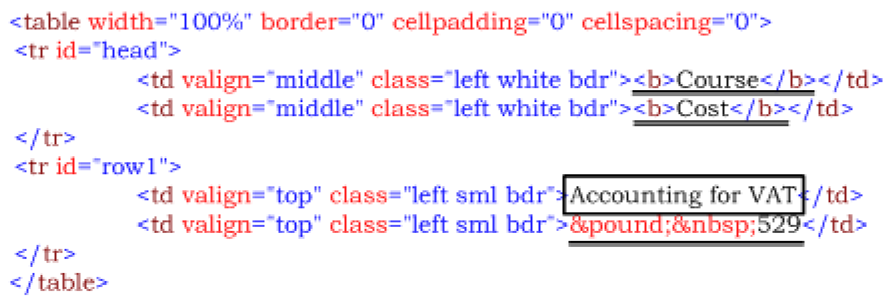
Table 5.13: Evolved Title REs

ID	Evolved RE	Target Area	Fitness Score
1	<title.*?><title.*?>([\s]?.*?[\s]?)</title></title>	Page Title	0
2	<title>.*?[\s]?.*?</title>	Page Title	0.83
3	(?<=<title>).*?(?=</title>)	Page Title	0.92
4	<title.*?>.*?</title>	Page Title	1
5	<td.*?><title.*?>([\s]?[\s]?[\s]?)<\/title><\/td>	Heading	0
6	<H1.*?>.*?</H1>	Heading	1
7	<H2.*?>.*?</H2>	Heading	0
8	<title.*?><tr[\s]?id="row1".*?>(.*?[\s]?.*?)<\/tr[\s]?id="row1"><\/title>	Table	0
9	<tr.*?><td.*?>.*?</td>	Table	0.56
10	<tr[\s]?id="row1".*?>[\s]?<td.*?>.*?</td>	Table	1

Note: All the above REs, apart from RE-3, extract some HTML Tag information together with the data, however this is not an issue as this information is removed before the data is sent for evaluation to the Fitness function. Also, note that the Fitness Scores assigned to the REs in Table 5.13 were computed in relation to specific web pages; the same RE may be awarded different Fitness Scores for different web pages, due to differences in page layouts and content.

- RE-2 and RE-4 may look very similar, however RE-2 has had points deducted for having an inefficient component collection (".*?[\s]?.*?" – a non-greedy search (.*?) followed by an optional space ([\s]?) followed by another non-greedy search captures the same information as only one non-greedy search (.*?)). Note that a template with the necessary heuristics has been manually created to identify inefficient components.

- RE-6 and RE-7 also look very similar, however they have been awarded opposite fitness scores. This is because they were applied to different web pages. Some web pages do display course name as level-2 headings (<H2>), however the web page, for which RE-7 was evolved, was not in this category and no results were captured.
- REs 1, 5 and 8 were also scored zero, because, although they are syntactically valid, they did not extract anything from the web pages on which they were tested.
- RE-9 was scored low because it was not specific enough. A small print screen of the web page against which it was tested, is shown in Figure 5.7. This clearly shows that RE-9 extracts all of the underlined information, rather than just the name of the course - "Accounting for VAT".
- REs 4, 6 and 10 were obtained in the 1st, 2nd and 58th generations respectively. RE-10 took longer due to its increased complexity.
- In these experiments, the maximum number of generations run per web page was 500.



```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr id="head">
        <td valign="middle" class="left white bdr"><b>Course</b></td>
        <td valign="middle" class="left white bdr"><b>Cost</b></td>
</tr>
<tr id="row1">
        <td valign="top" class="left sml bdr">Accounting for VAT</td>
        <td valign="top" class="left sml bdr">&pound;&nbsp;529</td>
</tr>
</table>
```

Figure 5.7: Web Page Sample

The results of the NB evaluation were analysed using the standard metrics of Accuracy, Precision, Recall and F-Measure (discussed in section 5.2). These were calculated based on the Confusion Matrix in Table 5.1. In this case, however, the elements in the Confusion Matrix represent the following:

TN (True-Negative) → Number of correct predictions that an RE is *unsuccessful*
FP (False-Positive) → Number of incorrect predictions that an RE is *successful*
FN (False-Negative) → Number of incorrect predictions that an RE is *unsuccessful*
TP (True-Positive) → Number of correct predictions that an RE is *successful*

The terms 'successful' and 'unsuccessful' are used to describe REs that extract (or do not extract) relevant course names from the web pages on which they are applied.

It is important to mention here that all of the websites encountered so far follow the same layout and general style for all their internal web pages. This means that, potentially, the same RE can be successful on all the web pages of the same website. The EXTRACTOR takes this into consideration and it only attempts to evolve new REs for a particular page, if:

- The REs saved against different web pages belonging to the same website (i.e. the same Allowed_host_link – table CIE_Allowed_links) are determined unsuccessful for the page in question.

- The REs saved against different web pages belonging to different websites are determined unsuccessful for the page in question (this is to allow for cases when different websites use the same formatting style).

The experiments carried out on 60,000 unseen-before web pages, retrieved from 24 different websites, resulted in the evolution of four successful REs (REs 4, 6, 7 and 10 - Table 5.13), which were effective in extracting relevant course names from this data corpus with an Accuracy of 94.4%. Out of the 3384 (5.64%) web pages for which the EXTRACTOR was unsuccessful, the system failed to evolve any successful REs for 372 web pages (0.62%) and extracted partially-relevant or irrelevant course names from the remaining 3012 web pages (5.02%).

One of the reasons why the EXTRACTOR failed to evolve any successful REs for 372 web pages can be linked to the fact that the CLASSIFIER was not 100% accurate and thus 0.55% (Precision value for the CLASSIFIER was 99.5%) or 582 web pages remained in the database for the EXTRACTOR to analyse, despite these links containing no course information. These proved confusing for the system.

Test 11, section 5.1

As discussed in chapter 4, the NB approach was reused to aid the fitness scoring of each RE evolved for the extraction of course names. This in itself is not a novel idea, however the way it is applied in this research makes it new and successful. Specifically, information collected by the NB network, is passed from generation to generation, independently of the genetic evolution process, which means that each generation has additional knowledge to help it evolve optimal offspring faster.

In order to test whether or not optimal offspring are evolved faster through this additional knowledge, the EXTRACTOR (for course names) was changed to:

- still use the knowledge base created by the NB network to aid the evaluation of the evolved REs.
- only update this knowledge base once, after the termination of the genetic evolution process (for any of the reasons listed in section 4.4.2.7), instead of updating it after the decision made on each evolved RE.

It is important to note that a touch of luck is involved in getting to a perfect solution quickly from evolutionary experiments. This is because, depending on the crossover of the genes and particularly on the (random) gene that gets chosen for mutation and the outcome of the mutation itself, a perfect solution may not be reached at the same time; even experiments carried out on the same system at different times, may not arrive at the same solution at the exact same generation.

For this reason, the above experiment was executed three times, to avoid misinterpreting lucky results for real results. As expected, all three experiment runs resulted in the evolution of RE solutions at much later generations than those presented in Table 5.13 (REs 4, 6, 7 and 10). Specifically, the quickest REs were evolved in generations [22nd, 30th, 30th and 106th] respectively, in comparison to the [1st, 2nd, 9th and 58th] generations achieved previously.

Test 12, section 5.1

This experiment was set up to test the EXTRACTOR for the extraction of course attributes such as: prices, locations and dates. As explained in section 4.4.2.5, the NB approach could not be used to evaluate the Regular Expressions (REs) evolved for the extraction of such attributes, thus a different fitness test needed to be used. The main approach followed for the validation of these attributes was based on utilising the training course information already stored in the database. Specifically, each RE output for a specific URL is checked against the data stored for that URL in table CIE_Courses[9]. The necessary changes are made if it is found that the data has changed from when it was last saved in the database. For cases when the information being evaluated does not exist in CIE_Courses, the fitness function evaluates the RE outputs based on a number of criteria (specified in section 4.4.2.5).

Table 5.14: Price, Date and Location REs

ID	Evolved RE	Target	Gen
1	\d{1,2}[/-]\d{1,2}[/-]\d{2,4}.*?\b[A-Z][a-z]+\b	Date-Location	186th
2	price.*?(£|&pound;)\b\d.*?\b(\.\d{2})?	Price	1360th
3	(price|cost|fee).*?(£|&pound;)\b\d.*?\b(\.\d{2})?	Price	1605th
4	(£|&pound;)\d[\d\,\.]+	Price	102nd
5	(£|&pound;)[1-9]{1}[0-9]{0,4}(\.[0-9]{2,3})?	Price	209th
6	\w*[\s]\d{2}/\d{2}/\d{4}	Location-Date	56th
7	(\w*[\s]\d{2}/\d{2}/\d{4})|(\d{2}/\d{2}/\d{4}[\s]\w*)	Date-Location OR Location-Date	2560th
8	<td.*?>(\d{1,2}[/-]\d{2}[/-]\d{2,4})(.*?)</td>	Date	49th
9	(?<!=)\b[0-9]{1,2}[/-][0-9]{2}[/-][0-9]{2,4}\b	Date	203rd
10	(?<!=)[0-9]{1,2}[/-][0-9]{2}[/-][0-9]{2,4}	Date	191st
11	<td.*?>(\b[A-Z][a-z]+\b).*?(\d{2,4}[/-]\d{2}[/-]\d{2,4}).*?(£|&pound;)\d{1,4}\.\d{0,2}	Location-Date-Price	22036th
12	<td.*?>.*?</td>	Date OR Location OR Price	2nd
13	\d+(th|rd|st)?[\s][A-Z][a-z]+[\s]\d{2,4}	Date	402nd
14	<td.*?>(\w*).*?(\d{2,4}[/-]\d{2}[/-]\d{2,4}).*?(£|&pound;)\d[\d\.\,]+	Location-Date-Price	10160th
15	(<td.*?>(\w*).*?(\d{2,4}[/-]\d{2}[/-]\d{2,4}).*?(£|&pound;)\d[\d\.\,]+)|(\d{2,4}[/-]\d{2}[/-]\d{2,4}).*?(<td.*?>(\w*).*?(£|&pound;)\d[\d\.\,]+)	Location-Date-Price OR Date-Location-Price	26982nd

The analysis of a number of training web pages has shown that information on course prices, dates and locations is presented either as part of a table, or as free text. However, either

[9] Note, that this has been possible, because this research did not start with an empty database. There was already a large number of (mostly out of date) course attributes saved in ATM's database, which were used here.

alternative includes a variety of styles and formats. Appendix E shows some of the ways training course web pages advertise courses.

This experiment was performed on a data corpus of 76,930 web pages, obtained from 24 different websites (listed in Table 5.5). Table 5.14 shows a list of the more successful REs evolved, the course attributes for which they were aiming and the generation on which they were created.

Note that, similarly to the evolved title REs presented in Table 5.13, the above REs also capture HTML tag information together with the data, however this information is removed before the data is evaluated by the fitness function. Also, the above REs were generated for specific web pages and not as general solutions for every web page analysed.

- RE-1 was awarded a fitness score of 1 for website: trainanddevelop.co.uk, because it managed to extract relevant Date and Location information from this site, without using inefficient component collections and whilst maintaining a reasonable length.
- RE-2 and RE-3 are the same apart from the fact that RE-3 expands the range of data that may be captured by allowing the keyword to be either ‘price’, ‘cost’, or ‘fee’. RE-2 limits this to just ‘price’. Although RE-3 is more generic and as such may apply to more web pages than RE-2, it is penalised for being longer than RE-2. Both REs were successful however for pages from website: dncc.co.uk
- RE-4 was awarded points for being concise and for not containing inefficient components, which contributed to its faster generation.
- RE-6 and RE-7 are again very similar, however RE-7 generalises the solution by allowing for ‘location-date’ and ‘date-location’ combinations to be captured, rather than just ‘location-date’ allowed by RE-6. These were successful for pages in websites: mitregroup.co.uk and cegos.co.uk.
- RE-8 works well for websites that only advertise courses that run on one date, however, for many different dates, this RE would not capture the corresponding locations or prices. Similarly, RE-13 also captures dates only. This RE is successful, however, for pages in website: dncc.co.uk
- RE-9 and RE-10 look almost identical, however they are different. (?<!=) is a Negative Lookbehind operator, which in this case means ‘match any position before which the prefix “=” is not found. However, RE-9 only discards the closest digit to “=” rather than the entire date closest to “=”, which is incorrect. RE-10 rectifies this problem by using “\b ... \b” which tell the RE to only look for full words (or structures) such as a complete date. Unfortunately, the fitness function is not capable of making this distinction, because both REs return a list of syntactically valid dates and following the fitness criteria, RE-10 gets penalised in comparison to RE-9 for being longer.
- RE-11, RE-14 and RE-15 capture information about all three attributes: location, date and price. RE-15 is the most generalised one out of the three as it allows for different combinations of the attributes. All three REs took the longest to be generated however, due to their higher complexity.
- RE-12 was the fastest RE generated. However, this RE is very general and it captures every table row there is on a web page.

The fitness test for this part of the research is designed to help with the dynamic nature of the Web by:

- Cleaning the web page from unnecessary HTML code, such as headers, footers, comments, navigation bars etc. prior to the execution of the RE against that page
- Removing HTML data from the captured information before analysis
- Splitting the ‘cleaned’ information into different groups
- Analysing each group to determine whether or not they are syntactically valid prices or dates
- Checking locations against a complete list of towns, cities and counties in the UK to eliminate situations when REs return normal text rather than locations
- Identifying the index position of extracted matches (for cases when singular attributes are captured), then executing previously evolved REs to attempt to capture the potentially missing information. For example, if a location was found at character 160, then previously evolved, price and date REs are executed between characters 130 and 190 to try to find this information.

The results obtained from this experiment were analysed in relation to two separate areas:

- The success of the EXTRACTOR based on the comparison of the captured information with existing information in the database
- The success of the EXTRACTOR based on the evaluation of entirely new information

65% of the web pages analysed in this experiment (i.e. 50,004 web pages) were already present in the database (associated with mostly out of date course information). The remaining 35% (26,926 web pages) had no information associated to them in table CIE_Courses.

Based on the comparison of the captured information with information already in the database, the EXTRACTOR correctly updated the course information for 45,503 web pages, which means an accuracy of 91%. An analysis of the remaining 4501 web pages, which were wrongly or partially updated showed that the majority of the mistakes happened for cases when singular REs (i.e. REs capturing singular information e.g. just price) were compared against web pages associated with multiple course attributes (i.e. multiple dates, locations and/or prices). In these cases the system gets confused as to which attribute collections to update, not knowing all the attributes.

Based on the evaluation of entirely new information, the EXTRACTOR correctly extracted prices, dates and locations from 18,010 web pages, which means an accuracy of 66.9%. The EXTRACTOR managed to extract partial information, for all of the remaining 8,916 web pages. The difficulty was again related to situations when the system did not know which attributes were associated with which other attributes, due to these attributes being captured separately by singular REs.

The latter results are lower than those obtained from other parts of this research, however work is already being done to improve these outcomes. As previously mentioned, another research project is currently underway to deal with cases when the EXTRACTOR fails to capture the correct training course information automatically (Siau et al., 2010). This research will involve the users at ATM to highlight the areas of interest in the unsuccessful pages and then generate the correct rules to get to this information in the future. This additional information will be added to the EXTRACTOR’s list of rules, which means that the users will not be required to deal with the same web page twice (see section 6.3).

Tests 13 & 14, section 5.1

Section 4.4.2.4 gave a very detailed discussion of the novel Genotype to Phenotype Mapping approach used in this research for the translation of genotypes to syntactically valid REs. In order to prove the adaptability of this approach to different domains, the mapping process was adapted to work for a completely different domain, that of 'Software Statements and Structures'. The effort involved for the alterations was minimal, as was explained in Xhemali et al. (2010-b).

Having proven that the genotype-phenotype mapping approach was adaptable, the next experiment was to prove that the complete GP system could be adapted to work for a different domain with minimal effort. The 'Complete Software Programs' domain was chosen this time, which encompasses the 'Software Statements and Structures' domain considered in the previous experiment.

Specifically, the adaptations were performed to allow the GP system in this research to evolve complete 'Sorting' solutions. Experiments showed that the only other system component that needed changing, in addition to the Genotype-Phenotype Mapping, was the Fitness test. Xhemali et al. (2010-c) discusses the experiment and the results obtained in detail.

Test 15, section 5.1

ATM has been using Windows Operating Systems (OS) since its establishment and will continue to use these OSs for the foreseeable future. The current OS used at the premises is Windows XP (Home Edition and Professional). The company has no immediate plans to change to the most recent Windows OS – Windows 7 – however it was decided to test each part of the prototype developed in this research on Windows 7 nonetheless, to ensure that, should ATM wish to upgrade to this OS in the near future, the prototype would still operate successfully.

The experiments showed that the CRAWLER, CLASSIFIER and EXTRACTOR exhibit the same behaviour and achieve the same results in both XP Professional and Windows 7.

Note that, the experiments were carried out on the 32-bit version of Windows 7. Further testing needs to be done to establish the compatibility of the prototype with the 64-bit version of Windows 7.

5.5 EVALUATION OF AIMS AND OBJECTIVES

The aims and objectives were created to define what was expected from this research and the tasks that were needed to realise the outcomes. The aim of this research study, as specified in Chapter 1, was to "develop a prototype system, which will automate the retrieval and extraction of training information from the web into the sponsoring company's database, guaranteeing an always up-to-date collection of training information, whilst keeping the user involvement to a minimum". Eight objectives were defined to achieve this aim:

Objective 1:	Review related work in the field of WIR and WIE.
Objective 2:	Analyse ATM's business processes and project requirements.
Objective 3:	Model database to store the results from each stage of the system.

Objective 4:	Automate the retrieval of web pages from the Web.
Objective 5:	Automate the classification of the retrieved pages into *relevant* and *irrelevant* categories.
Objective 6:	Automate the extraction of training course attributes from the relevant web pages into the database.
Objective 7:	Automate system execution to run at scheduled times.
Objective 8:	Critically evaluate the project, reviewing the success and achievements and recommend further improvements.

Objectives 1 and 2 involved researching not only the different technologies related to WIR and WIE, but also the sponsoring company. Time was spent to understand the business processes at ATM and the different procedures the advisors followed to locate course information. The existing tools and techniques used at ATM to extract the located information were also investigated. All this helped to recognise the gaps within the company and the research community. The knowledge acquired from this stage of the project was detailed in chapters 1, 2 and 3.

The information gained from the first two objectives, helped to refine the project requirements and to create the data storage for the research results (Objective 3). A detailed view of the database created was described in section 4.2.

Objectives 4, 5 and 6 were accomplished through the design and development of the different components of the prototype i.e. the CRAWLER, TRAINER, INDEXER, CLASSIFIER and EXTRACTOR. Chapters 4 and 5 discussed the work involved in the development of these components and the experimental results obtained.

The preferred approach for this research has always been the ‘Do-it-yourself’ approach, as it guarantees better control over systems and results. However, in relation to the automated system execution (Objective 7), the automated system is only needed to execute the program at scheduled times. There are no additional functionalities required from it, hence it does not need to be personalised in any way. Therefore, an existing software package was chosen to achieve this objective. The scheduler chosen is the System Scheduler Professional (SSP) (Splinterware Software Solutions, 2010). The SSP is a freeware scheduling tool, which is compatible with a range of Microsoft Windows platforms such as 2000, XP, 2003, Vista, 7 etc. It can also execute sequences of scheduled tasks, whilst the user is logged off. The SSP has all that is required for this research.

The above shows that the research objectives for this research have been satisfied. The rest of this chapter and chapter 6 continue to evaluate other important areas of the research, thus achieving Objective 8.

5.6 EVALUATION OF DATA SAMPLES

The data samples chosen for the experiments in this research consist of websites, web pages and keywords extracted from each web page. The websites chosen to test each part of the system were either recommended by the advisors at ATM, or were chosen to challenge the different components of the system with the diversity of features offered.

For example, some websites (underoak.co.uk, coursesplus.co.uk etc.) were chosen to test the

CRAWLER's ability to manage large volumes of data in comparison with other crawling systems. Other websites (reedlearning.co.uk, associatedtraining.co.uk etc.) were chosen as they contain a variety of file types such as: .HTML, .PHP, .ASPX, .PDF, .DOC, .GIF, .JPEG, .JS, .CSS etc. The CRAWLER proved successful in correctly identifying all the above. Image files (.GIF, .JPEG etc.) were disregarded in this version of the system.

The data sample prepared for the CLASSIFIER consisted of the web pages retrieved by the CRAWLER during the first stage of the system. Three differently sized data samples were created in order to test the CLASSIFIER's ability to manage different volumes of data. These same data samples were analysed by two other classifiers, thus each sample had to be restructured to suit the format of the additional classifiers, which was time consuming as explained in section 5.4.2. Each data sample was separated into a *training* and a *test* set. The training set was created by manually selecting random positive (*relevant*) and negative (*irrelevant*) examples (URLs) from the list of URLs retrieved by the CRAWLER. The test set was created automatically by the system using the remaining URLs in the above list. Section 6.2.2 explains the user involvement needed (at ATM) for the preparation of these sets.

The CLASSIFIER, as explained in section 4.3.4, categorises the web pages retrieved by the CRAWLER into *relevant* or *irrelevant*. The *irrelevant* pages are moved to a rejected list at this stage, thus the data sample considered by the final component of the system – the EXTRACTOR – consists of only the list of URLs categorised as *relevant* by the CLASSIFIER. Note that the existence of the CRAWLER and the CLASSIFIER is significant to the overall system, as, aside from locating and retrieving potential web pages from the Web, they also help towards providing a much cleaner data sample to the EXTRACTOR, in order to aid this complex task.

The data samples used in this research were applied to a variety of experiments and tested from different angles and different solutions (section 5.1), thus it is believed that they were appropriate for validating the competency of each part of the system.

5.7 EVALUATION OF APPROACH AND RESULTS

The prototype developed in this research consists of three main components: the CRAWLER, CLASSIFIER and EXTRACTOR and two sub-components: TRAINER and INDEXER, which aid the classification process.

The CRAWLER was developed to manage an unlimited number of websites in order to find as much course information as possible. External links are also recognised and crawled to aid the unlimited nature of the system. Sections 2.2.1 and 4.3.1 detailed the specific features of the CRAWLER. An additional functionality that makes the CRAWLER even more useful to this research, is the ability to reject web pages based on their URLs. A list of 'rejectable' URL keywords was identified through the use of the NB approach used for the CLASSIFIER. The CRAWLER eliminates all web pages that contain any of these keywords in their URLs. Section 5.4.1 showed the large number of *irrelevant* web pages eliminated this way.

The CRAWLER is scheduled to run periodically in the background, collecting new web pages and ensuring that existing links are still live. The CRAWLER achieved slower execution times than the two chosen commercial crawlers: WLV and LCP. However, the CRAWLER in this research needs no human involvement, thus execution time optimisation is not significant at this stage. Further work can be done to convert the CRAWLER from PHP to VB.NET, which was proven to achieve a 35% faster execution than PHP for the CLASSIFIER.

In relation to the CLASSIFIER, the NB approach was chosen, as an initial analysis of websites showed conditional independence amongst the features used. Despite later experiments showing a conditional dependence amongst some features, such as: price, date and location, the NB approach was tested on a number of web pages and proved very successful. Additionally, NB classifiers are simple to interpret, making it easier to check their performance after execution. This approach is also a practical solution, appropriate for the hardware specification available at ATM.

The standard NB approach in this research was improved to calculate the believed probability of features in each category. This additional step was added to manage situations when there is no information about the data, particularly during early stages of the execution of the CLASSIFIER. Furthermore, the CLASSIFIER was developed to benefit from some important web structures such as the page TITLE, Link and META Tag information.

The results of the CLASSIFIER are assisted by the TRAINER and the INDEXER. Processes like: the cleaning of the HTML code, the removal of stopwords and other unwanted parts of web pages and feature stemming contributed to a cleaner data sample for the CLASSIFIER.

The DTC and NNC were chosen for comparison, because they are powerful tools, which can quickly and robustly discover patterns within the data. They are also compatible with different operating systems, including Linux and Windows. The NNC, however, was deemed unsuitable for this project for reasons explained in section 5.4.2.

Decision Tree models are generally believed to perform better than NB Networks. However, the results in this research show that the CLASSIFIER achieved the highest Accuracy, Precision and F-Measure values (section 5.4.2), which is an important achievement for the application of the NB approach to attribute classifications.

The final part of the research related to the automatic extraction of training course information such as: title, price(s), dates(s) and location(s). Genetic Programming was the approach chosen for this part, because GP principles help to automate the generation of solutions, whilst aiming to evolve optimal solutions. For this research, GP was used to evolve REs for each particular web page under investigation.

This part of the research is associated with a number of innovative ideas, as summarised in section 6.2.1. The results obtained from this part of the system show potential, especially those from the extraction of course names, which achieve an accuracy of over 94%. The extraction of prices, dates and locations achieves moderately lower results, however steps have already been taken to improve these outcomes (see section 6.3).

Overall, this project has provided many contributions, not only to academia but also to the sponsoring company. Chapter 6 discusses these contributions and the implications of this research to the sponsoring company and the wider industry. The following section discusses and evaluates the deployment of the prototype at ATM.

5.7.1 DEPLOYMENT

The system that will enable the results of this research to be viewed by ATM's advisors is being developed by a third party. The system will include two components:

1. An application that will make the results of this research available to ATM through the CRM application.
2. A separate website, from which the advisors can view and manipulate training course information without the need to log into CRM first.

The developer has promised to deliver the system by the end of July 2010, which means that the deployment of the prototype built in this research will happen at that time as well.

Although the users at ATM have not yet been able to use the results obtained by this prototype, ATM has been kept informed of the achievements and limitations of every stage of the project throughout the EngD research period. The quality of the websites discovered has also been discussed and approved by the advisors.

The following discusses the different methods used to evaluate the system's worth to the sponsoring company. Due to the fact that the deployment of the prototype has not yet happened, the system's benefits to ATM are evaluated in terms of potential benefits rather than precise and measured findings.

There are currently two advisors at ATM who are responsible for searching for training course information on the Web and collecting this information for their clients. Unstructured and semi-structured interviews (discussed in chapter 3) were employed to discover the advisors' opinions on the functionalities and results of the new system and how these would affect their day-to-day responsibilities.

During the interviews, the advisors explained that, particularly during the past six months, they have experienced a big increase in the number of clients requesting courses. However, they found themselves incapable of managing all these clients' requests and having to turn them away, because each course involved several hours of web searching, which left little time to accommodate all the different clients. ATM has since had to limit the courses it provides to only the following:

- ADR – The carriage of heavy and dangerous goods
- Driver CPC Training – Certificate of Professional Competence
- SIA – Security Industry Authority (e.g. door supervision courses)
- Forklift Truck Training
- Plastering
- Tiling

Other areas are only considered if the advisors are not busy. This is damaging to the business, as ATM is a small company and cannot afford to turn clients away.

Due to the current limitation in training course variety and the limited time available for locating and extracting training courses, the advisors also use a limited number of websites. For example, one of the advisors admitted to only using one website – www.hotcourses.com –

for the clients' requests. Collectively, the advisors currently use only six websites[10] (although they do not all agree on the quality of each website). This shows that the advisors do not compare many providers for each course, which means that they may be losing out on better deals.

Another difficulty mentioned by the advisors regarding web searching, was that finding relevant information was difficult even when the advisors went directly to a provider's website known to offer the information needed. This is because some websites are too large and difficult to navigate and they categorise web pages very differently from one another. For example, the advisor mentioned that it took over thirty minutes to find the page with information on 'NVQ Advice and Guidance' from http://www.i-l-m.com, as this is categorised under the 'Caring' section in this website, while other websites categorise it under 'NVQ Courses' (http://www.adviceandguidance.org.uk), 'Qualifications' (http://www.edexcel.com), etc.

The list of the websites discovered by the CRAWLER, displayed on Table 5.4, was given to the advisors at ATM for them to check if they had heard of or used any of these websites before. Out of the 24 websites shown to the advisors, only three had been encountered during previous web searches. The advisors confirmed that the remaining 21 websites included training information that would be of interest to ATM. This means that, even for this small sample, the system increased the number of potential training websites from 6 to 27, which is a 350% increase.

Section 6.2.2 provides further information on the implications and benefits of this research to the sponsoring company. The results of each part of the prototype built (discussed previously in this chapter) are also testament to the success of this research.

5.8 SUMMARY

This chapter focused on the results obtained from each part of the prototype developed in this research. The data corpus and the testing metrics used to measure the performance of the different experiments carried out were also discussed. The different components of the developed prototype were compared to other existing approaches, in order to determine their strengths and weaknesses. The experimental results and the evaluation of the different aspects of the research demonstrate that WIE systems such as the one presented in this thesis, provide a solution that companies like ATM can easily adopt into their existing systems and work practices. Furthermore, the techniques developed in this research proved successful in meeting the initial aims and objectives for the research, with the CRAWLER and CLASSIFIER exceeding expectations and the EXTRACTOR achieving encouraging results.

[10] If no relevant information is found from any of these websites, the advisors then call the providers to enquire further.

6 RECOMMENDATIONS AND CONCLUSIONS

6.1 INTRODUCTION

This chapter concludes this thesis by discussing the implications of the research on both the industrial supervisor and the wider industry. A critical evaluation of the overall research is also provided, together with the contribution offered by this research and recommendations for further improvements.

6.2 CRITICAL EVALUATION OF THE RESEARCH

The aim of this research was to investigate the automation of web extraction for training course information. This is a very challenging task, considering the complexity of capturing information from a substantially sized, inconsistent and ever changing source such as the Web. The above, coupled with the limited timeframe available for the EngD project, mean that there are certain limitations associated with this research, in addition to the many contributions and benefits to the sponsoring company and the research community, as discussed below.

6.2.1 CONTRIBUTION TO EXISTING THEORY AND PRACTICE

This research has proven successful not only towards helping the sponsoring company, but also towards providing innovative and constructive ideas to the research community. Some of these ideas have been published in respectable, international conferences and journals (see Appendix F to J).

From a technical point of view, contributions of this research include:

- **CRAWLER Filtering**: Unlike standard crawlers, the CRAWLER in this research is responsible not only for locating and retrieving the web addresses of various web pages (and everything else that this task entails), but also for the exclusion of a large number of *irrelevant* web pages from the process. This was achieved through the use of the NB approach as explained in section 4.3.1. This additional functionality allows the CRAWLER to filter out a large number of *irrelevant* web pages before the CLASSIFIER is even underway. Section 5.4.1 lists the benefits of this approach.

- **CLASSIFIER Training**: Many researchers have expressed their confidence in the simplicity and merit of Naïve Bayes in the classification of attribute data. The CLASSIFIER in this research upholds this confidence by exceeding expectations in the classification of web pages. A contribution from this part of the system, however, comes from the fact that the CLASSIFIER outperforms rival techniques despite it being trained with fewer web pages than any other classification system that has been encountered during the research (Xhemali et al., 2009). The normal approach for classification systems is to train the classifier with 70-80% of the data and test it on the remaining 20-30% of the data. In this research, the reverse is the case; the CLASSIFIER is trained with around 20% of the data and tested on the remaining 80% of the data. The fact that the CLASSIFIER succeeds even with such minimal training, shows the great potential of this approach.

- **Genotype-Phenotype Mapping**: This research has presented a novel approach to mapping genotypes to phenotypes during the genetic evolution of REs, involving carefully written XML rules being fed to the system as a separate file. The existence of these rules as a separate file means that the file can be replaced to work on a different domain, without disrupting the rest of the GP system (explained in section 4.4.2.4). This was proven when the genotype-phenotype mapping approach was adapted to manage the evolution of Software Statements and Structures and Complete Software Programs with minimal effort (Xhemali et al., 2010-a; Xhemali et al., 2010-c). Furthermore, the use of XML denotes improved readability, compatibility with many programming languages, portability and extendibility (XML is not restricted to a limited set of keywords defined by the proprietary vendors, which aids the process of creating rules of different levels of complexity).

- **Initial Population Generation**: In this research, a novel combination of two known approaches was used. Specifically, the approach of randomly generating the individuals of the initial population and the approach of seeding the initial population with known solutions, were combined whereby, the first gene of the genotype is seeded with existing solutions in that category (e.g. if the system needs to evolve REs to extract course titles, then the first genes of the top ten RE genotypes, that were successfully evolved to extract titles, are used as the first genes of the initial population), whereas the remaining genes of each genotype are generated randomly. This approach allows for some initial knowledge to be injected in the evolutionary process, which means faster execution and little compromising of the search space.

- **Learning after each Genetic Generation**: This is another novel approach used in this research, specifically during the evolution of REs for the extraction of training course titles. As discussed in chapter 4, the NB approach was used to aid the fitness scoring of each evolved title RE. This in itself is not a novel idea, however the way it is applied in this research makes it innovative and successful. Specifically, information obtained by the NB part of the system, is passed from generation to generation, independently from the genetic evolution process, which means that each generation has additional knowledge to help it evolve good offspring faster. Chapter 5 showed the experiments carried out to prove the success of the above approach.

From a more practical point of view, this research offers the following contributions:

- **Domain**: To the best of our knowledge, no other work has concentrated on providing automated WIE solutions for the Training Courses Domain. Furthermore, as mentioned in section 2.4.6, very few researchers have investigated evolving REs for the purpose of WIE, which makes this research very beneficial.

- **Sponsoring Company**: Unlike conventional PhD research, this research was designed to meet the needs of an existing company, by managing a genuine commercial problem in a real business environment. All the achievements in this research (summarised in chapter 5), have contributed towards improving the business processes at the sponsoring company and having an impact on the wider industry. The following subsections focus on the impacts on the sponsor and the wider industry.

6.2.2 IMPLICATIONS/IMPACT ON THE SPONSOR

ATM's main objective for sponsoring this research was to reduce the time wasted at the company, searching for and manually dealing with online training course information. This EngD project has made a positive contribution to the sponsoring company, as it has provided not only a solution for finding and storing the relevant websites that may contain training course information, but also a solution that permeates each website and discovers specific information related to the different courses advertised.

It was never the intention of this research to replace the work and expertise of the advisors at ATM, instead, the final system was intended to act as a helpful guide to the advisors, leading them in the right direction towards finding the course information required. Section 1.4 showed the hectic and inefficient ways of dealing with online courses at ATM, prior to this research, where advisors had to actively search the Web each time training course options were needed. The system developed in this research ensures that, the advisors' first point of contact is now an always-up-to-date database. The database provides the initial information related to a course, such as the course title, price, date and location, which is enough for the advisors to determine whether or not that course could be suitable for a specific client. The database also provides the exact web location that contains more information on each course.

The system developed in this research makes a positive contribution to ATM even for cases when the system fails to extract any information from a web page, or only manages to extract partial information from it. This is because the system has still alerted the advisors to the existence of such web pages and the strong possibility of there being training course information in these pages. This corresponds to 2/3^{rds}, or ≈66% of the work that advisors would have had to accomplish themselves prior to this research. The additional training providers identified by the system are also useful for ensuring that the best deals, in terms of course quality and price, are made by ATM's advisors, which in turn can translate to potential financial benefits for the company.

Another advantage of this research for ATM is that the developed system requires minimal user involvement. In fact, the only involvement required is during the training stages of the CLASSIFIER and the GP part of the system that deals with the extraction of course titles. This is because in both cases, the Naïve Bayes approach is used to aid the decision making process and this approach requires training for it to work efficiently.

The training to be provided requires no knowledge of any kind from the user. The user is simply required to select a number of web pages, from a list of pages discovered by the CRAWLER, and use their human judgement to separate these into pages that do or do not contain training course information. A pre-prepared script can then be executed to prepare the system and the database for the training process, using the above information.

The system has also been designed to allow for growth at ATM. If the areas of interest to the business were expanded, few changes would be required as follows:

- New seed URLs for the CRAWLER
- Re-training of the CLASSIFIER with web pages from the new domain
- Additional XML rules to manage the new areas of interest
- Potential changes to the Fitness Test, depending on the extent of dissimilarities between the different domains.

6.2.3 IMPLICATIONS/IMPACT ON THE WIDER INDUSTRY

Organisations are cautious when adopting new systems. This can be attributed to the significant change required by some off-the-shelf and most bespoke systems in the organisations' infrastructure and existing technologies. Considerable efforts have been made in this research to integrate the system developed with existing systems and work practices at ATM, without disrupting any of the processes already in place. This would also be the case for any other organisation; in fact the only part of the system that would need to be amended, for it to integrate with existing systems and work practices in other organisations, would be the server details where the database is stored, so the system can connect to it.

The wider industry would benefit from the implementation of the system in this research in ways similar to those of the industrial sponsor. Other training brokerages could use this system immediately without any changes required. Other organisations, which are interested in extracting information from the Web, whatever this information may be, would be able to use the system after few changes, as listed in section 6.2.2.

An advantage of the system in this research is that it is created from three separate components: CRAWLER, CLASSIFIER and EXTRACTOR, which can be treated as stand-alone applications if necessary, and as such could be helpful to an even larger number of organisations. For example, the CRAWLER can be used by anyone as long as the system is seeded with a few initial web pages from the domain of interest; similarly, the CLASSIFIER can be used by any industry as long as it is trained with a set of positive and negative examples from the domain of interest; the GP part of the system can be used for a variety of domains. Xhemali et al. (2010-a; 2010-c) proved that this is possible even for domains that are radically different from that of REs, such as Software Statements and Structures.

An initial market research study conducted by ATM, with assistance from Atos Origin Ltd., has indicated that there is great demand for systems like the one developed in this research. This is because, people and organisations will always require information, thus there will always be need for tools to assist them to efficiently and effectively locate relevant sources of knowledge. The ability to take this further, and extract specific pieces of information into a database, makes this project even more appealing to the wider industry.

6.3 RECOMMENDATIONS FOR INDUSTRY/FURTHER RESEARCH

In spite of the extensive research conducted over the past four years, there remain some issues that require further research. This section discusses ideas for further improvements to the prototype built. Some of these ideas have been mentioned throughout the thesis.

- The current prototype does not use parallel processing when crawling. This would be very beneficial especially when checking if web pages are live or lead to dead ends. This could improve efficiency.

- PDF, DOC and XLS files are currently saved into the database, however they are not visited during the classification or extraction process. Some work was achieved on this area, however this was not sufficient enough to be incorporated into this research. Further research and development could transform this partial solution into a useful addition to the system.

- This is a very space-intensive system, as it stores a large number of web pages, features and course attributes. Database optimisation is, therefore, an area that needs to be researched further. The current concern is that, as with many off-the-shelf products, the system will become slower, the fuller the database becomes. Particularly, CIE_Rejected_Links can get out of control, as there are too many *irrelevant* web pages on the Web. A solution to this could be to use the Decision Trees approach to determine whether or not entire branches in websites (i.e. a collection of web pages that derive from each other) are *irrelevant*. The benefit of this solution to the system in this research would be that the CIE_Rejected_Links table would only store the 'root' URL of each branch, rather than each web page separately. Furthermore, the crawling process would be much faster, because it would not attempt to crawl any of these 'rejected' website branches.

- Due to the complexity of finding and capturing information from web sources and the limited time frame available for the EngD project, this research has concentrated on extracting information from HTML pages only. Xhemali et al. (2007) explained however, that there are different types of web pages to be considered such as: the Deep Web, XML pages etc. Furthermore, there are web pages that require a user to login before any of the information on that page can be displayed. The latter could be solved by inviting the user to log into the website in the very first instance of reaching this site. The system will then learn from this experience and will not require any more input from the user during future visits to this page.

- Section 4.4.2.4 discussed the novel approach of using XML rules to guide the genotype to phenotype mapping process. Recent work has highlighted the potential that may arise from updating the XML rules to follow a more grammatical approach. The rules would remain as a separate file, however a special grammar would be created, which would guide the mapping process. The potential of this technique comes from the fact that a Repairing function would no longer be required, as the grammatical approach would ensure that only syntactically correct phenotypes were produced. This would further simplify the customisation process of the system for new domains. This idea was formulated quite late on in the research, thus it was not possible to implement it, however sample grammatical rules have been created for both the RE domain and the Software Statements and Structures domain, to explain the idea further. These can be found in Appendix D.

- The XML rules that guide the mapping of genotypes to the corresponding phenotypes (REs) in the current prototype have been created manually after a thorough analysis of a large sample of web pages (as discussed in section 4.4.2.4). This number of XML rules is large enough to be successful on a large number of websites (as shown by the results obtained), however it still does not cover every possible RE combination that may be successful in extracting information from every website on the Web. An infinite number of rules could be added to the XML rules to widen and diverge the REs to be created, however this would require the involvement of a RE expert in creating these rules in the first place. One idea that may improve this system further is to attempt to evolve these rules. With each genetic generation, the system 'learns' information that can be used to evolve stronger and better rules. This approach would eliminate the need for an expert's involvement.

Another idea is to involve the user in creating these rules. Siau, Hinde and Stone (Siau et al., 2010) are currently working towards implementing this functionality, in collaboration with ATM. This would involve presenting the user with the web pages for which the current system is unsuccessful and asking them to highlight the areas on each page that are of interest to them. The system would then analyse these selections and convert them into rules, which will then be added to the XML rule file. This means that, the system will then 'know' how to get information from these websites (in the future) and any other websites that may follow similar presentation styles. This latter idea represents exciting news for ATM, because, the combination of the research presented in this thesis and the solution being developed by Siau et al. (2010), would create a powerful solution for ATM, one that would be able to extract a large amount of course information from the Web into the company's database.

6.4 OVERALL CONCLUSIONS

This thesis was designed to give a detailed walkthrough of the research and development carried out in the last four years. The work involved ranged from thorough investigation and research into the world of WIR/WIE and the business processes in the sponsoring company (Concept Development), to the different parts of the software life cycle (System Development), to the evaluation of the different aspects related to the project (System Evaluation). Each stage was discussed systematically to provide information regarding the contribution of this research and the amount of work that has been invested in it.

This research aimed to build a prototype to automate the retrieval, classification and extraction of training course information such as course titles, prices, dates and locations from the Web. The investigation into ATM's business processes confirmed the need for change at the sponsoring company. Research into existing work in the WIR and WIE fields also validated the industry's need for more research in this area, particularly in the area of WIE. The final components of the prototype include the CRAWLER, CLASSIFIER and EXTRACTOR, with the CLASSIFIER also including a TRAINER and an INDEXER. The TRAINER and the INDEXER are essential elements for the success of the CLASSIFIER.

In this project, the 'write-your-own' code approach was favoured to using 'off-the-shelf' software packages. The latter may be an easier approach to programming, however it limits the control over the system, particularly if errors occur or certain functionalities need to be changed or enhanced. The only 'off-the-shelf' system that was used in this research was to schedule the system execution, as this feature did not need to be enhanced in any way.

The review of the contributions shows that the research objectives have been satisfied. The results obtained from each component of the system demonstrate the potential of this research not only for the sponsoring company but also for the wider industry.

In summary, this research has: succeeded in identifying areas to consider when designing systems similar to the one presented in this thesis; designed, developed and tested a complex system that locates, filters and captures training course information from the web into the sponsoring company's database; highlighted the benefits of each of the separate components and the system as a whole to the sponsoring company and the wider industry, whilst offering novel ideas and other contributions to the research community.

7 REFERENCES

1. Aho, A.V. & Ullman, J.D. (1979). *Principles of Compiler and Design.* Addison Wesley, ISBN 0-201-00022-9.
2. Addin, O., Sapuan, S. M., Mahdi, E., & Othman, M. (2007). A Naive-Bayes classifier for damage detection in engineering materials. *Materials and Design*, pp. 2379-2386.
3. Ahmad, A.R., Khalid, M. & Yusof, R. (2002). Kernel methods and support vector machines for handwriting recognition. *Research and Development*, pp. 309-312.
4. Arfken, G. (1985). Scalar or Dot Product. *Mathematical Methods for Physicists, 3rd Edition.* Academic Press, pp. 13-18.
5. Armitage, A. (2007). Mutual Research Designs: Redefining Mixed Methods Research Design, *Proceedings of the Annual Conference of the British Educational Research Association, London.*
6. Asirvatham, A. P., & Ravi, K. K. (2001). Web Page Classification based on Document Structure. *Hyderabad, India: International Institute of Information Technology*.
7. Atkinson-Abutridy, J., Mellish, C., Aitken, S. (2004). Combining information extraction with genetic algorithms for text mining. *IEEE Intelligent Systems.* **19**(3), pp. 22-30.
8. Attardi, G., Gulli, A., & Sebastiani, F. (1999). Automatic Web Page Categorization by Link and Context Analysis. *Proceedings of THAI.* European Symposium of Telematics.
9. Aubrecht, P., Zakova, M. & Kouba, Z. (2005). Ontology Transformation Using Generalised Formalism. *The 2005 Knowledge Conference*, Slovakia, pp. 154-161.
10. Avison, D.E. & Fitzgerald, G. (2003). *Information Systems Development: Methodologies, Techniques and Tools,* 3rd Edition, McGraw Hill, Maidenhead.
11. Banko, M., Cafarella, M., Soderland, S., Broadhead, Etzioni, O. (2007). Open Information Extraction from the Web. *IJCAI-07.* pp. 2670-2676.
12. Banzhaf, W. (1994). Genotype-Phenotype-Mapping and Neutral Variation – A case study in Genetic Programming. *Proceedings of the International Conference on Evolutionary Computation*. Springer-Verlag, pp. 322-332.
13. Banzhaf, W. (2006). Genotype-Phenotype Mapping and Neutral Variation – A Case Study in Genetic Programming. *Lecture Notes in Computer Science,* Springer Berlin, **866**, pp. 322-332.
14. Bao, J., Cao, Y., Tavanapong, W. & Honavar, V. (2004). Integration of Domain-Specific and Domain-Independent Ontologies for Colonoscopy Video Database Annotation. *Proceedings of the International Conference on Information and Knowledge Engineering,* pp. 82-88.
15. Barrero, D., Camacho, D. & R-Moreno, M. (2009). Automatic Web Data Extraction Based on Genetic Algorithms and Regular Expressions. *Data Mining and Multi-agent Integration*. ISBN 978-1-4419-0523-9, Springer-Verlag US, pp. 143.

16. Basu, A., Walters, C., & Shepherd, M. (2003). Support vector machines for text categorization. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences.*

17. Baykan, E., Hezinger, M. & Weber, I. (2008). Web page language identification based on URLs. *Proceedings of the VLDB Endowmnet,* **1**(1), pp. 176-187.

18. Berry, M. & Linoff, G. (1997). *Data Mining Techniques*. New York: John Wiley & Sons.

19. Blaschke, C. (2002). Automatic ontology construction from the literature. *Medline,* Universal Academy Press.

20. Boger, Z. & Guterman, H. (1997). Knowledge extraction from artificial neural networks models. *Proceedings of the IEEE International Conference on Systems Man and Cybernetics,* Orlando, Florida, pp. 3030-3035.

21. Booch, G. (1986). Object-Oriented Development. *IEEE Transactions on Software Engineering,* Vol. SE-12.

22. Booch, G., Maksimchuk, R., Engle, M., Young, B., Conallen, J. & Houston, K. (2007). *Object-oriented analysis and design with applications, 3rd Edition*. Addison-Wesley Professional.

23. Braga, P., Oliveira, A. & Meira, S. (2008). A GA-based feature selection and parameters optimization for support vector regression applied to software effort estimation. *Proceedings of the 2008 ACM Symposium on Applied Computing,* pp. 1788-1792.

24. Brameier, M. & Banzhaf, W. (2007). *Linear Genetic Programming.* In Genetic and Evolutionary Computation Series, Series Editors: Goldberg, D.E. & Koza, J.R.

25. Brin, S. & Page, L. (1998). The anatomy of a large scale hyper textual web search engine. *Proceedings of the 7th World Wide Web Conference.*

26. Burstein, F. (2002). System Development in Information Systems Research. *Research Methods for Students, academics and Professionals: Information Management and Systems,* 2nd Edition, pp. 147-158.

27. Buttler, D., Liu, L. & Pu, C. (2001). A fully automated object extraction system for the world wide web. *Proceedings of the International Conference on Distributed Computing Systems,* **21**, pp. 361-370.

28. Cai, Y-D, Liu, X-J, Xu, X. & Zhou, G-P. (2001). Support Vector Machines for predicting protein structural class. *BMC Bioinformatics 2001,* **2**(3).

29. Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B. & Goncalves, M. (2003). Combining link-based and content-based methods for web document classification. *Proceedings of the 12th International Conference on Information and Knowledge Management,* pp. 394-401.

30. Calado, P., Cristo, M., Andre, M., Moura, G., Neto, B.R. & Ziviani, N. (2006). Link-based similarity measures for the classification of web documents. *Journal of the American Society for Information Science and Technology.* **57**(2), pp. 208-221.

31. Calvo, B., Larranaga, P. & Lozano, J. (2007). Learning Bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters,* **28**(16), pp. 2375-2384.

32. Ceci, M., & Malerba, D. (2003). Hierarchical Classification of HTML Documents with WebClassII, *Lecture Notes in Computer Science*, pp. 57-72.

33. Cetinkaya, A. (2007). Regular expression generation through grammatical evolution. *Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation* (London, UK). Workshop Session, pp. 2643-2646.

34. Chau, M., & Chen, H. (2007). A machine learning approach to web page filtering using content and structure analysis. *Decision Support Systems*, pp. 482-494.

35. Chen, G. & Bhattacharya, P. (2006). Function Dot Product Kernels for Support Vector Machine. *Proceedings of the 18th International Conference on Pattern Recognition*, **2**, pp. 614-617.

36. Cho, J., Garcia-Molina, H., & Page, L. (1998). Efficient Crawling through URL Ordering. *Journal of Computer Networks and ISDN Systems.* 161-172.

37. Christy, A. & Thambidurai, P. (2007). Intelligent Information Extraction with Soft Matching Rules and Knowledge Discovery using Genetic Algorithm for Text Mining. *International Conference on Computational Intelligence and Multimedia Applications*, **2**, pp. 141-145.

38. Clarke, M., Hinde, C.J., Whithall, M.S., Jackson, T.W., Phillips, I.W., Brown, S & Watson, R. (2009). Allocating Railway Platforms using a Genetic Algorithm. *Research and Development in Intelligent Systems XXVI*, Springer, pp. 421-434.

39. CMS (2005). Selecting a Development Approach. *Centers for Medicare and Medicaid Services, Office of Information Services*.

40. Collobert, R. & Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. *Proceedings of the 25th International Conference on Machine Learning*, pp. 160-167.

41. Conrad, E. (2007). Detecting Spam with Genetic Regular Expressions. *SANS Institute Reading Room*. Available online at: http://www.giac.org/certified_professionals/practicals/GCIA/00793.php

42. Cooper, H. (1998). Synthesizing Research: Third Edition – A Guide for Literature Reviews. *Applied Social Research Methods Series*, **2**(3).

43. Cormack, D. S. (1991). The research process. *Black Scientific: Oxford.*

44. Crescenzi, V., Mecca, G. & Merialdo, P. (2001). RoadRunner: Towards Automatic Data Extraction from Large Websites, *Proceedings of the 27th VLDB Conference.*

45. Crestani, F. (1993). An Adaptive Information Retrieval System Based on Neural Networks. *Proceedings of the International Workshop on Artificial Neural Networks: New Trends in Neural Computation*. 732-737.

46. Creswell, J.W. (2003). Research Design: Qualitative, Quantitative and Mixed Methods Approaches. *London: Sage*, 2nd Edition.

47. Dawson, R. (1997). *Relational Databases Design and Use. First Edition*. Loughborough University: Audio Visual Services.

48. De Kunder, M. (2006). How big is the World Wide Web right now? *Master's Thesis, Tilburg University, The Netherlands*

49. De Kunder, M. (2010). The Size of the World Wide Web. Available from: http://www.worldwidewebsize.com [Accessed on 24 May 2010]

50. Denis, F., Laurent, A., Gilleron, R., Tommasi, M. (2003). Text classification and co-training from positive and unlabeled examples. *Proceedings of ICML Workshop: The Continuum from Labeled to Unlabeled Data*, pp. 80-87.

51. Dubminsky, Y., Hazzan, O., Talby, D. & Keren, A. (2006). System analysis and design in a large-scale software project: The case of transition to agile development. *Proceedings of the 8th International Conference on Enterprise Information Systems,* Paphos, Cyprus.

52. Dumais, S. (1998). *Using SVMs for Text Categorization.* Microsoft.

53. Dyer, J. & Bentley, P. (2002). PLANTWORLD: Population Dynamics in Contrasting Environments. *In Cantu-Paz E., GECCO*, pp. 122-129.

54. Dzurec, L.C. & Abraham, I.L. (1993). The nature of inquiry: linking quantitative and qualitative research. *Advances in Nursing Science*, **16**(1), pp. 73-79.

55. Eikvil, L. (1999). Information Extraction from World Wide Web – A Survey. *Norwegian Computing Centre,* **945**.

56. Elsas, J., & Efron, M. (2000). HTML Tag Based Metrics for use in Web Page Type Classification. *University of North Carolina - ASIST Annual Meeting*.

57. Embley, D.W. (2004). Toward semantic understanding: an approach based on information extraction Ontologies. *Proceedings of the 15th Australasian Database Conference.* Vol. 27, pp. 3-12.

58. Embley, D.W., Tao, C. & Liddle, S.W. (2005). Automating the extraction of data from HTML tables with unknown structures. *Data & Knowledge Engineering.* **54**(1), pp. 3-28.

59. Estruch, V., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M.J. (2006). Web Categorisation Using Distance-Based Decision Trees. *Proceedings of the International Workshop on Automated Specification and Verification of Web Sites,* pp. 35-40.

60. Etzioni, O., Cafarella, M. & Downey, D. (2005). Web-Scale Information Extraction in KnowItAll (Preliminary Results). *WWW2005,* New York, U.S.A.

61. Feldman, S. (2004). The High Cost of not Finding Information, *KM World Magazine*.

62. Fletcher, P. (1995). The Role of Experiments in Computer Science. *Journal of Systems and Software.* **30**(1-2), pp. 161-163.

63. Fletcher, G.P & Hinde, C.J. (1994). Interpretation of Neural Networks as Boolean Transfer Functions. *Knowledge-Based Systems,* **7**(3), pp. 207-214.

64. Fletcher, G.P & Hinde, C.J. (1995). Using Neural Networks as a Tool for Constructing Rule Based Systems. *Knowledge-Based Systems*, **8**(4), pp. 183-189.

65. Fletcher, G.P & Hinde, C.J. (1996). Producing Evidence for the Hypotheses of Large Neural Networks. *Neurocomputing,* **10**, pp. 359-373.

66. Fogel, L.J., Owens, A.J. & Walsh, M.J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons Inc.

67. Friedl, J. (1997). *Mastering Regular Expressions*. O'Reilly & Associates (Jan 1997).

68. Friedl, J. (2006). *Mastering Regular Expressions*, Third Edition. O'Reilly & Associates (Aug 2006).

69. Gao, Y., Yuan, F. & Zhang, M. (2007). Data Extraction Based on Index Path in Web. *Proceedings of the 2nd International Workshop on Education Technology and Computer Science.* **3**, pp. 157-160.

70. Gerrand, P. (2007). Estimating Linguistic Diversity on the Internet: A Taxonomy to Avoid Pitfalls and Paradoxes. *Journal of Computer-Mediated Communication,* **12**(4).

71. Glass, R., Ramesh, V. & Vessey, I. (1994). An analysis of research in Computing disciplines. *Communications of the ACM,* **47**(6), pp. 89-94.

72. Gomez, M., Abasolo, C. & Plaza, E. (2001). Domain-Independent Ontologies for Cooperative Information Agents. *Lecture Notes in Computer Science,* pp. 118-129.

73. Google. (2009). http://googleblog.blogspot.com/2009/02/from-height-of-this-place.html [October, 2009]

74. Gordon, M., Fan, W. & Pathak, P. (2006). Adaptive web search: Evolving a program that finds information. *IEEE Intelligent Systems.* **21**(5), pp. 72-77.

75. Goyvaerts, J. (2009). RegexBuddy. http://www.regular-expressions.info/regexbuddy.html.

76. Greasly, C. (2006). Information Management. *Loughborough, UK: Apricot Training Management* [Internal Report].

77. Grosan, C. & Abraham, A. (2008). Evolving Computer Programs for Knowledge Discovery. *Social Science Research Network (SSRN).*

78. Grosky, W., & Deshpande, G. (2004). Web Page Retrieval by Structure. *International Journal of Information Theories & Applications.*

79. Hammer, J., Garcia-Molina, H., Cho, J., Aranha, R. & Crespo, A. (1997). Extracting Semistructured Information from the Web, *Proceedings of the Workshop on Management of Semi-structured Data*.

80. Harris, D. (2006). Open or Closed – That is the Question. *The 2006 Annual Conference of the British Educational Research Association, Warwick.*

81. Haykin, S. (2008). *Neural Networks and Learning Machines (3rd Edition),* London, Prentice Hall.

82. Hazzan, O., Dubinsky, Y., Eidelman, L., Sakhnini, V. and Teif, M. (2006). Qualitative research in Computer Science education. *Proceedings of SIGCSE 2006 – The 37th Technical Symposium on Computer Science Education, Houston, Texas,* pp. 408-412.

83. Heddad, A., Brameier, M. & MacCallum, R. (2004). Evolving Regular Expression-based Sequence Classifiers for Protein Nuclear Localisation. *In Applications of Evolutionary Computing*. Springer, Berlin, pp. 31-40.

84. Hinde, C.J., Fletcher, G.P., West, A.A. & Williams, D.J. (1997). Neural Networks. *ICL Systems Journal,* **11**(2), pp. 244-278.

85. Hodgett, R.A. (2003). The acceptance of Object-Oriented Development Methodologies in Australian Organisations and the Place of UML in Information System Programs. *Information Science*.

86. Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. *University of Michigan Press*.

87. Holscher, C. & Strube, G. (2000). Web Search Behaviour of Internet Experts and Newbies. *Proceedings of the 9th International World Wide Web Conference on Computer Networks*, Amsterdam, The Netherlands, pp. 337-346.

88. Hsu, P-H. (2007). Feature extraction of hyperspectral images using wavelet and matching pursuit. *ISPRS Journal of Photogrammetry and Remote Sensing*. Elsevier Science, Amsterdam, vol. 62 (2), 78-92.

89. Hu, W-C., Chang, K. & Ritter, G. (2000). WebClass: Web Document Classification Using Modified Decision Trees, *Proceedings of the 38th annual on Southeast regional conference*, pp. 262-263.

90. Hu, Y. & Xuan, Y. (2008). Research on Web Information Extraction Based on XML. *Proceedings of the 2nd International Conference on Genetic and Evolutionary Computing*. pp. 201-204.

91. Huang, C-L. & Wang, C-J. (2006). A GA-based feature selection and parameters optimizationfor support vector machines. *Expert Systems with Applications*, Elsevier Ltd., **31**(2), pp. 231-240.

92. Huang, J. & Shi, F. (2005). Support Vector Machines for Predicting Apoptosis Proteins Types. *ACTA BIOTHEORETICA*, **53**(1), pp. 39-47.

93. Huck, G., Fankhauser, P., Aberer, K. and Neuhold, E. (1998). Jedi: Extracting and Synthesizing Information from the Web, *CoopIS*.

94. IBM Research (2010). IBM Productivity Management Ideas. Available from: http://www.ibm.com/smarterplanet/us/en/productivity_management/ideas/ [Accessed on 16 March 2010]

95. Iida, T., Inaba, M., Mizoguchi, Y., Nagano, S. & Hattori, M. (2009). Preliminary Experiment of Ontology Maintenance for Ontology Metrics Formalization. *Proceedings of the 3rd Workshop on Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science*. **460**, pp. 61-71.

96. Indra, M., Rajaram, R. & Selvakuberan, K. (2007). Machine Learning Techniques for Automated Web Page Classification using URL Features. *Proceedings of the Internation Conference on Computational Intelligence and Multimedia Applications*. Vol. 2, pp. 116-120.

97. Jackson, D. (2005). Evolving Defence Strategies by Genetic Programming. *In Lecture Notes in Computer Science*. Springer Berlin, **3447**, pp. 281-290.

98. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the European Conference on Machine Learning*, pp. 137-142.

99. Joshi, P.M. & Liu, S. (2009). Web document text and images extraction using DOM analysis and natural language processing. *Proceedings of the 9th ACM Symposium on Document Engineering*, pp. 218-221.

100. Kamandi, A. & Habibi, J. (2008). A Framework for Classifying and Comparing Graphical Object Oriented Modelling Languages. *Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong*, vol. I.

101. Karlsson, F. & Hedstrom, K. (2009). Negotiating a Systems Development Method. *Information Systems Development,* Springer US, pp. 491-499.

102. Kauchak, D., Smarr, J. & Elkan, C. (2004). Sources of Success for Boosted Wrapper Induction. *The Journal of Machine Learning Research.* vol. 5, pp. 499-527.

103. Keaveney, D. & O'Riordan, C. (2009). Evolving Robust Strategies for an Abstract Real-time Strategy Game. *Proceedings of the 5th International Conference on Computational Intelligence and Games*, pp. 371-378.

104. Keller, R.E. & Banzhaf, W. (1996). Genetic Programming using Genotype-Phenotype Mapping from Linear Genomes into Linear Phenotypes. *Proceedings of the First Annual Conference on Genetic Programming*, California, pp. 116-122.

105. Khan, L. & Luo, F. (2002). Ontology construction for information selection. *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence,* Virginia, pp. 122-127.

106. Klank, U., Padoy, N., Feussner, H. and Navab, N. (2008). Automatic feature generation in endoscopic images. *International Journal of Computer Assisted Radiology and Surgery*. Springer, **3**, pp. 331-339.

107. Koza, J.R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. *MIT Press*.

108. Labsky, M., Svatek, V. & Nekvasil, M. (2008). Information Extraction based on Extraction Ontologies: Design, Deployment and Evaluation. *Proceedings of the 1st International and KI-08 Workshop on Ontology-Based Information Extraction Systems,* pp. 18-26.

109. Langdon, W. (2008). Evolving GeneChip correlation predictors on parallel graphics hardware. *Proceedings of WCCI,*Hong Kong.

110. Langdon, W., Poli, R., McPhee, N. & Koza, J.R. (2008). Genetic Programming: An Introduction and Tutorial with a Survey of Techniques and Applications. *In Studies in Computational Intelligence*. Springer, Berlin, **115**, pp. 927-1028.

111. Langdon, W., Roswell, J. & Harrison, A. (2009). Creating regular expressions as mRNA motifs with GP to predict human exon splitting. *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. Montreal, Canada, Poster Session, Track 3, pp. 1789-1790.

112. Langseth, H. & Nielsen, T. (2006). Classification using Hierarchical Naïve Bayes models, *Machine Learning,* **63**(2), pp. 135-159.

113. Lawrence, S. & Giles, C.L. (1999). Accessibility of information on the Web. *Nature,* vol. 400, pp. 107-109.

114. Lee, A. S., Liebenau, J. & DeGross, J. I. (1997). Information Systems and Qualitative Research. *Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research.* Philadelphia.

115. Li, Y., Chung, S. M., & Holt, J. D. (2008). Text document clustering based on frequent word meaning sequences. *Data & Knowledge Engineering*, pp. 381-404.

116. Link Checker Pro. [Available at: http://www.link-checker-pro.com/index.htm]

117. Liu, L., Han, W., Buttler, D, Pu, C. & Tang, W. (1999). An XML-based Wrapper Generator for Web Information Extraction. *Proceedings of ACM SIGMOD International Conference on Management of Data,* pp. 540-543.

118. Long, W., Yang, S., Min, Y. & Tong, S. (1997). Level-Oriented GA-based Test Generation of Logic Circuits. *Proceedings of the International Intelligent Processing Systems*, vol. 1, pp. 563-567.

119. Lonsdale, D.W., Embley, D.W. & Liddle, S.W. (2010). Ontologies for Multilingual Extraction. *WWW2010,* Raleigh, North Carolina.

120. Lu, Q. & Getoor, L. (2003). Link-based Text Classification. *Proceedings of the 20th International Conference on Machine Learning, Washington DC.*

121. Mahedero, J., Martinez, A. & Cano, P. (2005). Natural language processing of lyrics. *Proceedings of the 13th Annual ACM International Conference on Multimedia.* Singapore, pp. 475-478.

122. Manzano-Macho, D., Gomez-Perez, A. & Borrajo, D. (2008). Unsupervised and Domain Independent Ontology Learning: Combining Heterogeneous Sources of Evidence. *Proceedings of the 6th International Language Resources and Evaluation.*

123. Martin, T.P. & Azmi-Murad, M. (2005). An Incremental Algorithm to find Asymmetric World Similarities for Fuzzy Text Mining. *In Advances in Soft Computing.* **29**, pp. 838-847.

124. Martin, T.P. & Shen, Y. (2006). Improving Access to Multimedia using Multi-source Hierarchical Meta-data. *Lecture Notes in Computer Science,* **3877**, pp. 266-278.

125. Miniwatts Marketing Group, (2009). Internet Usage Statistics. Available from: http://www.internetworldstats.com/stats.htm [Accessed on 20 Apr 2010].

126. Mitchel, P. & El Kaliouby, R. (2003). Real Time Facial Expression Recognition in Video using Support Vector Machines. *Proceedings of ICMI'03,* Vancouver.

127. Mitchell, T. (1997). *Machine Learning*. McGraw Hill. ISBN 0-07-042807-7.

128. Mocian, H. (2009). Text Mining with Suffix Trees. *MSc Thesis, Imperial College London.*

129. Mohammadian, M. (2008). Classification of Data Based on a Fuzzy Logic System. *Proceedings of the 2008 International Conferences on Computational Intelligence for Modelling, Control and Automation; Intelligent Agenets, Web Technologies and Internet Commerce and Innovation in Software Engineering,* pp. 1288-1292.

130. Moore, J.P. (2000). Exploring and Exploiting Models of the Fitness Landscape: A Case against Evolutionary Optimization. *PhD Thesis,* University of Plymouth.

131. Myllymaki, J. (2002). Effective Web Data Extraction with Standard XML Technologies. *Computer Networks.* **39**(5), pp. 635-644.

132. Nasukawa, T. & Yi, J. (2003). Sentiment analysis: Capturing favourability using natural language processing. *Proceedings of the 2nd International Conference on Knowledge Capture,* pp. 70-77.

133. NIST, (last update: Apr 2010). Information Extraction Definitions [Homepage of National Institute of Standards and Technology], [Online]. Available from: http://www.itl.nist.gov/.

134. Ntoulas, A., Cho, J. & Olston, C. (2004). What's New on the Web? The Evolution of the Web from a Search Engine Perspective. *Proceedings of the 13th International World Wide Web Conference, New York.*

135. O'Neill, E.T, Lavoie, B.F. & Bennett, R. (2003). Trends in the Evolution of the Public Web. *D-Lib Magazine,* **9**(4), OCLC Office of Research.

136. Paolillo, J.C. (2005). Language Diversity on the Internet. *In Measuring Linguistic Diversity on the Internet, Paris,* UNESCO Report 142186, pp. 43-89.

137. Parr, C.L, Barikmo, I., Torheim, L.E., Ouattara, F., Kaloga, A. & Oshaug, A. (2002). Validation of the second version of a quantitative food-frequency questionnaire for use in Western Mali. *Public Health Nutrition,* **5**(6), pp. 769-782.

138. Peters, G. (2006). Some refinements of rough k-means clustering, *Pattern Recognition,* **39**(8), pp. 1481-1491.

139. Pierre, J. M. (2001). On the Automated Classification of Web Sites. Linköping Electronic Articles in Computer and Information Science, Vol. 6.

140. Poli, R., Langdon, W. & McPhee. (2008). N. *A Field Guide to Genetic Programming*. Published via http://lulu.com (With contributions from J. R. Koza).

141. Porter, M. (1980). An algorithm for suffix stripping, *Program,* **14**(3), pp. 130-137.

142. Provost, F. & Fawcett, T. & Kovahi, R. (1998). The Case against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the 15th International Conference on Machine Learning [ICML-59].*

143. Queensland University of Technology & Pennsylvania State University. (2007). Different Engines, Different Results.

144. Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research,* **4**, pp. 77-90.

145. Rafi, M., Qureshi, H. & Khatoon, H. (2009). Ontology Maintenance via Multi-agents. *Proceedings of the 5th International Joint Conference on INC, IMS and IDS,* pp.955-959.

146. Ramsin, R. (2006). The Engineering of an Object-Oriented Software Development Methodology. *PhD Thesis,* University of York, York, UK.

147. Ramsin, R. & Paige, R.F. (2008). Process-centred review of object oriented software development methodologies. *ACM Computing Surveys,* **40**(1).

148. Reis, D.C., Golgher, P.B., Silva, A.S. & Laender, A.F. (2004). Automatic Web News Extraction Using Tree Edit Distance. *Proceedings of the 13th International Conference on World Wide Web.* ACM Press, pp. 502-511.

149. REL Software. *Web Link Validator: The Broken Links Doctor.* [Available at: http://www.relsoftware.com/wlv/]

150. Rennie, J., Shih, L., Teevan, J. & Karger, D. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the 20th International Conference on Machine Learning,* Washington DC.

151. Riloff, E. & Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the National Conference on Artificial Intelligence,* pp. 474-479.

152. Romero, C. & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *International Journal of Expert Systems with Applications.* 33(1), pp. 135-146.

153. Rothlauf, F. (2006). Representations for Genetic and Evolutionary Algorithms. Springer-Verlag New York.

154. Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach.* London: Prentice Hall.

155. Ryan, A., Cohn, J., Lucey, S., Saragih, J., Lucey, P., Torre, F. & Rossi, A. (2010). Automated Facial Expression Recognition System. *IEEE Xplore.*

156. Sahuguet, A. & Azavant, F. (1999). Looking at the Web through XML Glasses. *Proceedings of the 4th IECIS International Conference on Cooperative Information Systems,* pp. 148.

157. Scacchi, W. (2001). Process Models in Software Engineering. *Encyclopedia of Software Engineering, 2nd Edition,* John Wiley and Sons Inc., New York.

158. Segaran, T. (2007). *Programming Collective Intelligence.* U.S.A: O'Reilly Media Inc.

159. Sells, C. (2002). RegexDesigner.NET. Available online at: http://www.sellsbrothers.com/Posts/Details/12425.

160. Shaeffer, E., Lampron, S., Krosnick, J., Tompson, T., Visser, P. & Janemann, M. (2004). A Comparison of Open vs. Closed Survey Questions for Valuing Environmental Goods. *The 2004 Annual Meeting of the American Association for Public Opinion Research, Phoenix, Arizona.*

161. Shah, M.A. & Deshpande, S.M. (2008). Web Page Classification Based on Document Structure without Negative Examples. *International Journal of Computer Science and Applications.* **1**(1), pp. 31-35.

162. Shamsfard, M. & Barforoush, A. (2004). Learning Ontologies from natural language texts. *International Journal of Human-Computer Studies.* **60**(1), pp. 17-63.

163. Sharp, H., Hovenden, F. & Woodman, M. (2005). Using metaphor to analyse qualitative data: Vulcans and humans in software development, *Empirical SE,* **10**(3), pp. 343-365.

164. Siau, N.Z., Hinde, C.H. & Stone, R.G. (2010). Teaching a Web Information Extraction System to Extend its Data Patterns. *Submitted to the 2010 UK Workshop on Computational Intelligence.*

165. Snajder, J., Basic, B.D., Petrovic, S. & Sikiric, I. (2008). Evolving new lexical association measures using genetic programming. *Proceedings of the Association for Computational Linguistics.* Ohio, U.S.A, pp. 181-184.

166. Snoussi, H., Magnin, L. & Nie, J. (2002). Heterogeneous web data extraction using ontology. *Proceedings of the AI-2002 Workshop on Business Agents and the Semantic Web,* Calgary.

167. Soderland, S. (1997). Learning to Extract Text-based Information from the World Wide Web, *Proceedings of Third International Conference on Knowledge Discovery and Data Mining,* American Association for Artificial Intelligence (AAAI).

168. Soderland, S., Fisher, D., Aseltine, J. & Lehnert, W. (1995). CRYSTAL: Inducing a Conceptual Dictionary, *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* pp. 1314-1321.

169. Spencer, J. (1983). Research with the human touch. *Researching Times,* pp. 24-27.

170. Splinterware Software Solutions (Last updated: 2010). Available at: http://www.splinterware.com/products/wincron.htm

171. Sun Microsystems. (2008). Lesson: Regular Expressions. Available online at: http://java.sun.com/docs/books/tutorial/essential/regex/index.html.

172. Syswerda, G. (1989). Uniform Crossover in Genetic Algorithms. *Proceedings of the 3rd International Conference on Genetic Algorithms,* San Francisco, U.S.A. Morgan Kaufmann, pp. 2-9.

173. Tal, B. (2003). Neural Network – Based System of Leading Indicators, *CIBC World Markets.*

174. Tang, W., Cen, G. & Cheng, J. (2010). Based on XML of Web Mining in Dynamic Dividing Level Instruction system. *Proceedings of the 2nd International Workshop on Education Technology and Computer Science.* **3**, pp. 468-472.

175. Tashakkori, A. & Teddlie, C. (2002). Handbook of Mixed Methods. *London: Sage.*

176. Tata, S., Hankins, R.A. & Patel, J.M. (2004). Practical Suffix Tree Construction. *Proceedings of the 30th VLDB Conference,* Canada, pp. 36-47.

177. Tedmori, S. (2008). Exploiting Email: Extracting Knowledge to Support Knowledge Sharing. *EngD Thesis, Loughborough University, Loughborough, UK.*

178. Tesfamariam, S. & Liu, Z. (2010). Earthquake induced damage classification for reinforced concrete buildings. *Structural Safety,* **32**(2), pp. 154-164.

179. Tethys Solutions. (2010). Automation Anywhere. Available from: http://www.automationanywhere.com/index.htm.

180. The UK Patent Office, (last updated: 2009), *Copyright Applies To ...* Available at: http://www.ipo.gov.uk/types/copy/c-applies.htm [10 Jan 2007].

181. Thompson, K. (1968). Programming techniques: Regular expression search algorithm. *Communications of the ACM,* **11**(6), pp. 419-422.

182. Thornton, C. (2000). *Truth from trash: how learning makes sense.* Massachusetts, MIT Press.

183. Toral, A. & Monachini, M. (2007). Simple-OWL: A Generative Lexicon Ontology for NLP and the Semantic Web. *Workshop of Cooperative Construction of Linguistic Knowledge Bases, 10th Congress of Italian Association for Artificial Intelligence.*

184. Touil, G. & Ahmed-Nacer, M. (2006). An Approach for Selecting Software Development Methodologies. *The 2006 International Conference on Software Engineering Applications, Dallas, U.S.A.*

185. Towell, G. & Shavlik, J. (1993). Extracting Refined Rules from Knowledge-Based Neural Networks. *Machine Learning,* **13**(1), 71-101.

186. Tu, J.V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology,* **50**(11), 1309-1310.

187. Turney, P. (1999). Learning to Extract Keyphrases from Text. *Technical Report ERB-1057,* Institute for Information Technology, National Research Council of Canada.

188. UC Berkeley (2006). Types of search tools. Retrieved: December, 2006 from http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/ToolsTables.html [no longer available]

189. United States District Court. (2000). eBay Inc. (Plaintiff). vs. Bidder's Edge (defendant) - Order Granting Preliminary Injunction. C-99-21200 RMW edn. California.

190. Van Couvering, E. (2007). Is relevance relevant? Market, science and war: Discourses of search engine quality. *Journal of Computer-Mediated Communication,* **12**(3).

191. VelocityScape Ltd.: http://www.velocityscape.com/Products/WebScraperPlus.aspx.

192. Wang C., Ding C., Meraz R., Holbrook S. (2006). PSoL: a positive sample only learning algorithm for finding non-coding RNA genes, *Bioinformatics*, **22**(21), pp. 2590-2596.

193. Wang, Y., Hodges, J. & Tang, B. (2003). Classification of Web Documents using a Naïve Bayes Method. *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence,* California, USA.

194. Ward, M. (1999). *Virtual Organisms: The Startling World of Artificial Life*. Pan Books.

195. Whigham, P.A. (1995). Grammatically-based Genetic Programming. *Workshop on Genetic Programming.*

196. Wisconsin-Madison (2009). Common Writing Assignments: Review of Literature. *The University of Wisconsin-Madison Writing Centre.* Available from: http://writing.wisc.edu/Handbook/ReviewofLiterature.html

197. Withall, M.S., Hinde, C.J. & Stone, R.G. (2008). An improved representation for evolving programs. *Journal of Genetic Programming and Evolvable Machines. Springer Netherlands*, **10**(1), pp. 37-70.

198. Witten, I.H. & Frank, E. (1999). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, *Morgan Kaufmann.*

199. Worth, D. & Greenough, C. (2005). *A Survey of Available Tools for Developing Quality Software Using Fortran 95,* CCLRC, Rutherford Appleton Laboratory.

200. Xhemali, D., Hinde, C.J. and Stone, R.G. (2007). Embarking on a Web Information Extraction Project. *UKCI Conference on Computational Intelligence,* London, UK.

201. Xhemali, D., Hinde, C.J. & Stone, R.G. (2009). Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. *International Journal of Computer Science Issues*, **4**(1), pp. 16-23.

202. Xhemali, D., Hinde, C.J. & Stone, R.G. (2010-a). Domain-Independent Genotype to Phenotype Mapping through XML Rules. *International Journal of Computer Science Issues*, **7**(3).

203. Xhemali, D., Hinde, C.J. & Stone, R.G. (2010-b). Genetic Evolution of Regular Expressions for the Automated Extraction of Course Names from the Web. *Proceedings of the International Conference on Genetic and Evolutionary Methods*, Las Vegas, U.S.A, pp.118-124.

204. Xhemali, D., Hinde, C.J. & Stone, R.G. (2010-c). Genetic Evolution of 'Sorting' Programs through a Novel Genotype-Phenotype Mapping. *Proceedings of the International Conference on Evolutionary Computation*, Valencia, Spain.

205. Yadav, D., Sharma, A.K. & Gupta, J.P. (2009). Topical Web Crawling Using Weighted Anchor Text, *Proceedings of the International Conference on Web Intelligence Systems, India*, pp. 145-152.

206. Yang, Y., Slattery, S. & Ghani, R. (2004). A Study of Approaches to Hypertext Categorization. *Journal of Intelligent Information Systems.* **8**(2-3), pp. 219-241.

207. Yang, Y. & Zhang, H. (2001). HTML Page Analysis based on Visual Cues. *Proceedings of the 6th International Conference on Document Analysis and Recognition.* IEEE Computer Society, pp. 859.

208. Yosif, N., Gramm, J., Wang, Q., Noble, W., Karp, R. & Sharan, R. (2010). Prediction of Phenotype Information from Genotype Data. *Communications in Information and Systems.* **10**(2), pp. 99-114.

209. Yu, H., Han, J. & Chang, K. (2004). PEBL: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering,* **16**(1), pp. 70-81.

210. Zanger, L., 2004. Mining of Data From Brokerage Listings Is Held Not to Infringe Copyright. *Holland+Knight,* **7**(3).

211. Zhang, H. (2004). The Optimality of Naïve Bayes. *Proceedings of the American Association for Artificial Intelligence.*

212. Zhang, J. & Gong, S. (2010). Action categorization by structural probabilistic latent semantic analysis. *Computer Vision and Image Understanding*, Elsevier.

213. Zhang, H. & Ling, C.X. (2001). An Improved Learning Algorithm for Augmented Naïve Bayes. *Lecture Notes in Computer Science,* **2035**, pp. 581-586.

214. Zhang, N. (2002). Hierarchical latent class models for cluster analysis. *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 230-237.

215. Zhang, Y. & Rockett, P.I. (2006). Feature Extraction using Multi-Objective Genetic Programming. *Studies in Computational Intelligence,* Springer Berlin, **16**, pp. 75-99.

216. Zhang, Y., Zincir-Heywood, N. & Milios, E. (2004). World wide web site summarization. *Web Intelligence and Agent Systems.* **2**(1), pp. 39-54.

217. Zhu, S., Yu, K., Chi, Y. & Gong, Y. (2007). Combining Content and Link for Classification using Matrix Factorization. *Proceedings of the 30th Annual International ACM SIGIR Conference, Amsterdam.*

APPENDIX A: WIE AND COPYRIGHT INFRINGEMENT

The World Wide Web (WWW) is very often addressed as a 'public' resource of information, however this does not mean that the content of websites can be used without permission from their authors, unless the website clearly states that the content can be downloaded for personal use. Copyright laws exist to enforce the above and to ensure that original pieces of work are not exploited without any benefit to the author of the work. The UK Patent Office believes that Copyright laws are important in promoting creative thinking and the development of beneficial new products.

The content of every website is automatically protected by the Copyright Law as soon as it is published online. As stated by The UK Patent Office (2009) some of the online materials protected by the Copyright Law are:

- Original written work: Also referred to as Literacy Work, this includes novels, newspaper articles, song lyrics etc. There is no copyright in names, titles, phrases or slogans.

- Original photographs

- Downloading or uploading other people's information from/onto the internet

- Databases: Their content must be original for the law to apply and / or there must have been a considerable investment in presenting its content

- Computer Programs: These are protected because for one to make use of such programs, one has to download them and install them onto a personal machine, which is a case of copying the work. Additionally, if a person creates a computer program by changing the programming language used to write the code, then this is classified as adapting the work and it is also prohibited.

The above protection is valuable for website owners, however it presents a problem for new technologies such as WIE, Internet Mashups etc., which rely on online information to automate the users' processes of gathering information. WIR and WIE are intrusive by nature as they access web pages without permission, however, the biggest problem lies in these systems storing information from web pages into local data stores such as databases.

The following discusses two court cases related to WIR and WIE. These cases show that due to WIE being relatively new and a great technological advancement, the Copyright laws do not apply to these cases as rigidly as standard cases.

Case 1: Nautical Solutions Marketing Inc. vs. Boats.com

According to a Holland+Knight Intellectual Property and Technology paper by Zanger (2004), the case arose when Nautical Solutions Marketing (NSM) created a website containing listings of yachts, which it extracted from different websites including Yachtworld.com which is owned and operated by Boats.com. The extraction was performed using WIE techniques, which meant that NSM would temporarily store the content of the web pages visited onto their local machines, whilst trying to extract certain details about yachts such as manufacturer, model, price, location and URL of the web page containing the initial yacht listing.

Boats.com filed a lawsuit against NSM for copyright infringement on their side.

The court ruled in favour of NSM. The decision was based on the following:

- NSM copied the content of web pages only momentarily and only for the purpose of extracting facts, thus it was a fair use and not an infringement.
- The material extracted as regards to yachts were purely factual and as such not protected under the copyright laws.

Case 2: eBay vs. Bidder's Edge

According to the file created on this case from the United States District Court (2000), eBay is described as a big online auction site, advertising items grouped in over 2500 categories. The file continues to state that by year 2000, eBay had over 7 million users, 400,000 new items added to the site every day and 600 bids placed every minute on nearly 3 million items. This shows the size and power of this corporation.

Bidder's Edge (BE) on the other hand, is a small company with 22 employees working for it. Its purpose is not to act as an auctioning site. BE collects information from different auction sites in order to give people the ability to search for items being auctioned without having to search each website individually. Similar to the previous court case, BE also used WIE techniques to extract the required information from various sites, including eBay, giving the users full details of products and the URLs to the auctioning sites offering each item.

eBay filed a law suit against BE claiming Infringement of Copyright and Unauthorised Interference.

The court's decision ruled out the Copyright Infringement theory, based on the fact that copying auction listings is a right within the federal copyright law. However, eBay provided evidence that BE was accessing their website 100,000 times a day, which slowed down the overall performance and processes of the system. On this evidence, the court granted an order that barred BE from data mining eBay's sites.

Both cases are very similar to the author's research, because this research also extracts information from different websites and stores them in the ATM's database. The sponsoring company does not take ownership of the courses retrieved; ATM provides, together with other course details, the name and link of the training provider offering each course.

The ruling out of copyright infringement in both court cases is good news for this research and others like it. Furthermore, the CRAWLER is designed to adhere to the mining rules set out by websites and it is adapted to only visit the same server every 1000 links (explained in section 2.2.1)

Currently, Copyright laws linked to WIR and WIE are softer that they could be. However, the risk exists that stricter laws will apply to these systems in the future, potentially bringing the whole data mining industry to a halt.

APPENDIX B: ALTERNATIVE DATABASE STRUCTURES

The following database structures were considered for the classification part of the system, however they were then deemed incorrect for reasons explained below:

Database Structure – 2:

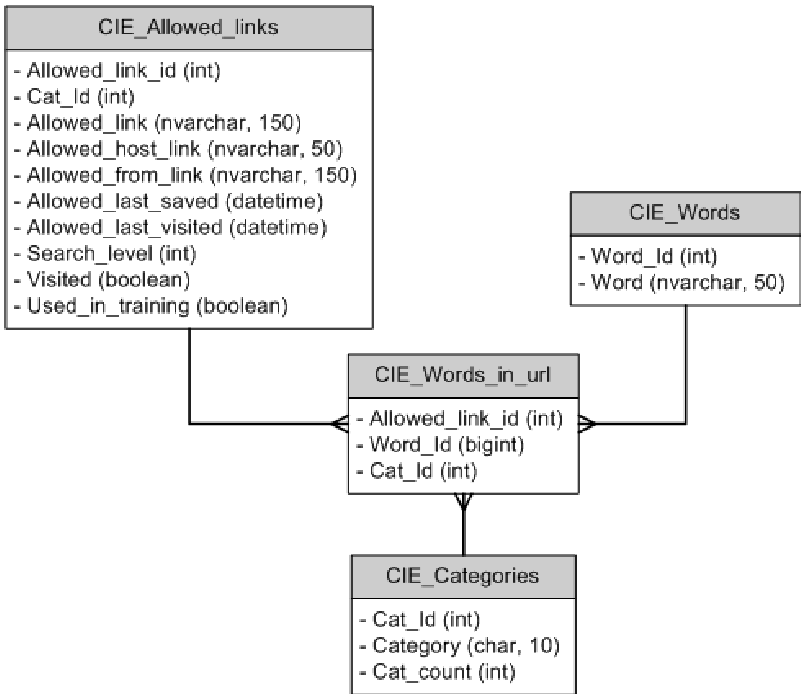


Figure 7.1: Database Structure - 2

Figure 7.1 shows a ternary relationship, where table CIE_Words_in_url resolves the many-to-many (N:N) relationships that exists between the three pairs of tables. The primary key (PK) of CIE_Words_in_url is a composite key created by the PKs of each of the other three tables.

Normalising the above relationship to Fifth Normal Form (5NF) would be equivalent to converting it to the relationship shown in Figure 7.2.

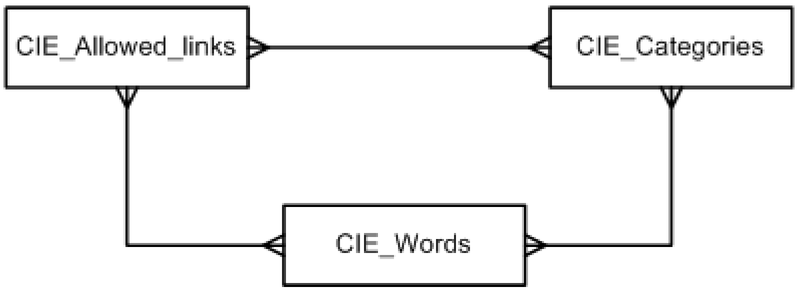


Figure 7.2: Fifth Normal Form

According to Dawson (1997), if this conversion were possible, it would mean that the original design of the 3-way relationship was a mistake. The relationships in Figure 7.2 however, are not all correct, as one link can only belong to one category, thus the relationship between

CIE_Allowed_links and CIE_Categories is 1:N not N:N. This observation gives hope that the database structure in Figure 7.1 is correct; however this relationship would fail to produce the count of a specific word in a particular category regardless of the link(s) in which it is found. The relationship would only produce the count of a specific word, found in a specific link, assigned to a specific category. This is not sufficient, thus the database structure in Figure 7.1 was deemed incorrect.

Database Structure – 3:

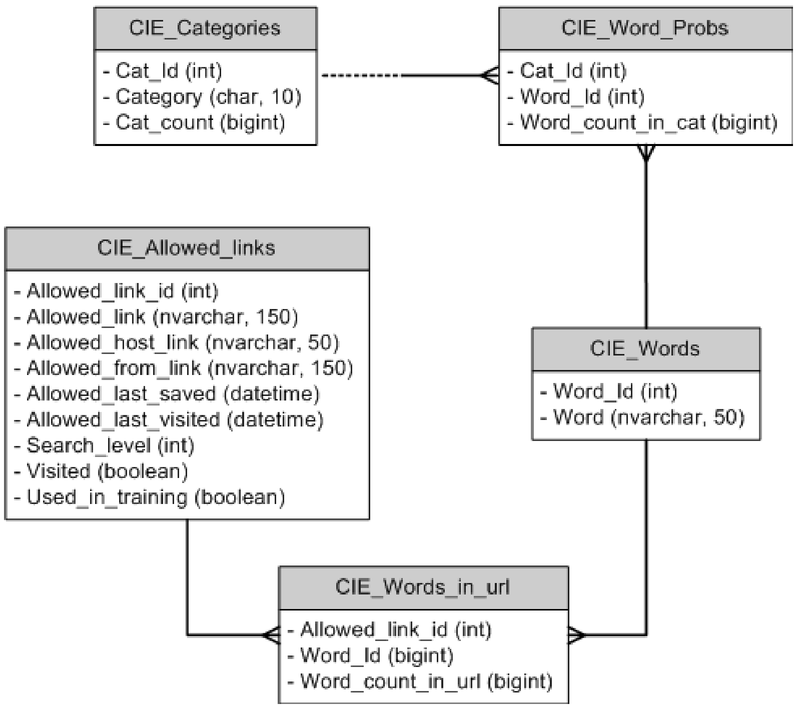


Figure 7.3: Database Structure - 3

This structure is very similar to the one chosen for the project (Figure 4.2). The difference is that in this structure there is no relationship between tables CIE_Categories and CIE_Allowed_links.

This structure is incomplete. If some of the relationships are optional, eliminating them may cause information loss. This is also known as 'Chasm Trap' (Dawson, 1997). In this case, by removing the relationship between CIE_Categories and CIE_Allowed_links, there is no way of finding out the categories to which links are assigned. This would be loss of information and as such this structure would be an insufficient solution to this project.

APPENDIX C: STOPWORDS

Table 7.1 shows the main list of Stopwords used, which is standard to most domains:

Table 7.1: Stopwords

a's	able	about	above	according
accordingly	across	actually	after	afterwards
again	against	ain't	all	allow
allows	almost	alone	along	already
also	although	always	am	among
amongst	an	and	another	any
anybody	anyhow	anyone	anything	anyway
anyways	anywhere	apart	appear	appreciate
appropriate	are	aren't	around	as
aside	ask	asking	associated	at
available	away	awfully	be	became
because	become	becomes	becoming	been
before	beforehand	behind	being	believe
below	beside	besides	best	better
between	beyond	both	brief	but
by	c'mon	c's	came	can
can't	cannot	cant	cause	causes
certain	certainly	changes	clearly	co
com	come	comes	concerning	consequently
consider	considering	contain	containing	contains
corresponding	could	couldn't	course	currently
definitely	described	despite	did	didn't
different	do	does	doesn't	doing
don't	done	down	downwards	during
each	edu	eg	eight	either
else	elsewhere	enough	entirely	especially
et	etc	even	ever	every
everybody	everyone	everything	everywhere	ex
exactly	example	except	far	few
fifth	first	five	followed	following
follows	for	former	formerly	forth
four	from	further	furthermore	get
gets	getting	given	gives	go

goes	going	gone	got	gotten
greetings	had	hadn't	happens	hardly
has	hasn't	have	haven't	having
he	he's	hello	help	hence
her	here	here's	hereafter	hereby
herein	hereupon	hers	herself	hi
him	himself	his	hither	hopefully
how	howbeit	however	i'd	i'll
i'm	i've	ie	if	ignored
immediate	in	inasmuch	inc	indeed
indicate	indicated	indicates	inner	insofar
instead	into	inward	is	isn't
it	it'd	it'll	it's	its
itself	just	keep	keeps	kept
know	knows	known	last	lately
later	latter	latterly	least	less
lest	let	let's	like	liked
likely	little	look	looking	looks
ltd	mainly	many	may	maybe
me	mean	meanwhile	merely	might
more	moreover	most	mostly	much
must	my	myself	name	namely
nd	near	nearly	necessary	need
needs	neither	never	nevertheless	new
next	nine	no	nobody	non
none	noone	nor	normally	not
nothing	novel	now	nowhere	obviously
of	off	often	oh	ok
okay	old	on	once	one
ones	only	onto	or	other
others	otherwise	ought	our	ours
ourselves	out	outside	over	overall
own	particular	particularly	per	perhaps
placed	please	plus	possible	presumably
probably	provides	que	quite	qv
rather	rd	re	really	reasonably
regarding	regardless	regards	relatively	respectively
right	said	same	saw	say

saying	says	second	secondly	see
seeing	seem	seemed	seeming	seems
seen	self	selves	sensible	sent
serious	seriously	seven	several	shall
she	should	shouldn't	since	six
so	some	somebody	somehow	someone
something	sometime	sometimes	somewhat	somewhere
soon	sorry	specified	specify	specifying
still	sub	such	sup	sure
t's	take	taken	tell	tends
th	than	thank	thanks	thanx
that	that's	thats	the	their
theirs	them	themselves	then	thence
there	there's	thereafter	thereby	thereafter
therein	theres	thereupon	these	therefore
they'd	they'll	they're	they've	they
third	this	thorough	thoroughly	think
though	three	through	throughout	those
thus	to	together	too	thru
toward	towards	tried	tries	took
try	trying	twice	two	truly
under	unfortunately	unless	unlikely	un
unto	up	upon	us	until
used	useful	uses	using	use
value	various	very	via	usually
vs	want	wants	was	viz
way	we	we'd	we'll	wasn't
we've	welcome	well	went	we're
weren't	what	what's	whatever	were
whence	whenever	where	where's	when
whereas	whereby	wherein	whereupon	whereafter
whether	which	while	whither	wherever
who's	whoever	whole	whom	who
why	will	willing	wish	whose
within	without	won't	wonder	with
would	wouldn't	yes	yet	would
you'd	you'll	you're	you've	you
yours	yourself	yourselves	zero	your

APPENDIX D: GRAMMATICAL GENOTYPE-PHENOTYPE MAPPING

<u>Sample of Grammatical Rules for the ‘Software Statements and Structures’ Domain:</u>

- <statements> ::= <statement>
- <statements> ::= <statement> <statements>
- <statements> ::= <statement>*

Rules:

- <statement> ::= IF <expression> THEN <statements> ELSE <statements> END IF
- <statement> ::= WHILE <expression> <statements> END WHILE
- <statement> ::= FOR <variable><comparison><constant> TO <variable> <statements> END FOR
- <statement> ::= <variable>
- <statement> ::= <expression>

Expressions:

- <expression> ::= <variable> <comparison> <variable>
- <expression> ::= <constant> <comparison> <variable>
- <expression> ::= <variable> <comparison> <constant>
- <expression> ::= <constant>

Variables:

- <variable> ::= ‘x’
- <variable> ::= ‘y’
- <variable> ::= ‘z’
- <variable> ::= ‘temp’

Comparisons:

- <comparison> ::= ‘<>’
- <comparison> ::= ‘==’
- <comparison> ::= ‘>’
- <comparison> ::= ‘<’

Constants:

- <constant> ::= ‘0’

Note that in “<statements> ::= <statement>*” the asterisk corresponds to a specific number of statements required for a particular software structure. This number would be specified by one of the genes in the genotype. This is important as it means that the evolution process could identify the optimal number of statements required for the generation of a complete software program.

The above grammar can be written in different ways. The different choices that are made here could affect the success and execution speed of the evolutionary system:

Variant 1: This option is most compact, however it requires the use of three genes from the genotype to determine the operator and the different operands:

<expression> ::= <varcon> <comparison> <varcon>
<varcon> ::= <variable> | <constant>
<comparison> ::= '<>' | '==' | '>' | '<'

Variant 2: This option is most verbose, however this only requires the use of one gene from the genotype. This one gene will tell the system which variation of expression to follow.

<expression> ::= <variable> '<>' <variable> | <variable> '<>' <constant> | <constant> '<>' <variable> | <constant> '<>' <constant> | <variable> '==' <variable> | <variable> '==' <constant> | <constant> '==' <variable> | <constant> '==' < constant> | …etc

<u>Sample of Grammatical Rules for the REs Domain:</u>

- <RE> ::= <REComponents>
- <REComponents> ::= <REComponent>
- <REComponents> ::= <REComponent> <REComponents>
- <REComponent> ::= <REStructure> | <RETag> | <REKeyword>

REStructures:
- <REStructure> ::= '.*?'
- <REStructure> ::= '[\s]?'

REKeywords:
- <REKeyword> ::= 'price'
- <REKeyword> ::= 'date'
- <REKeyword> ::= 'location'
- <REKeyword> ::= 'title'
- <REKeyword> ::= 'fee'

RETags:
- <RETag> ::= <REStartTag> <RETagAttributes>
- <RETag> ::= <REStartTag>
- <RETag> ::= <REEndTag>

REStartTags:
- <REStartTag> ::= '<' <Tag> <RETagAttributes> '>'
- <REStartTag> ::= '<' <Tag> '>'

REEndTags:
- <REEndTag> ::= '</' <Tag> '>'

Tags:
- <Tag> ::= 'tr'
- <Tag> ::= 'td'
- <Tag> ::= 'title'

RETagAttributes:

- <RETagAttributes> ::= " " | <RETagAttributes>
- <RETagAttributes> ::= <RETagAttribute> | <RETagAttributes>

- <RETagAttribute> ::= <AttributeName> '=' <AttributeValue>

- <AttributeName> ::= <letter> | <letter> <AttributeName>
- <AttributeValue> ::= <string> | <number>
- <string> ::= " ' " <characters> " ' "
- <number> ::= <digit> | <digit> <number>
- <characters> ::= <character> | <character> <characters>

- <digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
- <character> ::= '!' | '@' | '#' | '?' | '^' | '*' | '(' | ')' ... etc.
- <letter> ::= 'a' | 'A' | 'b' | 'B' | 'c' | 'C' | 'd' | 'D' | 'e' | 'E' ... etc.

Similarly to the grammar for 'Software Statements and Structures', the above grammar can also be written in different ways, depending on the elements to be evolved. In this research, the decision was made against character by character evolution as this would dramatically increase the search space and thus increase the execution speed of the GP system.

The above grammar falls into the character by character evolution category, however this could be changed by ignoring what the tag attributes really are and just identifying them as a whole:

<RETagAttribute> ::= AttributeName '=' AttributeValue

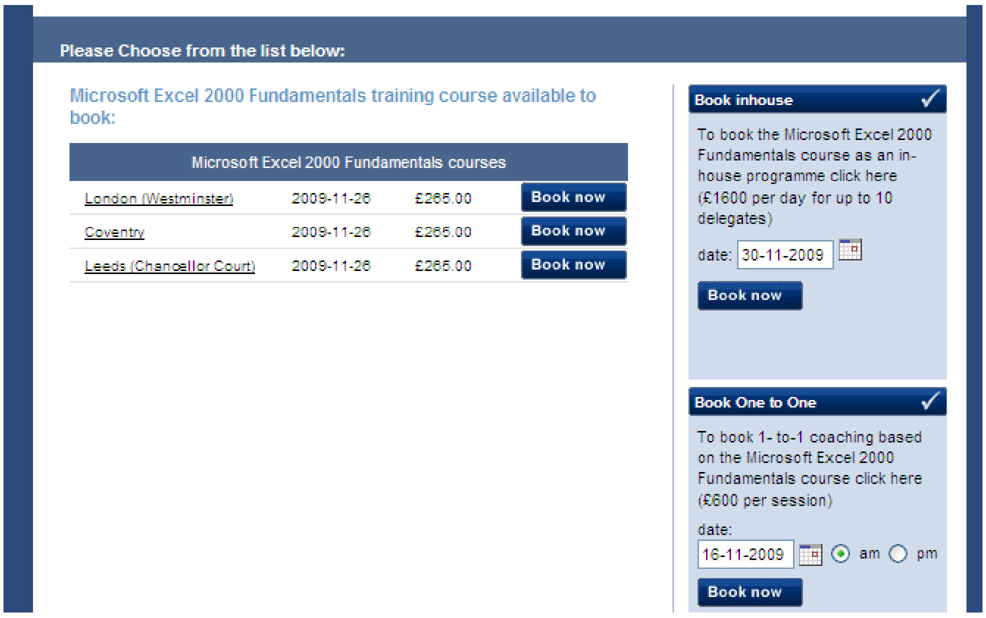
The above means that the system does not need to break down and analyse the AttributeName and AttributeValue. The system will be satisfied that these are attributes without really worrying about truly recognising them.

In summary, there is no one single way of writing these grammatical rules, however the best combination of rules could be identified through experimentation with the GP system.

APPENDIX E: WEBSITES FORMAT VARIETY

The following shows the variety in formatting styles and presentation techniques for training course information in some of the websites analysed in this research. Some HTML code has also been included from each website, to show what the EXTRACTOR needs to examine.

http://www.ptp.co.uk/Courses/Microsoft-Excel-2000-Fundamentals/ini/249



Please Choose from the list below:

Microsoft Excel 2000 Fundamentals training course available to book:

Microsoft Excel 2000 Fundamentals courses

London (Westminster)	2009-11-26	£265.00	Book now
Coventry	2009-11-26	£265.00	Book now
Leeds (Chancellor Court)	2009-11-26	£265.00	Book now

Book inhouse ✓

To book the Microsoft Excel 2000 Fundamentals course as an in-house programme click here (£1600 per day for up to 10 delegates)

date: 30-11-2009

Book now

Book One to One ✓

To book 1-to-1 coaching based on the Microsoft Excel 2000 Fundamentals course click here (£600 per session)

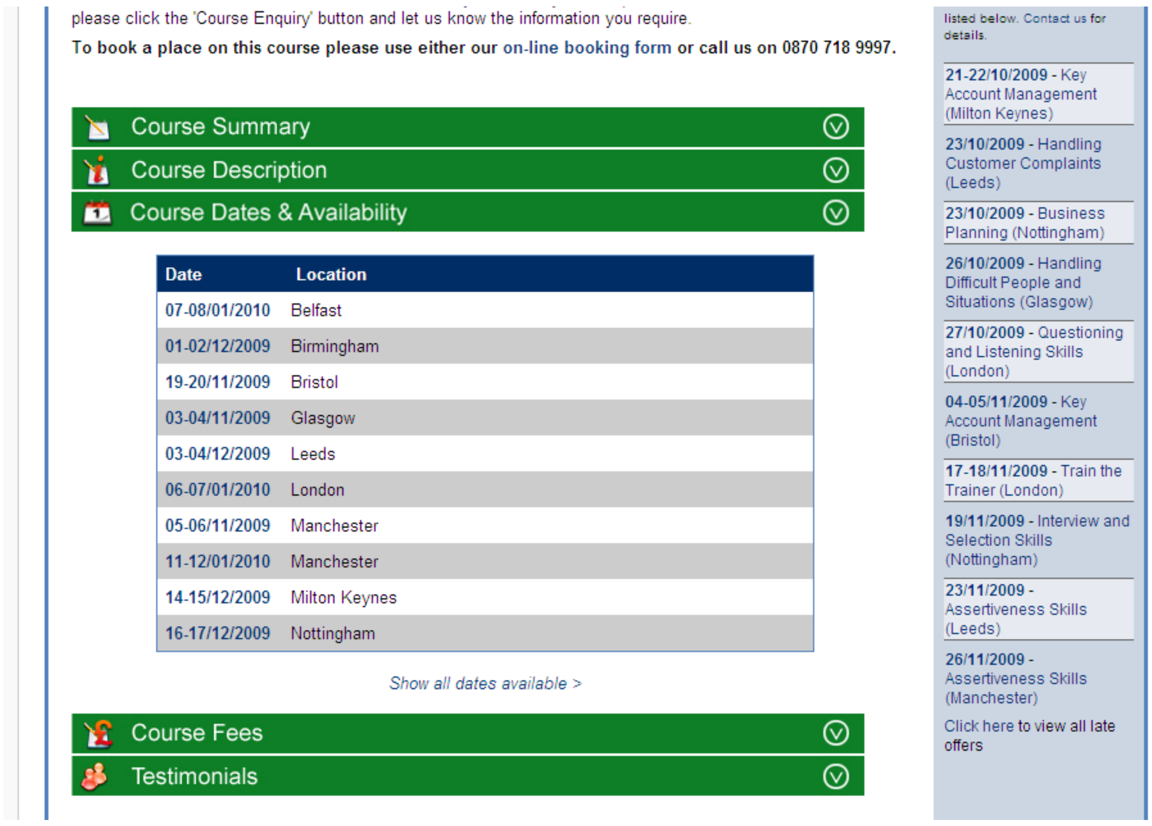
date: 16-11-2009 [✓ am] pm

Book now

```
<div id="courses"><div>
<strong>Please Choose from the list below:</strong>
</div></div>
<div id="bookcourses">
<div class="coursesdetails">
 <h3> Microsoft Excel 2000 Fundamentals training course available to book:</h3>
<a id="openCourses"></a>
 <table class="courses">
<caption>Microsoft Excel 2000 Fundamentals courses</caption>
<tr>
<th>Training Location</th>
<th>Training Date</th>
<th>Training Price</th>
<th>Book Trainging</th>
</tr>
<tr><td style='width:130px;'><a style='color:#000;display:inline; background:none;'
href="http://www.ptp.co.uk/Venues/concrete/110  ">London (Westminster) </a>
</td>
<td>2009-11-26</td>
<td style='text-align:center;'>&pound;265.00</td>
<td align="right"><a
href="http://www.ptp.co.uk/Order/putUChooseG/?uchoose_course_id=249&amp;uchoose_location_id=110&am
p;uchoose_course_date=2009-11-26"><span>Book Microsoft Excel 2000 Fundamentals Course
Online</span></a></td>
</tr>
<tr>
<td style='width:130px;'><a style='color:#000;display:inline; background:none;'
href="http://www.ptp.co.uk/Venues/concrete/111  ">Coventry </a></td>
```

```
<td>2009-11-26</td>
<td style='text-align:center;'>&pound;265.00</td>
<td align="right"><a
href="http://www.ptp.co.uk/Order/putUChooseG/?uchoose_course_id=249&amp;uchoose_location_id=111&am
p;uchoose_course_date=2009-11-26"><span>Book Microsoft Excel 2000 Fundamentals Course
Online</span></a></td>
</tr>
<tr>
<td style='width:130px;'><a style='color:#000;display:inline; background:none;'
href="http://www.ptp.co.uk/Venues/concrete/115  ">Leeds (Chancellor Court)
 </a></td>
<td>2009-11-26</td>
<td style='text-align:center;'>&pound;265.00</td>
<td align="right"><a
href="http://www.ptp.co.uk/Order/putUChooseG/?uchoose_course_id=249&amp;uchoose_location_id=115&am
p;uchoose_course_date=2009-11-26"><span>Book Microsoft Excel 2000 Fundamentals Course
Online</span></a></td>
</tr>
</table>
```

http://www.trainanddevelop.co.uk/view_course.php?CourseID=2


please click the 'Course Enquiry' button and let us know the information you require.

To book a place on this course please use either our on-line booking form or call us on 0870 718 9997.

Date	Location
07-08/01/2010	Belfast
01-02/12/2009	Birmingham
19-20/11/2009	Bristol
03-04/11/2009	Glasgow
03-04/12/2009	Leeds
06-07/01/2010	London
05-06/11/2009	Manchester
11-12/01/2010	Manchester
14-15/12/2009	Milton Keynes
16-17/12/2009	Nottingham

Show all dates available >

listed below. Contact us for details.

21-22/10/2009 - Key Account Management (Milton Keynes)

23/10/2009 - Handling Customer Complaints (Leeds)

23/10/2009 - Business Planning (Nottingham)

26/10/2009 - Handling Difficult People and Situations (Glasgow)

27/10/2009 - Questioning and Listening Skills (London)

04-05/11/2009 - Key Account Management (Bristol)

17-18/11/2009 - Train the Trainer (London)

19/11/2009 - Interview and Selection Skills (Nottingham)

23/11/2009 - Assertiveness Skills (Leeds)

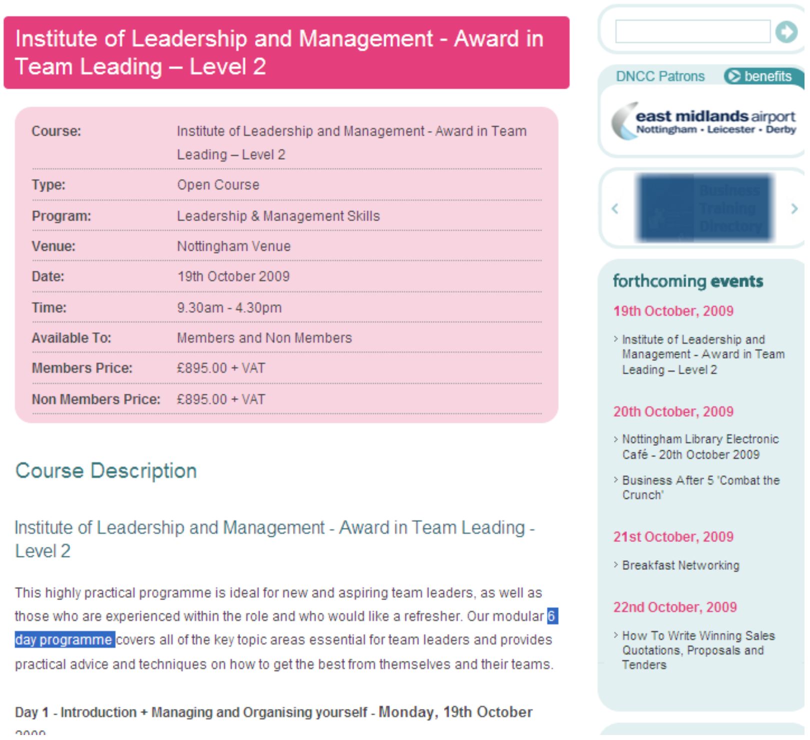
26/11/2009 - Assertiveness Skills (Manchester)

Click here to view all late offers

```
<table width="100%" border="0" cellspacing="0" cellpadding="2" style='margin:auto'>
<tr>
<th scope="col"  style='width:20%' align='left'>Date</th>
<th style='width:35%' scope="col" align='left' >Location</th>
 <th  scope="col" ></th>
</tr>
<tr >
<th  scope='row'  class='nobg'   align='left'><A  HREF='book.php?CourseID=2'   class='listing'>07-
08/01/2010</a></th>
```

```
<td>Belfast</td>
<td align='center'></td>
</tr>
<tr  style='background-color:#CCCCCC'>
<th   scope='row'   class='thbg'     align='left'><A   HREF='book.php?CourseID=2'     class='listing'>01-
02/12/2009</a></th>
<td>Birmingham</td>
<td align='center'></td>
</tr>
<tr >
<th   scope='row'   class='nobg'     align='left'><A   HREF='book.php?CourseID=2'     class='listing'>19-
20/11/2009</a></th>
<td>Bristol</td>
<td align='center'></td>
</tr>
<tr  style='background-color:#CCCCCC'>
<th   scope='row'   class='thbg'     align='left'><A   HREF='book.php?CourseID=2'     class='listing'>03-
04/11/2009</a></th>
<td>Glasgow</td>
<td align='center'></td>
</tr>
<tr >
<th   scope='row'   class='nobg'     align='left'><A   HREF='book.php?CourseID=2'     class='listing'>03-
04/12/2009</a></th>
<td>Leeds</td>
<td align='center'></td>
</tr>
</table>
```

http://www.dncc.co.uk/your-chamber/2954/institute-of-leadership-and-management-award-in-team-leading-level-2



Institute of Leadership and Management - Award in Team Leading – Level 2

Course:	Institute of Leadership and Management - Award in Team Leading – Level 2
Type:	Open Course
Program:	Leadership & Management Skills
Venue:	Nottingham Venue
Date:	19th October 2009
Time:	9.30am - 4.30pm
Available To:	Members and Non Members
Members Price:	£895.00 + VAT
Non Members Price:	£895.00 + VAT

Course Description

Institute of Leadership and Management - Award in Team Leading - Level 2

This highly practical programme is ideal for new and aspiring team leaders, as well as those who are experienced within the role and who would like a refresher. Our modular 6 day programme covers all of the key topic areas essential for team leaders and provides practical advice and techniques on how to get the best from themselves and their teams.

Day 1 - Introduction + Managing and Organising yourself - Monday, 19th October

DNCC Patrons benefits

forthcoming events

19th October, 2009

› Institute of Leadership and Management - Award in Team Leading – Level 2

20th October, 2009

› Nottingham Library Electronic Café - 20th October 2009

› Business After 5 'Combat the Crunch'

21st October, 2009

› Breakfast Networking

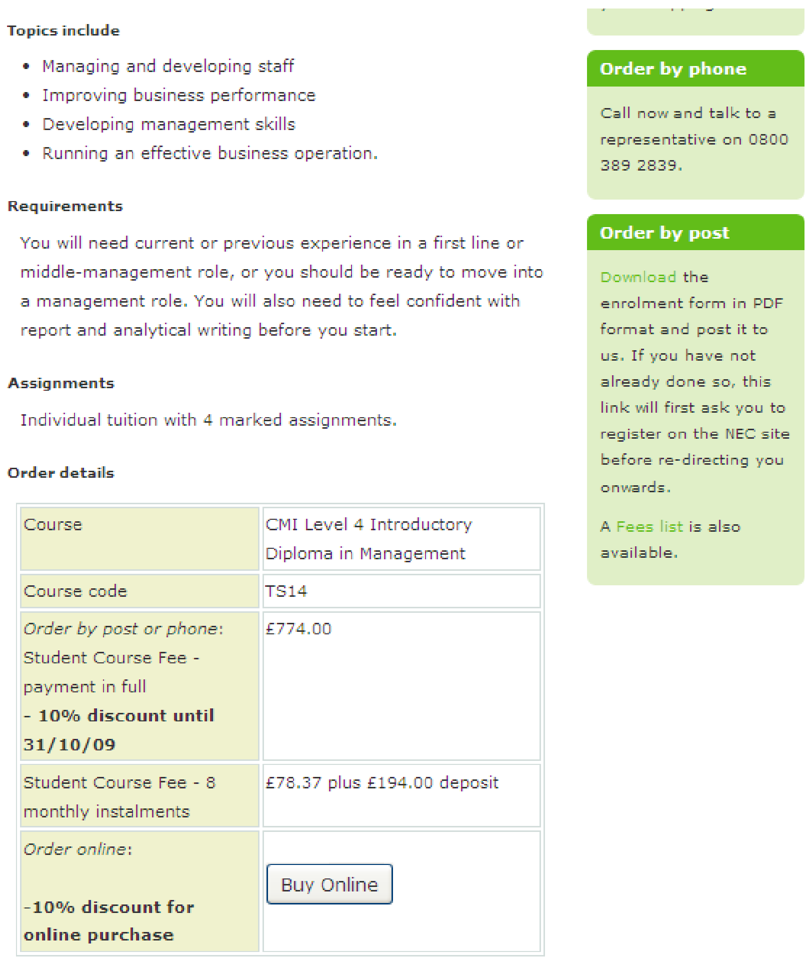
22nd October, 2009

› How To Write Winning Sales Quotations, Proposals and Tenders

```
<div id="pageContent">
```

```
<!--      <span class="breadCrumbs"><a href="/">Home</a> &raquo; <a href="/">Events &amp;
Seminars</a> &raquo; <a href="/">From Friend to Supervisor</a></span> -->
<div id="pageTitleTop"><div id="pageTitleBottom">
<h1>Institute of Leadership and Management - Award in Team Leading – Level 2</h1>
</div></div>
<div id="pageContentPad">
<div id="contentForm">
<div class="contFormTop">
<div class="contFormBottom">
<table cellspacing="0" cellpadding="0" width="100%">
<tr>
<td class="contFormRow" width="130"><strong>Course:</strong></td>
<td class="contFormRow">Institute of Leadership and Management - Award in Team Leading – Level
2</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Program:</strong></td>
<td class="contFormRow">Leadership &amp; Management Skills</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Venue:</strong></td>
<td class="contFormRow">Nottingham Venue</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Date:</strong></td>
<td class="contFormRow">19th October 2009</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Time:</strong></td>
<td class="contFormRow">9.30am - 4.30pm</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Available To:</strong></td>
<td class="contFormRow">Members and Non Members</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Members Price:</strong></td>
<td class="contFormRow">&pound;895.00 + VAT</td>
</tr>
<tr>
<td class="contFormRow" width="130"><strong>Non Members Price:</strong></td>
<td class="contFormRow">&pound;895.00 + VAT</td>
</tr>
</table>
</div></div></div>
<h2>Course Description</h2>
<!-- content starts here -->
<h3>Institute of Leadership and Management - Award in Team Leading - Level 2</h3>
<br /><br /><strong>Day 1 - Introduction + Managing and Organising yourself -&nbsp;<span style="font-
size: 10pt; font-family: Arial">Monday, </span></strong><span style="font-size: 10pt; font-family:
Arial"><strong>19th October</strong> 2009 </span></p><span style="font-size: 10pt; font-family:
Arial"><ul><li style="margin-left: 0.5in; text-indent: -0.5in; tab-stops: list 15.9pt"><span style="font-size:
10pt; font-family: Arial">Welcome &amp; introductions</span></li>
```

http://www.nec.ac.uk/management/product?product_id=1443&category_id=4620



Topics include

- Managing and developing staff
- Improving business performance
- Developing management skills
- Running an effective business operation.

Requirements

You will need current or previous experience in a first line or middle-management role, or you should be ready to move into a management role. You will also need to feel confident with report and analytical writing before you start.

Assignments

Individual tuition with 4 marked assignments.

Order details

Course	CMI Level 4 Introductory Diploma in Management
Course code	TS14
Order by post or phone: Student Course Fee - payment in full **- 10% discount until 31/10/09**	£774.00
Student Course Fee - 8 monthly instalments	£78.37 plus £194.00 deposit
Order online: **-10% discount for online purchase**	Buy Online

Order by phone

Call now and talk to a representative on 0800 389 2839.

Order by post

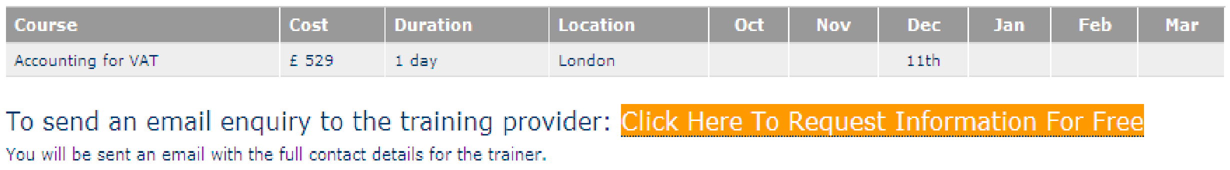
Download the enrolment form in PDF format and post it to us. If you have not already done so, this link will first ask you to register on the NEC site before re-directing you onwards.

A Fees list is also available.

```
<h4>Requirements</h4>
<p>You will need current or previous experience in a first line or middle-management role, or you should be ready to move into a management role. You will also need to feel confident with report and analytical writing before you start.</p>
<h4>Assignments</h4>
<p>Individual tuition with 4 marked assignments.</p>
<h4>Order details</h4>
<table id="order-table">
<colgroup>
<col class="labels" />
<col />
</colgroup>
<tr>
<td>Course</td>
<td>CMI Level 4 Introductory Diploma in Management</td>
</tr>
<tr>
<td>Course code</td>
<td>TS14</td>
</tr>
<tr>
<td><i>Order by post or phone</i>:<br>Student Course Fee - payment in full <br>
<b>- 10% discount until 31/10/09</b>
</td>
```

```
<td>&pound;774.00</td>
</tr>
<tr>
<td>Student Course Fee - 8 monthly instalments</td>
<td>&pound;78.37  plus &pound;194.00 deposit</td>
</tr>
<tr>
<td><i>Order online</i>:<br><br><b>-10% discount for online purchase</b></td>
<td>
<br><form method=post action="shopping-cart-add">
<input type="hidden" name="product_id" value="1443" />
<input type="hidden" name="category_id" value="4620" />
<input type="hidden" name="catalogue_id" value="13" />
<input type="hidden" name="return_url" value="" />
<input type=submit value="Buy Online"><br>
</form>
</td>
</tr>
</table>
```

http://www.underoak.co.uk/public-training-courses/accountancy-courses/skills-training.html



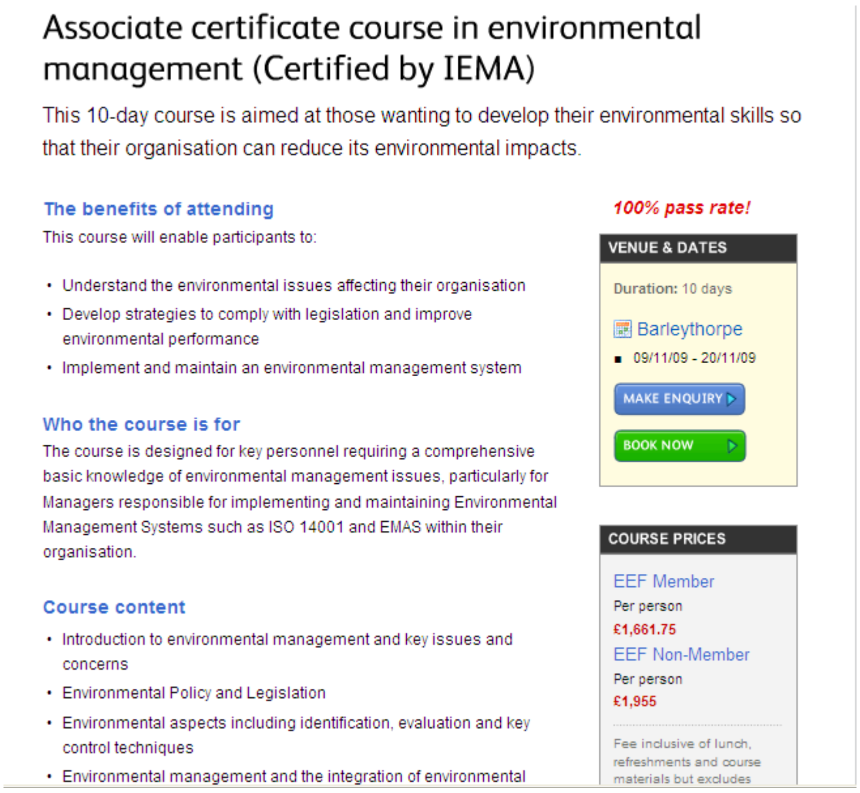
Course	Cost	Duration	Location	Oct	Nov	Dec	Jan	Feb	Mar
Accounting for VAT	£ 529	1 day	London			11th			

To send an email enquiry to the training provider: Click Here To Request Information For Free

You will be sent an email with the full contact details for the trainer.

```
<div class="column span-22">
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr id="head">
<td valign="middle" class="left white bdr"><b>Course</b></td>
<td valign="middle" class="left white bdr"><b>Cost</b></td>
<td valign="middle" class="left white bdr"><b>Duration</b></td>
<td valign="middle" class="left white bdr"><b>Location</b></td>
<td valign="middle" class="center white bdr"><b>Oct</b></td>
<td valign="middle" class="center white bdr"><b>Nov</b></td>
<td valign="middle" class="center white bdr"><b>Dec</b></td>
<td valign="middle" class="center white bdr"><b>Jan</b></td>
<td valign="middle" class="center white bdr"><b>Feb</b></td>
<td valign="middle" class="center white bdr"><b>Mar</b></td>
</tr>
<tr id="row1">
<td valign="top" class="left sml bdr">Accounting for VAT</td>
<td valign="top" class="left sml bdr">&pound;&nbsp;529</td>
<td valign="top" class="left sml bdr">1 day</td>
<td valign="top" class="left sml bdr">London</td>
<td class="sml bdr">&nbsp;</td>
<td class="sml bdr">&nbsp;</td>
<td valign="top" class="center sml bdr">11th</td>
<td class="sml bdr">&nbsp;</td>
<td class="sml bdr">&nbsp;</td>
</tr>
</table>
</div>
```

http://www.barleythorpe.com/courses.aspx?r=116


Associate certificate course in environmental management (Certified by IEMA)

This 10-day course is aimed at those wanting to develop their environmental skills so that their organisation can reduce its environmental impacts.

The benefits of attending

This course will enable participants to:

- Understand the environmental issues affecting their organisation
- Develop strategies to comply with legislation and improve environmental performance
- Implement and maintain an environmental management system

Who the course is for

The course is designed for key personnel requiring a comprehensive basic knowledge of environmental management issues, particularly for Managers responsible for implementing and maintaining Environmental Management Systems such as ISO 14001 and EMAS within their organisation.

Course content

- Introduction to environmental management and key issues and concerns
- Environmental Policy and Legislation
- Environmental aspects including identification, evaluation and key control techniques
- Environmental management and the integration of environmental

100% pass rate!

VENUE & DATES

Duration: 10 days

Barleythorpe

- 09/11/09 - 20/11/09

COURSE PRICES

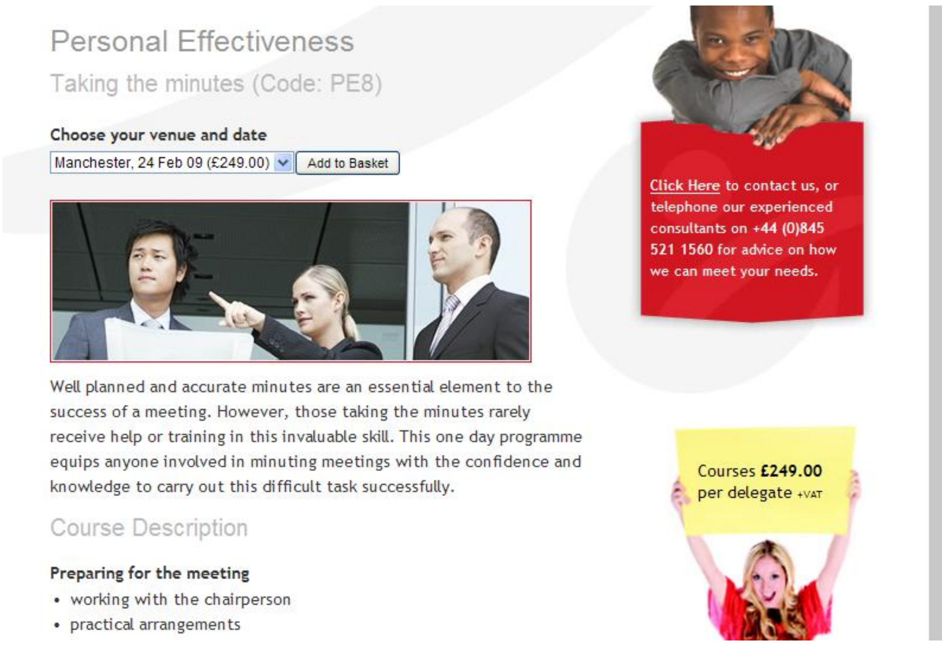
EEF Member
Per person
£1,661.75
EEF Non-Member
Per person
£1,955

Fee inclusive of lunch, refreshments and course materials but excludes

```
<div class="course-module">
<p class="back"><a href="/courses.aspx">Back to course list</a></p>
<h1>Associate certificate course in environmental management (Certified by IEMA)</h1>
<p class="intro">This 10-day course is aimed at those wanting to develop their environmental skills so that their organisation can reduce its environmental impacts. </p>
<div class="course-sidebar"><p class="new">100% pass rate!</p>
<div class="dates">
<h2><img src="/themes/images/titles/dates.gif" alt="Venue &amp; Dates" width="150" height="22" /></h2>
<p><strong>Duration:</strong>10 days</p>
<p></p>
<h3>Barleythorpe</h3>
<ul><li>09/11/09 - 20/11/09</li></ul>
<p class="button"><a href="/contact.aspx?r=7735&t=e"><img src="/themes/images/buttons/make-enquiry.gif" alt="Make Enquiry" width="101" height="26" /></a></p><p class="button"><a href="/contact.aspx?r=7735&t=b"><img src="/themes/images/buttons/book-now.gif" alt="Book Now" width="102" height="25" /></a></p></div>
<div class="prices"><h2><img src="/themes/images/titles/prices.gif" alt="Course prices" width="150" height="22" /></h2>
<h3 class="first">EEF Member</h3>
<p class="price">Per person<br/><strong>&#163;1,661.75</strong></p>
<h3 class="first">EEF Non-Member</h3>
<p class="price">Per person<br/><strong>&#163;1,955</strong></p>
<p class="additional">Fee inclusive of lunch, refreshments and course materials but excludes VAT.</p>
</div>
<div class="prices">
<h2><img src="/themes/images/titles/optional.gif" alt="Optional" width="150" height="22" /></h2>
Bed &amp; breakfast
<h3 class="first">EEF Member</h3>
<p class="price"><strong>£48.75</strong></p>
<h3 class="first">EEF Non-Member</h3>
```

```
<p class="price"><strong>£51.10</strong></p>
<p class="additional"> Price per person per night in ensuite bedroom inclusive of full English breakfast but
excluding VAT.</p>
</div>
</div>
<h3>The benefits of attending </h3>
<p>This course will enable participants to: </p>
…
```

http://www.cegos.co.uk/products/taking-the-minutes.php



Personal Effectiveness

Taking the minutes (Code: PE8)

Choose your venue and date

Manchester, 24 Feb 09 (£249.00) | Add to Basket

Click Here to contact us, or telephone our experienced consultants on +44 (0)845 521 1560 for advice on how we can meet your needs.

Well planned and accurate minutes are an essential element to the success of a meeting. However, those taking the minutes rarely receive help or training in this invaluable skill. This one day programme equips anyone involved in minuting meetings with the confidence and knowledge to carry out this difficult task successfully.

Courses £249.00 per delegate +VAT

Course Description

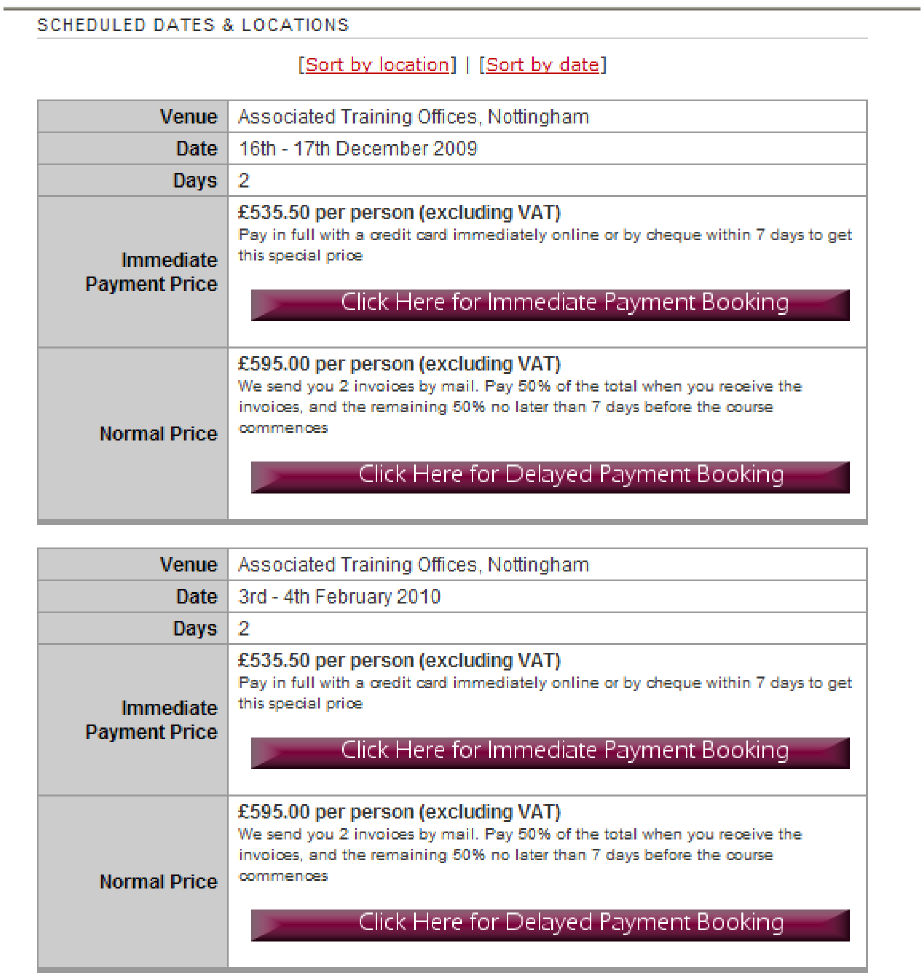
Preparing for the meeting

- working with the chairperson
- practical arrangements

```
<form action="/products/taking-the-minutes.php" method="post" enctype="multipart/form-data">
<div id="product-details-page">
<!-- H1 --><h3>Open Courses</h3>
<!-- Breadcrumb -->
<div id="breadcrumb">
<p>
<a href="index.php">Home</a>
<a href='categories/tutored.php'>Open Courses</a>
<a href='categories/personal-effectiveness.php'>Personal Effectiveness</a>
Taking the minutes (Code: PE8)                              </p>
</div>
<!-- H2 -->
<!-- if the category is same as h1 tag (eg in custom courses) display the course name -->
<h2>Personal Effectiveness</h2>
<!-- H3 -->
<!-- if the category is NOT the same as h1 tag print the course -->
<h1>Taking the minutes (Code: PE8)</h1>
</div>
<p><strong>Choose your venue and date</strong></p>
<input name='qty' type='hidden' value='1' />
<select name='variantsList' onchange='this.form.submit();'>
<option value='115'>Manchester, 24 Feb 09 (&pound;249.00)</option>
<option value='114'>London, 08 Oct 09 (&pound;249.00)</option>
</select>
<input name='Taking the minutes (Code: PE8)' type='submit' value='Add to Basket' />&nbsp;
<div style='margin-top: 20px;'>
```

```
<img src='pics/products/taking-the-minutes-view1-large.jpg' alt='Taking the Minutes' />
</div>
<p><p>Well planned and accurate minutes are an essential element to the success of a meeting. However, those taking the minutes rarely receive help or training in this invaluable skill. This one day programme equips anyone involved in minuting meetings with the confidence and knowledge to carry out this difficult task successfully.</p>
<h3>Course Description</h3><br />
<p><strong>Preparing for the meeting</strong></p>
<ul>
<li>working with the chairperson</li>
<li>practical arrangements</li>
</ul>
…
<h3>Course Duration</h3>
<p>1 days</p>
</form>
```

http://www.associatedtraining.co.uk/courses/management/train_the_trainer.php


SCHEDULED DATES & LOCATIONS

[Sort by location] | [Sort by date]

Venue	Associated Training Offices, Nottingham
Date	16th - 17th December 2009
Days	2
Immediate Payment Price	£535.50 per person (excluding VAT) Pay in full with a credit card immediately online or by cheque within 7 days to get this special price
Normal Price	£595.00 per person (excluding VAT) We send you 2 invoices by mail. Pay 50% of the total when you receive the invoices, and the remaining 50% no later than 7 days before the course commences

Venue	Associated Training Offices, Nottingham
Date	3rd - 4th February 2010
Days	2
Immediate Payment Price	£535.50 per person (excluding VAT) Pay in full with a credit card immediately online or by cheque within 7 days to get this special price
Normal Price	£595.00 per person (excluding VAT) We send you 2 invoices by mail. Pay 50% of the total when you receive the invoices, and the remaining 50% no later than 7 days before the course commences

Click Here for Immediate Payment Booking
Click Here for Delayed Payment Booking
Click Here for Immediate Payment Booking
Click Here for Delayed Payment Booking

```
<h2 class="sub">Scheduled Dates &amp; Locations</h2>
<p align="center">[<a href="/courses/management/page_anchors/train_the_trainer_location.php">Sort by location</a>] | [<a href="/courses/management/page_anchors/train_the_trainer_date.php">Sort by date</a>]</p>
<a name = "161209">
<table class="schedule">
```

```
<tr>
<th>Venue</th>
<td>Associated Training Offices, Nottingham</td>
</tr>
<tr>
<th>Date</th>
<td>16th - 17th December 2009</td>
</tr>
<tr>
<th>Days</th>
<td>2</td>
</tr>
<tr>
<th>Immediate Payment Price</th>
<td><b>&pound;535.50 per person (excluding VAT)</b><br />
<font size="1">Pay in full with a credit card immediately online or by cheque within 7 days to get this special
price</font><br /><br />
<p align="center"><a href="/online_booking/ttt_161209-main.php"><img border="0"
src="/images/online_booking/immediate_payment_banner.gif" width="375" height="20"></a></td>
</tr>
<tr>
<th>Normal Price</th>
<td><b>&pound;595.00 per person (excluding VAT)</b><br />
<font size="1">We send you 2 invoices by mail. Pay 50% of the total when you receive the invoices, and the
remaining 50% no later than 7 days before the course commences</font><br /><br />
<p align="center"><a href="/contact_form_course_booking.php"><img border="0"
src="/images/online_booking/delayed_payment_banner.gif" width="375" height="20"></a></td>
</tr>
</table>
<a name = "030210">
<table class="schedule">
<tr>
<th>Venue</th>
<td>Associated Training Offices, Nottingham</td>
</tr>
<tr>
<th>Date</th>
<td>3rd - 4th February 2010</td>
</tr>
<tr>
<th>Days</th>
<td>2</td>
</tr>
<tr>
<th>Immediate Payment Price</th>
<td><b>&pound;535.50 per person (excluding VAT)</b><br />
<font size="1">Pay in full with a credit card immediately online or by cheque within 7 days to get this special
price</font><br /><br />
<p align="center"><a href="/online_booking/ttt_030210-main.php"><img border="0"
src="/images/online_booking/immediate_payment_banner.gif" width="375" height="20"></a></td>
</tr>
<tr>
<th>Normal Price</th>
<td><b>&pound;595.00 per person (excluding VAT)</b><br />
</td>
</tr></table>
```

APPENDIX F: PAPER 1

Full Reference:

Xhemali, D., Hinde, C.J. and Stone, R.G. (2007). Embarking on a Web Information Extraction Project. *The 2007 UK Workshop on Computational Intelligence*, London.

Paper type: Conference Paper (Published)

Embarking on a Web Information Extraction Project

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

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

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

ABSTRACT

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

1. INTRODUCTION

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

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

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

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

2. COURSE INFORMATION EXTRACTOR (CIE)

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

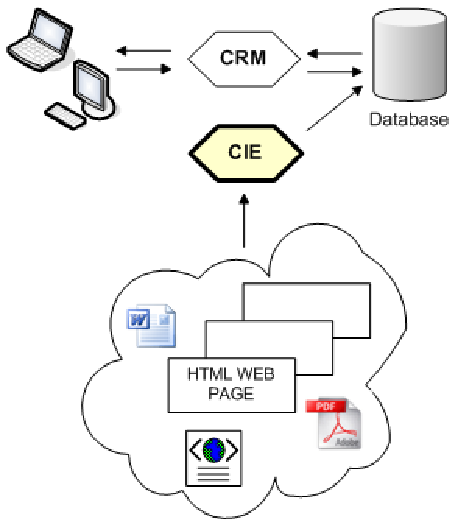


Figure 1: Course Information Extractor

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

3. WIE APPROACHES

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

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

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

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

3.1. HTML Wrapper Generation

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

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

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

A different technique is the analysis of two HTML web pages. The source code of both pages is compared and the similarities and differences found used to generate a wrapper that would apply to as many pages as possible. Road Runner [3] is a good example of this, which automates the Wrapper Generation process with no prior knowledge of the target pages, whilst managing to deal with nested HTML structures.

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

3.2. XML Wrapper Generation

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

Liu [11] develops mechanisms that provide a clean separation of the semantic meaning of information on each web page from the process of wrapper generation. Myllymaki [13] also converts each HTML web page to XML as the first step of the extraction process, however, in this case the HTML code is first translated to XHTML (eXtensible HTML), which 'repairs' the code from any abnormalities such as missing tags, etc. There are various tools that achieve this such as the Tidy package [17]. The XHTML code is then converted to XML. The XSLT language is used to achieve this by using XPath to find parts of the XHTML document that match a template, then transforming the matching part into XML.

Despite the success of this research in relying less on HTML and more on the content itself, it is still too rigid in the way it finds items of interest on the target web pages. E.g. they will look into the cell of a table and extract the information that is in bold. Needless to say, this technique will not work if the item in bold is changed to e.g. italic. Nevertheless, this is an improvement on techniques such as the one used byW4F [22], which uses absolute paths to the location of interest; the path to the third column of the second row of the first table would be: HTML.BODY.TABLE[1].TR[2].TD[3]. W4F cannot deal with layout changes.

3.3. Machine Learning (ML)

ML techniques are popular because of their ability to make intelligent guesses when extracting information from the Web and eliminate much of the user involvement. Besides Information Extraction and Data Mining, ML has an extensive spectrum of applications including: Medical diagnosis, Credit card fraud detection, Speech recognition etc. Techniques used can be categorised based on how much available feedback is given to the learning process [21]. These categories include:

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

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

Unsupervised Learning: The system learns relationships between inputs alone.

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

Crystal [26] uses a ML algorithm, however it requires a semantic hierarchy of the data, as well as the training data to be manually annotated by an expert. Crystal learns and creates extraction rules by generating multi-slot concept frames. This allows for related information to be extracted together. The multi-slot concept is essential to our research as well, because a web page may list many courses with associated titles, prices, locations etc. however, unless all this information is extracted as a set, the result would not be useful.

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

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

3.4. Extraction Ontologies

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

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

3.4.1. Reusability

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

3.4.2. Ontology Maintenance

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

The focus of the research from Snoussi [23] is websites which change the content frequently but not the overall structure e.g. stock exchange quotes. The system works by first converting the page content into XML, then using Ontologies to model the data, assigning it semantics and finally carrying out the extraction of the data. The difficulty with this approach is that the description showing how the data is found on a page is created manually, nevertheless, this shows that if we identify how the data is organised on a web page then it is simpler to extract.

Embley [6] concentrates on the car advertisements domain, however, his main goal is to show that Extraction Ontologies can be used to aid semantic understanding and the Semantic Web. Unlike [23], this research uses ML rules over the chosen heuristics, to determine whether a web page is applicable for a given Ontology. The research achieves over 90% for both recall and precision ratios and it is also successful in retargeting the 'car ads' application to other domains such as: mobile phones, restaurants, games etc. However, the Ontology needed or each different domain requires a few dozen person-hours to be updated to fit the description of the new domain. This can quickly turn into a very expensive process.

3.5. Natural Language Processing (NLP)

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

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

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

The above give only a taste of the problems facing NLP techniques. Our research will need to deal with extraction from free text especially when dealing with course descriptions. Many research works have concentrated in this field. These works usually incorporate techniques such as: filtering, lexical analysis of words and phrases to separate the free text into tokens of text, part-of-speech tagging or otherwise known as grammatical tagging which uses algorithms to tag the words in free text as parts of speech e.g. nouns, adjectives, verbs etc.

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

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

The above gave a brief description of the various methods that can be used to deal with WIE. Each approach has its pros and cons, however the question still remaining is: "How does one choose the right approach for a project?" The following section discusses our attempt to test some of the simpler solutions to WIE and the issues encountered on the way.

4. ISSUES AND IRREGULARITIES

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

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

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

Our approach is based on a web page being chosen at random from a predefined website. The pattern recognition process is performed on this page. The resulting wrapper is then applied to all other pages, enhancing the wrapper if necessary so it applies to all pages of that website. The wrappers created for the five websites share many common functions, particularly in finding anchors within each page that serve as starting points for the extraction process, however there are also differences due to the many ways the same concept is expressed in different websites. This shows that the CIE will need to construct a 'generalized' wrapper using rules from each of the existing individual wrappers. Thus a knowledge base will be needed to determine which part of the wrapper to use, and fuzzy logic will be necessary when exact matching is not applicable and the closest match will need to be found instead.

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

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

4.1. Data Formats

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

4.2. Areas of Interest

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

4.3. Tabular Information

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

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

It has also been observed that some cells contain ambiguous information e.g. it is not uncommon for a website to display N/A in a cell or even leave the cell empty. Some tables run over many pages and users need to click somewhere to get to the following set of results. This data is not displayed in the main page's source code, thus it presents a challenge for the extraction process. Furthermore, tables are used not only to display results but also to give structure to other information on a page such as links, ads etc. These should clearly not be extracted. Some research works deal with this issue by only considering for extraction tables with more than 1 row and 1 column. This assumption is not restrictive enough, thus many tables would get extracted unnecessarily giving inaccurate results.

4.4. Logins/Registration

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

4.5. The "Deep Web"

Also known as the Hidden Web, WIE from the Deep Web is a popular topic amongst researchers [14, 18]. One of the reasons behind this popularity is that incredible results have been achieved at extracting information from public sites, however, there is a large number of websites still 'hidden' from the crawler-based Search Engines, which cannot create their indices unless they can 'see' the information on the pages they are crawling. The weakness of Search Engines with the Deep Web data means that over 85% of users who regularly take advantage of Search Engines to locate information [10] will never come in contact with this type of data.

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

4.6. Vague Notation

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

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

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

5. CONCLUSIONS

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

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

One can also decide to choose ‘off-the-shelf’ vs. ‘write-your-own’ code, as an easier and cheaper approach to programming, however, this limits the control over the system, particularly if errors occur or a certain functionality needs to be changed or enhanced. The time then required in finding and fixing problems may be as long as writing the entire code from scratch. For the successful completion of our project, we have chosen the ‘write-your-own’ code method and we believe that we may need a combination of approaches, including Machine Learning and potentially Fuzzy Browsers.

ACKNOWLEDGEMENTS

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

REFERENCES

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

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

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

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

[5] L. Eikvil. Information extraction from world wide web - a survey. Technical Report 945, Norwegian Computing Centre. Nor-way, 1999.

[6] D. Embley. Toward semantic understanding: An approach based on information extraction ontologies. In Proceedings of the 15th Australasian database conference

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

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

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

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

[11] L. Liu, W. Han, D. Buttler, C. Pu, and W. Tang. An xml-based wrapper generator for web information extraction. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 540– [23] 543, 1999.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

APPENDIX G: PAPER 2

Full Reference:

Xhemali, D., Hinde, C.J. & Stone, R.G. (2009). Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. *International Journal of Computer Science Issues*, **4**(1), pp. 16-23.

Keywords: Web Classification, Naïve Bayesian Classifier, Decision Tree Classifier, Neural Network Classifier, Supervised Learning.

Paper type: Journal Paper (Published)

Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages

Daniela XHEMALI[1], Christopher J. HINDE[2] and Roger G. STONE[3]

[1] Computer Science, Loughborough University
Loughborough, Leicestershire, LE11 3TU, UK
D.Xhemali@lboro.ac.uk

[2] Computer Science, Loughborough University
Loughborough, Leicestershire, LE11 3TU, UK
C.J.Hinde@lboro.ac.uk

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

ABSTRACT

Web classification has been attempted through many different technologies. In this study we concentrate on the comparison of Neural Networks (NN), Naïve Bayes (NB) and Decision Tree (DT) classifiers for the automatic analysis and classification of attribute data from training course web pages. We introduce an enhanced NB classifier and run the same data sample through the DT and NN classifiers to determine the success rate of our classifier in the training courses domain. This research shows that our enhanced NB classifier not only outperforms the traditional NB classifier, but also performs similarly as good, if not better, than some more popular, rival techniques. This paper also shows that, overall, our NB classifier is the best choice for the training courses domain, achieving an impressive F-Measure value of over 97%, despite it being trained with fewer samples than any of the classification systems we have encountered.

1 INTRODUCTION

Managing the vast amount of online information and classifying it into what could be relevant to our needs is an important step towards being able to use this information. Thus, it comes as no surprise that the popularity of Web Classification applies not only to the academic needs for continuous knowledge growth, but also to the needs of industry for quick, efficient solutions to information gathering and analysis in maintaining up-to-date information that is critical to the business success.

This research is part of a larger research project in collaboration with an independent brokerage organisation, Apricot Training Management (ATM), which helps other organisations to identify and analyse their training needs and recommend suitable courses for their employees. Currently, the latest prospectuses from different training providers are ordered, catalogued, shelved and the course information found is manually entered into the company's database. This is a time consuming, labour-intensive process, which does not guarantee always up-to-date results, due to the limited life expectancy of some course information such as dates and prices and other limitations in the availability of up-to-date, accurate information on websites and printed literature. The overall project is therefore to automate the process of retrieving, extracting and storing course information into the database guaranteeing it is always kept up-to-date.

The research presented in this paper is related to the information retrieval side of the project, in particular to the automatic analysis and filtering of the retrieved web pages according to their relevance. This classification process is vital to the efficiency of the overall system, as only relevant pages will then be considered by the extraction process, thus drastically reducing processing time & increasing accuracy.

The underlining technique used for our classifier is based on the NB algorithm, due to the independence noticed in the data corpus analysed. The traditional technique is enhanced however, to analyse not only the visible textual content of web pages, but also important web structures such as META data, TITLE and LINK information. Additionally, a 'believed probability' of features in each category is calculated to handle situations when there is little evidence about the data, particularly in the early stages of the classification process. Experiments have shown that our classifier exceeds expectations, achieving an impressive F-Measure value of over 97%.

2 RELATED WORK

Many ideas have emerged over the years on how to achieve quality results from Web Classification systems, thus there are different approaches that can be used to a degree such as Clustering, NB and Bayesian Networks, NNs, DTs, Support Vector Machines (SVMs) etc. We decided to only concentrate on NN, DT and NB classifiers, as they proved more closely applicable to our project. Despite the benefits of other approaches, our research is in collaboration with a small organisation, thus we had to consider the organisation's hardware and software limitations before deciding on a classification technique. SVM and Clustering would be too expensive and processor intensive for the organisation, thus they were considered inappropriate for this project. The following discusses the pros and cons of NB, DTs and NNs, as well as related research works in each field.

2.1 Naïve Bayes Models

NB models are popular in machine learning applications, due to their simplicity in allowing each attribute to contribute towards the final decision equally and independently from the other attributes. This simplicity equates to computational efficiency, which makes NB techniques attractive and suitable for many domains.

However, the very same thing that makes them popular, is also the reason given by some researchers, who consider this approach to be weak. The conditional independence assumption is strong, and makes NB-based systems incapable of using two or more pieces of evidence together, however, used in appropriate domains, they offer quick training, fast data analysis and decision making, as well as straightforward interpretation of test results. There is some research ([13], [26]) trying to relax the conditional independence assumption by introducing latent variables in their tree-shaped or hierarchical NB classifiers. However, a thorough analysis of a large number of training web pages has shown us that the features used in these pages can be independently examined to compute the category for each page. Thus, the domain for our research can easily be analysed using NB classifiers, however, in order to increase the system's accuracy, the classifier has been enhanced as described in section 3.

Enhancing the standard NB rule or using it in collaboration with other techniques has also been attempted by other researchers. Addin et al in [1] coupled a NB classifier with K-Means clustering to simulate damage detection in engineering materials. NBTree in [24] induced a hybrid of NB and DTs by using the Bayes rule to construct the decision tree. Other research works ([5], [23]) have modified their NB classifiers to learn from positive and unlabeled examples. Their assumption is that finding negative examples is very difficult for certain domains, particularly in the medical industry. Finding negative examples for the training courses domain, however, is not at all difficult, thus the above is not an issue for our research.

2.2 Decision Trees

Unlike NB classifiers, DT classifiers can cope with combinations of terms and can produce impressive results for some domains. However, training a DT classifier is quite complex and they can get out of hand with the number of nodes created in some cases. According to [17], with six Boolean attributes there would be need for 18,446,744,073,709,551,616 distinct nodes. Decision trees may be computationally expensive for certain domains, however, they make up for it by offering a genuine simplicity of interpreting models, and helping to consider the most important factors in a dataset first by placing them at the top of the tree.

The researchers in [7], [12], [15] all used DTs to allow for both the structure and the content of each web page to determine the category in which they belong. An accuracy of under 85% accuracy was achieved by all. This idea is very similar to our work, as our classifier also analyses both structure and content. WebClass in [12] was designed to search geographically distributed groups of people, who share common interests. WebClass modifies the standard decision tree approach by associating the tree root node with only the keywords found, depth-one nodes with descriptions and depth-two nodes with the hyperlinks found. The system however, only achieved 73% accuracy. The second version of WebClass ([2]) implemented various classification models such as: Bayes networks, DTs, K-Means clustering and SVMs in order to compare findings of WebClassII. However, findings showed that for increasing feature set sizes, the overall recall fell to just 39.75%.

2.3 Neural Networks

NNs are powerful techniques for representing complex relationships between inputs and outputs. Based on the neural structure of the brain ([17]), NNs are complicated and they can be enormous for certain domains, containing a large number of nodes and synapses. There is research that has managed to convert NNs into sets of rules in order to discover what the NN has learnt ([8], [21]), however, many other works still refer to NNs as a ‘black box’ approach ([18], [19]), due to the difficulty in understanding the decision making process of the NN, which can lead to not knowing if testing has succeeded.

AIRS in [4] used the knowledge acquired during the training of a NN to modify the user’s query, making it possible for the adapted query to retrieve more documents than the original query. However, this process would sometimes give more importance to the knowledge ‘learnt’, thus change the original query until it lost its initial keywords.

Researchers in [6] and [14] proposed a term frequency method to select the feature vectors for the classification of documents using NNs. A much later research ([3]) used NNs together with an SVM for better classification performance. The content of each web page was analysed together with the content of its neighbouring pages. The resulting feature scores were also used by the SVM.

Using two powerful techniques may radically improve classification; however, this research did not combine the techniques to create a more sophisticated one. They simply used them one after the other on the same data set, which meant that the system took much longer to come up with results.

3 NB CLASSIFIER

Our system involves three main stages (Figure 1). In stage-1, a CRAWLER was developed to find and retrieve web pages in a breadth-first search manner, carefully checking each link for format accuracies, duplication and against an automatically updatable rejection list.

In stage-2, a TRAINER was developed to analyse a list of relevant (training pages) and irrelevant links and compute probabilities about the feature-category pairs found. After each training session, features become more strongly associated with the different categories.

The training results were then used by the NB Classifier developed in stage-3, which takes into account the 'knowledge' accumulated during training and uses this to make intelligent decisions when classifying new, unseen-before web pages. The second and third stages have a very important sub-stage in common, the INDEXER. This is responsible for identifying and extracting all suitable features from each web page. The INDEXER also applies rules to reject HTML formatting and features that are ineffective in distinguishing web pages from one-another. This is achieved through sophisticated regular expressions and functions which clean, tokenise and stem the content of each page prior to the classification process. Features that are believed to be too common or too insignificant in distinguishing web pages from one another, otherwise known as stopwords, are also removed. Care is taken, however, to preserve the information extracted from certain Web structures such as the page TITLE, the LINK and META tag information. These are given higher weights than the rest of the text, as we believe that the information given by these structures is more closely related to the central theme of the web page.

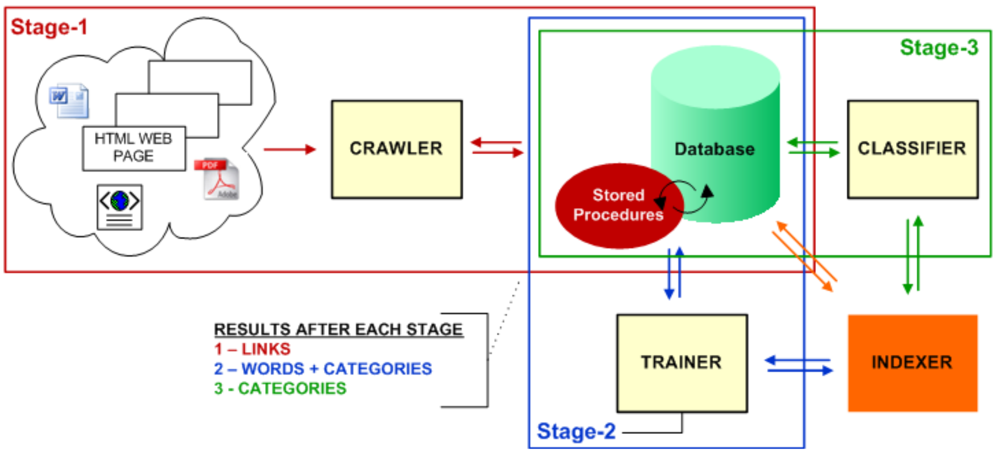


Figure 1. System Stages

The classification algorithm is then applied to each category stored in the database. There are only two categories currently used in this research, relevant and irrelevant, however, the system is designed to work with any number of categories. This is to allow for future growth and potential changes at ATM.

Our classification algorithm is based on the NB approach. The standard Bayes rule is defined as follows:

$$\arg\max_n\{P(C_n|W)\} = \frac{P(W|C_n) \times P(C_n)}{P(W)} \tag{1}$$

where:
P(Cn) = the prior probability of category n
w = the new web page to be classified
P(w|Cn)= the conditional probability of the test page, given category n.

The *P(w)* can be disregarded, because it has the same value regardless of the category for which the calculation is carried out, and as such it will scale the end probabilities by the exact same amount, thus making no difference to the overall calculation. Also, the results of this calculation are going to be used in comparison with each other, rather than as stand-alone probabilities, thus calculating *P(w)* would be unnecessary effort. The Bayes Theorem in (eq. 1) is therefore simplified to:

$$\arg\max_n\{P(C_n|W)\} \propto P(W|C_n) \times P(C_n) \tag{2}$$

The algorithm used in this research is based on the mathematical manipulation of the probabilities of the top 100 most used keywords extracted from each web page. However, the separate probabilities alone would not be sufficient in classifying a new web page. The classification of each page requires the combination of the probabilities of all the separate features {f1, f2, … fi, …, fn} into one probability, as in eq. 3:

$$\arg\max_n\{P(C_n|W)\} \propto \frac{1}{z} P(C_n) \times \prod_{i=1}^{n} P(f_i|C_n) \tag{3}$$

where z is a scaling factor dependent on the feature variables, which helps to normalise the data. This scaling factor was added as we found that when experimenting with smaller datasets, the feature probabilities got very small, which made the page probabilities get close to zero. This made the system unusable.

Once the probabilities for each category have been calculated, the probability values are compared to each other. The category with the highest probability, and within a predefined threshold value, is assigned to the web page being classified. All the features extracted from this page are also paired up with the resulting category and the information in the database is updated to expand the system's knowledge.

Our research adds an additional step to the traditional NB algorithm to handle situations when there is very little evidence about the data, in particular during early stages of the classification. This step calculates a Believed Probability, which helps to calculate more gradual probability values for the data. An initial probability is decided for features with little information about them; in this research the probability value decided is equal to the probability of the page, given no evidence (*P(Cn)*) The calculation followed is as shown below:

$$Believed\ Probability = \frac{\left(bpw \times P(C_n)\right) + \left(\sum_{i=1}^{C_n} nf_i \times P(f_i|C_n)\right)}{bpw + \sum_{i=1}^{C_n} nf_i} \tag{4}$$

where: nf_i is the count of feature f_i in all categories and *bpw* is the believed probability weight (in this case *bpw* = 1, meaning the Believed Probability is weighed the same as one feature).

The above eliminates the assignment of extreme probability values when little evidence exists; instead, much more realistic probability values are achieved, which has improved the overall accuracy of the classification process as shown in the Results section.

4. NB CLASSIFIER

4.1 Performance Measures

The experiments in this research are evaluated using the standard metrics of accuracy, precision, recall and f-measure for Web Classification. These were calculated using the predictive classification table, known as Confusion Matrix (Table 1).

Table 1: Confusion Matrix

		PREDICTED	
		IRRELEVANT	RELEVANT
ACTUAL	IRRELEVANT	**TN**	**FP**
	RELEVANT	**FN**	**TP**

Considering Table 1:

TN (True Negative) - Number of correct predictions that an instance is irrelevant
FP (False Positive) - Number of incorrect predictions that an instance is relevant
FN (False Negative) - Number of incorrect predictions that an instance is irrelevant
TP (True Positive) - Number of correct predictions that an instance is relevant

Accuracy – The proportion of the total number of predictions that were correct:

(5)

Accuracy (%) = (TN + TP) / (TN + FN + FP + TP)

Precision – The proportion of the predicted relevant pages that were correct:

(6)

Precision (%) = TP / (FP + TP)

Recall – The proportion of the relevant pages that were correctly identified

(7)

Recall (%) = TP / (FN + TP)

F-Measure – Derives from precision and recall values:

$$\textbf{F-Measure}\ (\%) = (2 \times \text{Recall} \times \text{Precision}) / (\text{Recall} + \text{Precision}) \tag{8}$$

The F-Measure was used, because despite Precision and Recall being valid metrics in their own right, one can be optimised at the expense of the other ([22]). The F-Measure only produces a high result when Precision and Recall are both balanced, thus this is very significant.

A Receiver Operating Characteristic (ROC) curve analysis was also performed, as it shows the sensitivity (FN classifications) and specificity (FP classifications) of a test. The ROC curve is a comparison of two characteristics: TPR (true positive rate) and FPR (false positive rate). The TPR measures the number of relevant pages that were correctly identified.

$$\textbf{TPR} = TP / (TP + FN) \tag{9}$$

The FPR measures the number of incorrect classifications of relevant pages out of all irrelevant test pages.

$$\textbf{FPR} = FP / (FP + TN) \tag{10}$$

In the ROC space graph, FPR and TPR values form the x and y axes respectively. Each prediction (FPR, TPR) represents one point in the ROC space. There is a diagonal line that connects points with coordinates (0, 0) and (1, 1). This is called the "line of no-discriminations' and all the points along this line are considered to be completely random guesses. The points above the diagonal line indicate good classification results, whereas the points below the line indicate wrong results. The best prediction (i.e. 100% sensitivity and 100% specificity), also known as 'perfect classification', would be at point (0, 1). Points closer to this coordinate show better classification results than other points in the ROC space.

4.2 Data Corpus

In this research, each web page is referred to as a sampling unit. Each sampling unit comprises a maximum of 100 features, which are selected after discarding much of the page content, as explained previously. The total number of unique features examined in the following experiments was 5217. The total number of sampling units used was 9436. These units were separated into two distinct sets: a training set and a test set.

The training set for the NB classifier consisted of 711 randomly selected, positive and negative examples (i.e. relevant and irrelevant sampling units). The test collection created consisted of data obtained from the remaining 8725 sampling units. The training set makes for under 10% of the size of the entire data corpus. This was intentionally decided, in order to really challenge the NB classifier. Compared to many other classification systems encountered, this is the smallest training set used.

4.3 Experimental results

The first experiment that was carried out was to test our enhanced NB classifier against the standard naïve bayes algorithm, in order to determine whether or not the changes made to the original algorithm had enhanced the accuracy of the classifier. For this purpose, we stripped our system of the additional steps and executed both standard and enhanced NB classifiers with the above training and test data. The results showed that the enhanced NB classifier was comfortably in the lead by over 7% in both accuracy and F-Measure value.

In the second set of experiments, the sampling units analysed by the NB classifier were also run by a DT classifier and an NN classifier. The results were compared to determine which classifier is better at analysing attribute data from training web pages. The DT classifier is a ‘C’ program, based on the C4.5 algorithm in [16], written to evaluate data samples and find the main pattern(s) emerging from the data. For example, the DT classifier may conclude that all web pages containing a specific word are all relevant. More complex data samples however, may result in more complex configurations found.

The NN classifier used is also a ‘C’ program, based on the work published in [8]-[11]. MATLAB’s NN toolbox ([20]) could have also been used, however in past experiments MATLAB managed approximately 2 training epochs compared to the ‘C’ NN classifier, which achieved approximately 60,000 epochs in the same timeframe. We, therefore, abandoned MATLAB for the bespoke compiled NN system.

All three classifiers were initially trained with 105 sampling units and tested with a further 521 units, all consisting of a total of 3900 unique features. The NB classifier achieved the highest accuracy (97.89%), precision (99.20%), recall (98.61%) and F-Measure (98.90%) values, however, the DT classifier achieved the fastest execution time. The NN classifier, created with 3900 inputs, 3900 midnodes and 1 output, came last in all metrics and execution time.

For the most recent test, all classifiers were trained with 711 sampling units and they were then tested on the remaining 8725 sampling units. The NB and DT classifiers were adequately fast for exploitation and delivered good discriminations. The test results are shown in Table 2 and Table 3.

Table 2: Confusion Matrix for NB Classifier

		PREDICTED	
		IRRELEVANT	RELEVANT
ACTUAL	IRRELEVANT	TN / 876	FP / 47
	RELEVANT	FN / 372	TP / 7430

Table 3: Confusion Matrix for DT Classifier

		PREDICTED	
		IRRELEVANT	RELEVANT
ACTUAL	IRRELEVANT	TN / 794	FP / 129
	RELEVANT	FN / 320	TP / 7482

The vocabulary used in our experiments, consisting of 5217 features, was initially mapped onto a NN with 5217 inputs, one hidden layer with 5217 nodes and 1 output, in keeping with the standard configuration of a NN, where the number of midnodes is the same as the number of inputs. A fully connected network of this size would have over 27 million connections, with each connection involving weight parameters to be learnt. Our attempt at creating such a network resulted in the NN program failing to allocate the needed memory and crashing.

After paying more attention to the function complexity, we decided to change the number of midnodes to reflect this complexity. We, therefore, created a NN with 5217 inputs, 1 output and only 200 midnodes. This worked well and the resulting NN successfully established all connections. However, we realised that the NN would need to be extended (more nodes and midnodes created) to model any additional, new features, each time they are extracted from future web pages. This would potentially take the NN back to the situation where it fails to make all the required connections and this would be an unacceptable result for ATM. Technology exists for growing nodes; however, this would be complex and expensive. Furthermore, the NN took 200 minutes to train, which is much longer than the other classifiers, which took seconds for the same training sample. Therefore, we decided not to proceed with NNs any further, as they would be unsuitable for our project and other projects of this kind.

Table 4: Final Results

Classifier	Accuracy	Precision	Recall	F-Measure
NB Classifier	**95.20%**	**99.37%**	95.23%	**97.26%**
DT Classifier	94.85%	98.31%	**95.90%**	97.09%

Table 4 shows the Accuracy, Precision, Recall and F-Measure results achieved by the NB and DT classifiers, following the calculations in section 4.1. These results show that both classifiers achieve impressive results in the classification of attribute data in the training courses domain. The DT classifier outperforms the NB classifier in execution speed and Recall value (by 0.67%). However, the NB classifier achieves higher Accuracy, Precision and most importantly, overall F-Measure value, which is a very promising result.

This result is further confirmed by the comparison of the two classifiers on the ROC space (Figure 2.), where it is shown that the result set from the NB classifier falls closer to the 'perfect classification' point than the result set from the DT classifier.

Table 5: ROC Space Results

Classifier	FPR	TPR
NB Classifier	0.05092	0.95232
DT Classifier	0.13976	0.95899

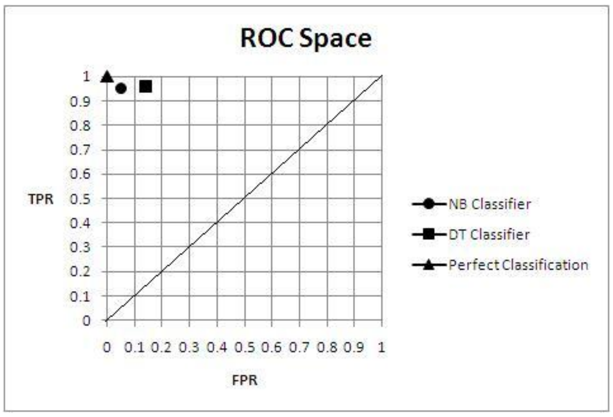


Figure 1. ROC Space

The ROC space was created using the values in Table 5, following the calculations in equations (9) and (10).

6. CONCLUSIONS

To summarise, we succeeded in building a NB Classifier that can classify training web pages with 95.20% accuracy and an F-Measure value of over 97%. The NB approach was chosen as thorough analysis of many web pages showed independence amongst the features used. This approach was also a practical choice, because ATM, like many small companies, has limited hardware specifications available at their premises, which needed to be taken into account.

The NB approach was enhanced, however, to calculate the believed probability of features in each category. This additional step was added to handle situations when there is little evidence about the data, in particular during early stages of the classification process. Furthermore, the classification process was enhanced by taking into consideration not only the content of each web page, but also various important structures such as the page TITLE, META data and LINK information. Experiments showed that our enhancements improved the classifier by over 7% in accuracy, in comparison with the original naïve bayes algorithm.

The NB classifier was tested against 8725 sampling units after being trained with only 711 units. This exact same sample was also analysed by a DT and a NN classifier and the results from all systems were compared to one-another. Our experiments showed that although some NN classifiers can be very accurate for some domains, they take the longest to train and have extensibility issues due to their extremely large and complex nature. It was therefore realized that NNs would be too expensive for ATM and unsuitable for handling a potentially large number of features created by the classification process.

On a more positive note, our experiments produced exciting findings for the application of the NB algorithm in the training courses domain, as the NB classifier achieved impressive results, including the highest Precision value (99.37%) and F-Measure (97.26%). Although some of the results are close to the results from the DT classifier, these experiments show that Naïve Bayes Classifiers should not be considered inferior to more complex techniques such as Decision Trees or Neural Networks. They are fast, consistent, easy to maintain and accurate in the classification of attribute data, such as the training courses domain. In one of our previous papers ([25]), we expressed our concern that many researchers go straight for the more complex approaches without trying out the simpler ones first. We hope this paper will encourage researchers to exploit the simpler techniques, as they can be, as this paper showed, more efficient and much less expensive.

The system may be improved further by reducing the number of features analysed. More research needs to be done to establish a possible cut off point for the extracted features. This may speed up the classification process as well as potentially improve the classifier further. More tests will also be done to confirm the NB classifier's success on a grander scale. In conclusion, this research has shown that the NB approach, enhanced to perform even with limited information, whilst analysing both web content and structural information, gives very promising results in the training courses domain, outperforming powerful and popular rivals such as decision trees and neural networks.

ACKNOWLEDGEMENTS

We would like to thank the whole team at ATM for the support and help they have offered us since the first day of the project. Also, thank you to both ATM and the Centre for Innovative and Collaborative Engineering (CICE) for funding our work.

REFERENCES

[1] Addin, O., Sapuan, S. M., Mahdi, E., & Othman, M. “A Naïve-Bayes classifier for damage detection in engineering materials”, Materials and Design, 2007, pp. 2379-2386.

[2] Ceci, M., & Malerba, D. “Hierarchical Classification of HTML Documents with WebClassII”, Lecture Notes in Computer Science, 2003, pp. 57-72.

[3] Chau, M., & Chen, H. “A machine learning approach to web page filtering using content and structure analysis”, Decision Support Systems, Vol. 44, No. 2, 2007, pp. 482-494.

[4] Crestani, F. “An Adaptive Information Retrieval System Based on Neural Networks”, in: International Workshop on Artificial Neural Networks: New Trends in Neural Computation, Vol. 686, 1993, pp. 732-737.

[5] Denis, F., Laurent, A., Gilleron, R., Tommasi, M. “Text classification and co-training from positive and unlabeled examples”, in: ICML Workshop: The Continuum from Labeled to Unlabeled Data, 2003, pp. 80-87.

[6] Enhong, C., Shangfei, W., Zhenya, Z. & W. Xufa. “Document classification with CC4 neural network”, in: Proceedings of ICONIP, Shanghai, China, 2001.

[7] Estruch, V., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. “Web Categorisation Using Distance-Based Decision Trees”, in: International Workshop on Automated Specification and Verification of Web Site, 2006, pp. 35-40.

[8] Fletcher, G.P & Hinde, C.J. “Interpretation of Neural Networks as Boolean Transfer Functions”, Knowledge-Based Systems, Vol. 7, No. 3, 1994, 207-214.

[9] Fletcher, G.P & Hinde, C.J. “Using Neural Networks as a Tool for Constructing Rule Based Systems”, Knowledge-Based Systems, Vol. 8, No. 4, 1995, 183-189.

[10] Fletcher, G.P & Hinde, C.J. “Producing Evidence for the Hypotheses of Large Neural Networks”, Neurocomputing, Vol. 10, 1996, pp. 359-373.

[11] Hinde, C.J., & Fletcher, G.P., West, A.A. & Williams, D.J. “Neural Networks”, ICL Systems Journal, Vol. 11, No. 2, 1997, pp. 244-278.

[12] Hu, W., Chang, K. & Ritter, G. “WebClass: Web Document Classification Using Modified Decision Trees”, in: 38th Annual Southeast Regional Conference, 2000, pp. 262-263.
[13] Langseth, H. & Nielsen, T. “Classification using Hierarchical Naïve Bayes models”, Machine Learning, Vol. 63, No. 2, 2006, pp. 135-159.
[14] Liu, Z. & Zhang, Y. “A competitive neural network approach to web-page categorization”, International Journal of Uncertainty, Fuzziness & Knowledge Systems, Vol. 9, 2001, pp. 731-741.
[15] Orallo, J. “Extending Decision Trees for Web Categorisation”, in: 2nd Annual Conference of the ICT for EU-India Cross Cultural Dissemination, 2005.
[16] Quinlan, J. R. “Improved use of continuous attributes in C4.5”, Journal of Artificial Intelligence Research, Vol. 4, 1996, pp. 77-90.
[17] Russell, S. & Norvig, P. Artificial Intelligence: A Modern Approach, London: Prentice Hall, 2003.
[18] Segaran, T. Programming Collective Intelligence, U.S.A: O’Reilly Media Inc, 2007.
[19] Tal, B. “Neural Network - Based System of Leading Indicators”, CIBC World Markets, 2003.
[20] TheMathsWork, http://www.mathworks.com/products/neuralnet/
[21] Towell, G. & Shavlik, J. “Extracting Refined Rules from Knowledge-Based Neural Networks”, Machine Learning, Vol. 13, No. 1, 1993, pp. 71-101.
[22] Turney, P. “Learning to Extract Keyphrases from Text”, Technical Report ERB-1057, Institute for Information Technology, National Research Council of Canada, 1999.
[23] Wang C., Ding C., Meraz R., Holbrook S. “PSoL: a positive sample only learning algorithm for finding non-coding RNA genes”, Bioinformatics, Vol. 22, No. 21, 2006, pp. 2590-2596.
[24] Wang, L., Li, X., Cao, C. & Yuan, S. “Combining decision tree and Naïve Bayes for classification”, Knowledge Based Systems, Vol. 19, 2006, pp. 511-515.
[25] Xhemali, D., Hinde, C.J. & Stone, R.G. 2007. “Embarking on a Web Extraction Project”, in: The 2007 UK Conference on Computational Intelligence, 2007.
[26] Zhang, N. “Hierarchical latent class models for cluster analysis”, in: 18th National Conference on Artificial Intelligence, 2002, pp. 230-237.

Daniela Xhemali is an Engineering Doctorate (EngD) student at Loughborough University, UK. She received a First Class (Honours) BSc in Software Engineering from Sheffield Hallam University in 2005 and an MSc with Distinction in Engineering, Innovation and Management from Loughborough University in 2008. Daniela Xhemali has also worked in industry for two years as a Software Engineer, programming multi-user, object oriented applications, with large database backend. Her current research focuses on Web Information Retrieval and Extraction, specifically on the use of Bayes Networks, Decision Trees and Neural Networks in the classification of web pages as well as the use of Genetic Programming and Evolution in the extraction of web information.

Dr. Christopher J. Hinde is a Senior Lecturer at Loughborough University. He is the Programme Director of the Computer Science & Artificial Intelligence group as well as the Programme Director of the Computer Science & E-business group. Dr. Hinde is also the leader of the Intelligent and Interactive Systems Research division. His research interests include: Artificial intelligence, fuzzy reasoning, logic programming, natural language processing, neural nets etc.

Dr. Roger G. Stone is a lecturer at Loughborough University. He is DANS Coordinator and the Quality Manager at Loughborough University. Dr. Stone is also a member of the Interdisciplinary Computing Research Division. His research interests include: Web programming, web accessibility, program specification techniques, software engineering tools, compiling etc

APPENDIX H: PAPER 3

Full Reference:

Xhemali, D., Hinde, C.J. & Stone, R.G. (2010-a). Genetic Evolution of Regular Expressions for the Automated Extraction of Course Names from the Web. P*roceedings of the 2010 International Conference on Genetic and Evolutionary Methods.* Las Vegas, U.S.A, pp. 118-124

Keywords: Web Extraction, Genetic Programming, XML, Regular Expressions, Naïve Bayes Networks.

Paper Type: Conference Paper (Published)

Genetic Evolution of Regular Expressions for the Automated Extraction of Course Names from the Web

Daniela Xhemali[1], Christopher J. Hinde[1], Roger G. Stone[1]
[1]computer Science, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK

ABSTRACT

There are a variety of technologies that attempt to automate the extraction of relevant information from the Web into more manageable data stores such as databases or excel spreadsheets. Such technologies range from basic HTML structure analysis to more complex techniques that attempt to 'reason' about the content and/or structures of web pages. One area of research that is still relatively untouched, relates to the use of genetic evolution in the automation of web extraction. This paper describes an innovative approach to automating web extraction through genetically evolved regular expressions (REs). Particularly, we focus on the extraction of online information related to training course names/titles. The evolved REs are evaluated using a Naïve Bayes Network. Successful or 'fit' extractions correlate to accurate course names being located and pulled out into a database. Experiments show an accuracy of over 94% in the extraction of course names from 300 web pages.

1. INTRODUCTION

There is currently a tremendous amount of information on the Web, due to more and more individuals and companies uploading their data online. According to a study by Maurice de Kunder at the Tilburg University [8], there exist at least 19 billion web pages on the Web (as of February 23, 2010). Some of this information can be duplicated and some of it can even be misleading and inaccurate, however, much of the online information can be very useful in understanding what individuals and organisations have to offer. For example, most training providers now advertise their training courses online. Extracting this 'knowledge' accurately is the key to being able to use this information to suit our needs.

Online information is by no means standardised; different web pages may use different formats and languages to display similar information. The ever growing size of the Web and its changing nature makes it close to impossible to take advantage of all this information without the aid of automated tools. This is why Web Extraction is a popular research topic amongst academics and industry.

There are different technologies that attempt to automate the extraction of information from the Web into more manageable data stores, such as databases or excel spreadsheets. These include: HTML wrapper generators, which rely on finding patterns in the source code of web pages, such as: HTML tags, font differences etc.; XML wrapper generators, which convert the HTML code to a more structured format first (XML) and then apply the extraction rules to the XML document; Machine Learning techniques, which aim to inject some intelligence into the decision making process, resulting in extraction rules that are much less rigid than in the previous two techniques; Extraction Ontologies, which are able to reason about relations in the data, thus handling similar structures well; Natural Language Processing, which attempts to 'understand' the content of the web page through complex grammatical rules. For details on these technologies, please refer to [36].

Another approach that could be used to automate web extraction is Genetic Programming (GP). GP research has attracted attention in fields such as: game strategies [20], military defence [19], plant biology [9], electronics [27], railway platform allocation [6], spam filtering [7], feature extraction from media files [16], [21] etc., however, very few researchers have investigated using GP to automate information extraction from the Web.
This paper introduces an innovative approach to Web Information Extraction (WIE), discussing and explaining the thought process behind our decisions. Our goal is to genetically evolve regular expressions to extract training course names/titles from the Web. This research is in collaboration with an independent brokerage – Apricot Training Management (ATM) – which helps organisations to identify and analyse their training needs and recommend suitable courses for their employees. ATM currently uses time-consuming, labour-intensive, manual techniques to gather information related to training courses, however, this process often results in out-of-date courses being stored in the company's database and many hours being wasted on Web browsing and data entry. Our overall project is to provide ATM with a system that will automate the extraction and storing of training course information into the company's database, guaranteeing always up-to-date training data.

2. RELATED RESEARCH

This paper investigates the use of GP in the extraction of web information. Other general WIE technologies mentioned above such as Ontologies, Natural Language Processing etc., are covered in [36], thus they will not be covered again here.

One can use different terminologies to define GP, but fundamentally: "At the most abstract level, GP is a systematic, domain independent method for getting computers to automatically solve problems starting from a high-level statement of what needs to be done" [25].

GP was inspired by the biological evolution process and Darwin's theory of evolution through natural selection. Fogel [12] and Holland [16] were amongst the first pioneers who foresaw how nature's evolutionary qualities could be applied to AI systems. Fogel's work included predicting prime numbers, predicting symbols (in the latter, the evolutionary process consistently outperformed humans [12]); whilst Holland managed to demonstrate that a population of fixed length strings could be genetically reproduced. Their work was carried forward by many other researchers, however, it would be nearly impossible to summarise everyone's GP research and achievements in this paper, thus we recommend the reader to look at [10], [11] and [1] for a large selection of GP papers and surveys of the history of evolutionary computing. The rest of this section concentrates on GP research in the area of WIE, as this is more closely related to our project.

To the best of our knowledge, there has only been one other investigation in the automation of information extraction from online sources through genetically evolved REs [2], which makes this research distinctive. Barrero et al. [2] concentrate on the genetic evolution of REs for the extraction of information such as URLs and phone numbers. They use a multi-agent system (MAS), where all agents guide the evolutionary process by sharing a training set composed of positive and negative examples. Each agent manages the evolution of a population of fixed-length chromosomes. Populations of different agents can influence each other by means of migration, thus the final genotypes from which REs are created are of variable length. The basic REs created are then combined through subsets of RE operators such as "|, (,), +, ?" to create more complex REs. However, the combination rules are basic, such as "RE1+RE2?", "RE1|RE2", "RE1RE2" and "(RE2)+|(RE1)+" and according to their experiments, the composed REs always end up following the "RE1|RE2" rule, making this additional system stage redundant.

Other research work discussed in this section are important to mention as they cover areas that are similar to WIE to an extent, or to specific parts of our research. For example, a few researchers have concentrated on evolving REs for various reasons. Heddad et al. [15] generated regular expression-based classifiers to locate motifs in DNA sequences that had been manually discovered in labs. Langdon [23], [24] used GP to evolve REs to predict probe quality after observing that the DNA sequences that form the probes, can indeed indicate poor performance in the probes. Continuing in a similar environment, Langdon et al. [26] carried on with the evolution of REs, this time for the classification of gene exons. The REs were represented in Backus-Naur form (BNF). This was done in order to verify the validity of the REs to be evolved. We use a variation of the BNF idea in our research, whereby rules are created to determine the way the different genes are combined to create the appropriate chromosomes. This follows a similar idea by Withall et al. [35], which will be discussed in section 4.2. Cetinkaya [5] also used the BNF syntax to verify a subset of POSIX regular expressions before evolution took place. This research used a rather rigid Fitness Test whereby a multiline text file, containing all the possible outcomes of the regular expressions, was specified before the process began. An optimal RE was one that matched all the lines in the file provided. This technique would be unsuitable for our research, because the information we need to extract may be entirely different in different web pages, thus it is impossible to create a file that would have a thorough list of all the pieces of information required from the extraction process. Conrad's [7] involvement with RE evolution is for the purpose of detecting spam emails. Our research is similar in that we both use linear GP, however, unlike Conrad, we separate Genotypes from Phenotypes (see section 3.2).

Some research has been done in using GP to optimise the finding of textual relations and feature selection to aid information retrieval. Bergstrom et al. [3] investigated GP to aid the automatic extraction of relations between text tokens in order to allow small screens, such as mobile phones or PDAs, to display information without too much stress on the user interface. The training time, however, was extensive and their experiments took at least 90 hours to evolve the required individuals. Some researchers [18], [4] have worked on using GP as well as other well known AI techniques (in this case Support Vector Machines) to calculate fitness scores. For our research, we have developed a Naïve Bayes Network to help with the fitness scores (see section 3.5).

3. REGULAR EXPRESSIONS (REs)

REs [33], [13], [30], [32] are powerful tools used to detect patterns in data. They can range from basic to very complex, matching from just literal text to very specific instances of text based on certain criteria. For example: ^[A-Z][a-z]+ matches all instances that begin (^) with an uppercase letter ([A-Z]) followed by one or more (+) lowercase letters ([a-z]) such as "Regular" or "Expression" but not "RE" or "REs".

REs are very well known, particularly in the UNIX community and they feature largely in some programming languages such as Perl, PHP or AWK. However, the manual generation of REs can be a difficult, error-prone and time consuming undertaking, especially for complex patterns. Tools have been developed to evaluate the validity of REs [14], [31], however, very little, if anything has been done towards the automatic generation of REs.

Our approach proposes the automatic generation of REs through the use of genetic programming principles in order to automate web extraction. The following section discusses the GP side of our approach, justifying the choices made throughout.

4. GENETIC EVOLUTION OF REs (GERE)

GERE takes as input the links considered for extraction, as well as an initial population of REs to serve as seeds to the GP process and outputs the course names as well as the evolved REs, responsible for the successful extraction of this information. The latter is very important because the same RE will be used against the same web page for as long as it is successful. The failure of the RE to extract the correct information from that page[11] triggers GERE to execute in an attempt to produce solutions that work on the web page.

4.1 GP Representation

There are two main representations in GP: the Tree-Based GP and the Linear GP [35]. The Tree-Based GP represents programs as syntax trees, where program variables and constants make up the tree leaves or terminals, whereas the program operators (e.g. *, +, –, /) are the internal nodes or functions. For complex programs (e.g. programs with sub routines), the tree representation can get very complex with the main tree containing many sub trees.

Linear GP represents programs as a linear sequence of instructions. Based on biological evolution, the whole set of instructions is known as the chromosome and each individual instruction is called a gene. Linear GP representations may have either fixed length chromosomes i.e. the same number of genes for every program, or variable ones. This depends on the problem to be solved.

We have chosen the linear approach for our research, because it is more appropriate for the representation of non-standard models like REs. Furthermore, linear GP runs faster than tree-based GP [29]. This is because almost all computer architectures represent computer programs in a linear fashion – as a sequence of commands to execute. The execution of tree-shaped programs is not natural for computers, thus interpreters or compilers would need to be used as part of the tree-based GP [29], which is computationally expensive.

4.2 Genotype to Phenotype Mapping

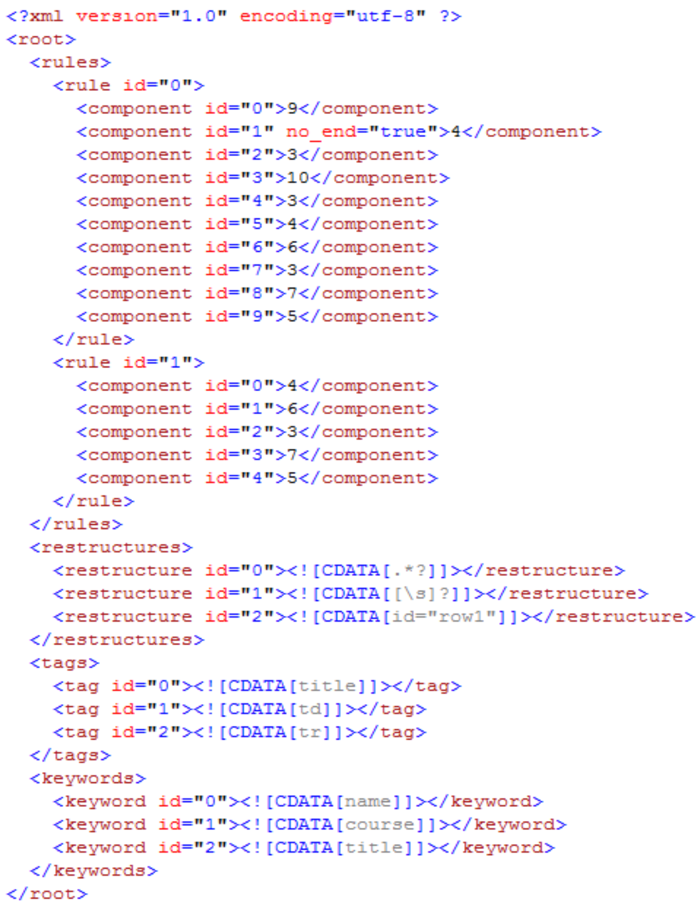
Another point to consider regarding representation is the way individuals of a population are represented. These individuals can either remain the same throughout the process of GP, or they can be encoded to a form, known as the ‘genotype’, which allows for simpler reproduction and then decoded back to the corresponding program, referred to as the ‘phenotype’. We prefer the latter to the former and have chosen it for our research because genotype-phenotype representation allows for a string of numbers to be used to represent the

[11] This may happen if the site format or structure gets updated

genes, which simplifies the process of reproduction as numbers can be manipulated much more easily than strings.

It is very important for the genotypes and phenotypes to have a direct relationship between them, in order for essential characteristics to be preserved from the parents and inherited by the offspring [35]. Koza [22] did not encounter this problem, because the genotypes were not separated from the phenotypes in his work, however, representations such as grammatical evolution [28] could be in danger of creating offspring that do not inherit important qualities from their parents, due to a lack of direct mapping between the parent and offspring phenotypes.

Our genotype to phenotype mapping is similar to the work of Withall et al. [35], whereby the genotypes are used for the genetic manipulation, whereas the phenotypes are used for the Fitness Test. One of the differences, however, is that Withall et al. use fixed-length gene blocks, whereas GERE uses variable-length gene blocks. This is because different REs may contain a varying number of components (i.e. tags, RE structures, keywords etc.). One can create a regular expression that is a centimetre long or one that covers a whole A4 page. Withall et al., however, deal with the evolution of structures such as If – Else – Then, For loops etc., which are more rigid when it comes to the number of components they contain.



```xml
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <rules>
    <rule id="0">
      <component id="0">9</component>
      <component id="1" no_end="true">4</component>
      <component id="2">3</component>
      <component id="3">10</component>
      <component id="4">3</component>
      <component id="5">4</component>
      <component id="6">6</component>
      <component id="7">3</component>
      <component id="8">7</component>
      <component id="9">5</component>
    </rule>
    <rule id="1">
      <component id="0">4</component>
      <component id="1">6</component>
      <component id="2">3</component>
      <component id="3">7</component>
      <component id="4">5</component>
    </rule>
  </rules>
  <restructures>
    <restructure id="0"><![CDATA[.*?]]></restructure>
    <restructure id="1"><![CDATA[[\s]?]]></restructure>
    <restructure id="2"><![CDATA[id="row1"]]></restructure>
  </restructures>
  <tags>
    <tag id="0"><![CDATA[title]]></tag>
    <tag id="1"><![CDATA[td]]></tag>
    <tag id="2"><![CDATA[tr]]></tag>
  </tags>
  <keywords>
    <keyword id="0"><![CDATA[name]]></keyword>
    <keyword id="1"><![CDATA[course]]></keyword>
    <keyword id="2"><![CDATA[title]]></keyword>
  </keywords>
</root>
```

Fig. 1. Sample of XML Rules

A unique attribute of GERE is that it achieves the genotype to phenotype conversion through XML rules (Fig. 1). This technique has many advantages including: improved readability, compatibility with many programming languages, portability and extendibility (XML is not restricted to a limited set of keywords defined by the proprietary vendors, which aids the process of creating rules of different levels of complexity). The initial XML rules were created manually after an extensive analysis of a number of web pages. However, in the future, new rules will be able to be added automatically (Hinde, Stone and Siau are currently working towards implementing this functionality).

The following explains the way XML is used to guide GERE in the mapping of genotypes to phenotypes. Note that this is not a character by character evolution, because this would increase the search space and dramatically increase the execution time. Instead REs are divided into components belonging to three different collections: HTML tags (e.g. "title", "tr" etc.), keywords (e.g. "course", "title" etc.) and RE substructures (e.g. ".*?" or "[\s]?" etc.). These components are incorporated in the rules as <component ... > Component_No </component> structures (Fig. 1) e.g. component 4 corresponds to an "Opening Tag", component 5 to a "Closing Tag" etc.

The XML rules determine the RE components and the order in which they are to be chosen by GERE. The first gene in the genotype is always associated with the rule choice. The remaining string of numbers in the genotype maps to the different components within the chosen rule using the modulo function, e.g. if the rule specifies that the next component to be used is an Opening Tag and there are three tags available in the tag collection and the gene number is 6, the tag chosen will be "title", as 6 mod 3 = 0 (Fig. 1). The mapping process continues until all the genes are mapped to their corresponding RE components.

Regarding tags, the rules can also determine whether or not a Closing Tag is needed (this is useful in cases when the specification of the closing tag extracts different results to the ones needed). This is achieved by including "no_end="true"" in the rule (Fig. 1). Whenever the genotype has fewer genes than the XML rule requires, GERE uses all the available genes from the genotype as described above, then scrutinises the phenotype created in order to guarantee the validity of the RE .

The GP steps in our research follow a straightforward algorithm as shown below:

Pseudo-code: GP Algorithm

```
P = Initial population of regular expressions
F = Apply Fitness Test to P
Loop through to the (predefined) maximum number of generations
Loop through P until offspring population is the same size as the parent population
   Select two parents from P for breeding
   O = Produce two offspring
   M = Mutate one gene from each offspring
End Loop
P = Map offspring genotypes to the corresponding phenotypes.
F = Apply Fitness Test to P
If perfect solution is found
   Stop and exit the evolutionary system
End Loop
```

4.3 Parent Selection

Selecting fit parents for reproduction will increase the chances of the offspring also having good qualities.
The selection method in this research is the Tournament Selection. Tournament Selection is successful in giving all chromosomes a chance to be selected, slightly leaning towards the ones with better fitness scores. It is also immune from any constant offset in the fitness evaluation. All this avoids premature convergence, whilst still selecting good quality parents. The process is as follows:

Pseudo-code: Tournament Selection Process

```
1. C = Tournament-Size (20% of the population size)
2. Randomly select C individuals from population
3. Select the chromosome with the highest fitness score from the C individuals chosen
   previously.
4. Repeat steps 2 and 3 once more to choose the second parent chromosome.
```

4.4 Reproduction

Reproduction is significant in creating new generations of solutions, which would ideally inherit the useful characteristics from the parent population and discard the ineffective ones. In our research, the offspring created may be of variable length, depending on the lengths of their parent chromosomes. Reproduction is aided by two main operators: crossover and mutation, described below:

Crossover

Crossover involves combing genes from two or more parents (normally two) to create one or more offspring. Depending on the number of crossover points selected in both parent chromosomes, we can have single-point crossover, two-point crossover, multiple-point crossover or uniform crossover (these are the four most popular techniques; however there are others, such as half-uniform crossover, edge recombination etc.).
This research has chosen the uniform crossover [33], which involves all the genes in both parent chromosomes getting swapped based on 50% probability. Two offspring are generated each time two parents reproduce. The second offspring is the exact opposite of the first one, created with the ‘left over’ genes.

Mutation

Mutation usually follows the crossover process, although one may choose to perform mutation prior to crossover. The goal of mutation is to help to break out of local optima. There is no guarantee, however, that mutation will create a better chromosome, due to mutation occurring at random.
In this research, each offspring is mutated after the crossover process. Mutation is rare however, with only one gene in each offspring genotype being manipulated. The offspring are mutated instead of the parents, which simplifies the process because, this way, the parent population does not need to be reset.

4.5 Fitness Test

The fitness test is a very important part of a Genetic program. It is responsible for validating the efficiency of an individual from a population at solving the given problem. Normally, a higher score means greater fitness. These fitness scores are used to aid the selection of parents before reproduction can take place. The highest score can also terminate the evolutionary system, because it means that the perfect individual has been generated.

In this research the fitness test is concerned with the validation of REs, which attempt to extract the course name/title. The title is pure text thus its general format is not helpful. Instead, we found that the analysis of its content provides better answers. For this reason, a Naïve Bayes system was developed that 'learns' through initial training and predicts the validity of the generated REs. The Naïve Bayes approach was chosen due to the independence noticed in the data corpus analysed. Details on our enhanced Naïve Bayes System can be found in [37].

Once an offspring (in this case the evolved RE) has been evaluated as a fit candidate by the fitness test, the parent replacement takes place. The process of selecting the better individuals is known as Elitism. In this research, a parent is only substituted with its offspring if the fitness test has proven that the offspring is fitter than the parent, otherwise the offspring is rejected and the parent is taken forward to the next generation.

5. EXPERIMENTAL RESULTS AND EVALUATION

The experimental evaluation has been divided into two stages. The first stage involves a number of tests to establish the GP parameters necessary for the following stage. The second stage relates to the evolution of REs and their fitness evaluations in the extraction of course names/titles.

The execution time for GERE is not a main criterion in this research, because the end product will be scheduled to run undisturbed in the background, particularly during office closing hours at ATM.

5.1 GERE Parameters – Stage One

Some of the results from the first stage of experiments have been discussed throughout the paper, thus, due to the lack of space, the main results from the first stage of experiments are listed here without further discussion.

Population size: It was observed that a population of ten chromosomes is a good compromise between convergence speed and computational resources.

Tournaments: Based on this population size, experiments showed that the best results were achieved with a tournament size of 40% i.e. 4 genomes, when selecting parents for reproduction.

Fitness Target: The fitness score that terminates the execution of the evolutionary system is one (1).

Crossover & Mutation Rates: Whilst the probability rate (50%) chosen for the Uniform Crossover was never in question, it was discovered that the best choice for mutation was the mutation of one gene per genotype, regardless of the number of genes in that genotype.

5.2 GERE Results – Stage two

The analysis of a number of training web pages showed that the course names/titles are displayed in at least three different areas on the page:

1. As part of the overall web page title
2. As a large heading on the page
3. In a table (there may exist multiple course names in any one table)

GERE was trained with positive and negative examples from 100 mixed web pages. The Naïve Bayes Network [37] was then used to calculate the fitness scores for the evaluated REs for an additional 200 web pages. The fitness scores are also influenced by other determining factors such as: the length of the evolved RE, the length of the extracted data and the efficiency of the RE's components. For example: RE component ".*?.*?" is less efficient than ".*?" despite having the same effect. Table 1 shows some of the REs evolved, the areas of the web pages for which the REs were aiming and the fitness scores assigned to them.

Table 1. Evolved REs

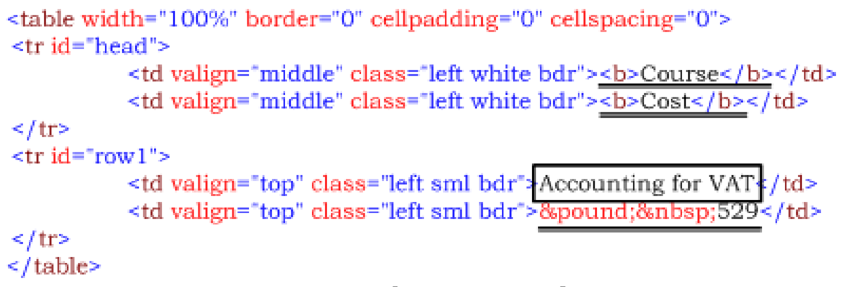
ID	Evolved RE	Target Area	Fitness Score
1	`<title.*?><title.*?>([\s]?.*?[\s]?)</title></title>`	Page Title	0
2	`<title>.*?[\s]?.*?</title>`	Page Title	0.83
3	`(?<=<title>).*?(?=</title>)`	Page Title	0.92
4	`<title.*?>.*?</title>`	Page Title	1
5	`<td.*?><title.*?>([\s]?[\s]?[\s]?)<\/title><\/td>`	Heading	0
6	`<H1.*?>.*?</H1>`	Heading	1
7	`<title.*?><tr[\s]?id="row1".*?>(.*?[\s]?.*?)<\/tr[\s]?id="row1"><\/title>`	Table	0
8	`<tr.*?><td.*?>.*?</td>`	Table	0.56
9	`<tr[\s]?id="row1".*?>[\s]?<td.*?>.*?</td>`	Table	1

Note: All the above REs, apart from RE-3, extract the HTML Tag information together with the data, however, this is not an issue as this information is removed from the data before it is sent for evaluation to the Fitness function.

- RE-2 and RE-4 may look very similar, however, RE-2 has had points deducted for having an inefficient component collection (.*?[\s]?.*?).
- REs 1, 5 and 8 have scored zero, because, although they are valid REs, they did not extract anything from the web pages on which they were tested.
- RE-8 was scored low because it was not specific enough. A small print screen of the web page against which it was tested, is shown in Fig. 2. This clearly shows

that RE-8 extracts all of the underlined information, rather than just the name of the course - “Accounting for VAT”.

- REs 4, 6 and 9 were obtained in the 1st, 2nd and 58th generations respectively
- RE-9 took longer due to its increased complexity.
- In these experiments, the maximum number of generations run per web page was 100.



```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
 <tr id="head">
            <td valign="middle" class="left white bdr"><b>Course</b></td>
            <td valign="middle" class="left white bdr"><b>Cost</b></td>
 </tr>
 <tr id="row1">
            <td valign="top" class="left sml bdr">Accounting for VAT</td>
            <td valign="top" class="left sml bdr">&pound;&nbsp;529</td>
 </tr>
</table>
```

Fig. 2. Web Page Sample

The results of the Naïve Bayes Network were analysed using the standard metrics of accuracy, precision and recall. The Precision and Recall balance was also quantified with the F-Measure. These were described in detail in [37] and will not be repeated here. The Naïve Bayes Network achieved an accuracy of over 94%, a precision of over 97%, a recall value of over 96% and an F-Measure of over 96%.

6. CONCLUSION AND FURTHER WORK

This paper has discussed an innovative approach to automating web extraction through genetically evolved REs, an XML based system for genotype to phenotype conversion and Naïve Bayes fitness scoring. Specifically, it has concentrated on the extraction of course names/titles.

The GP algorithm followed is discussed in this paper together with the different GP steps involved. Preliminary results from experiments are also presented. These results and the rest of the paper have shown that it is possible to genetically evolve REs for the automation of WIE. It has also shown the benefits of using XML rules for the mapping of genotypes to phenotypes and Naïve Bayes for the fitness testing of independent data.

The next stage of the research will concentrate on the evolution of REs for the extraction of other training course details such as prices, dates and locations. Analysis of some preliminary web pages has indicated a partial dependence amongst this data, which may mean using a different approach for the fitness testing.

ACKNOWLEDGEMENTS

We would like to thank the whole team at ATM for their support and help. Also, thank you to ATM, CICE and Loughborough University for funding our work.

REFERENCES

[1] Back, T., Hammel, U. & Schwefel, H-P. 1997. Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation.

[2] Barrero, D., Camacho, D. & R-Moreno, M. 2009. Automatic Web Data Extraction Based on Genetic Algorithms and Regular Expressions. Data Mining and Multi-agent Integration, Springer-Verlag US, 143.

[3] Bergstrom, A., Jaksetic, P. & Nordin, P. 2000. Enhancing Information Retrieval by Automatic Acquisition of textual Relations using Genetic Programming. Proceedings of the 5th International Conference on Intelligent User Interfaces, 29-32.

[4] Braga, P., Oliveira, A. & Meira, S. 2008. A GA-based feature selection and parameters optimization for support vector regression applied to software effort estimation. Proceedings of the 2008 ACM Symposium on Applied Computing, 1788-1792.

[5] Cetinkaya, A. 2007. Regular expression generation through grammatical evolution. Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation (London, UK). Workshop Session, 2643-2646.

[6] Clarke, M., Hinde, C.J., Whithall, M.S., Jackson, T.W., Phillips, I.W., Brown, S & Watson, R. 2009. Allocating Railway Platforms using a Genetic Algorithm. Research and Development in Intelligent Systems XXVI, Springer London, 421-434.

[7] Conrad, E. 2007. Detecting Spam with Genetic Regular Expressions. SANS Institute Reading Room.

[8] De Kunder, M. 2010. The size of the World Wide Web (Tilburg University). Retrieved February 23rd, 2010 from http://www.worldwidewebsize.com

[9] Dyer, J. & Bentley, P. 2002. PLANTWORLD: Population Dynamics in Contrasting Environments. In Cantu-Paz E., GECCO, 122-129.

[10] Fogel, D.B. 1994. An Introduction to Simulated Evolutionary Optimisation. IEEE Transactions on Neural Networks.

[11] Fogel, D.B. 1998. The Fossil Record. IEEE Press

[12] Fogel, L.J., Owens, A.J. & Walsh, M.J. 1966. Artificial Intelligence Through Simulated Evolution. John Wiley & Sons Inc.

[13] Friedl, J. 1997. Mastering Regular Expressions. O'Reilly & Associates (Jan 1997)

[14] Goyvaerts, J. RegexBuddy. http://www.regular-expressions.info/regexbuddy.html

[15] Heddad, A., Brameier, M. & MacCallum, R. 2004. Evolving Regular Expression-based Sequence Classifiers for Protein Nuclear Localisation. In Applications of Evolutionary Computing. Springer, Berlin, 31-40.

[16] Holland, J.H. 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

[17] Hsu, P-H. 2007. Feature extraction of hyperspectral images using wavelet and matching pursuit. ISPRS Journal of Photogrammetry and Remote Sensing. Elsevier Science, Amsterdam, vol. 62 (2), 78-92.

[18] Huang, C-L. & Wang, C-J. 2006. A GA-based feature selection and parameters optimizationfor support vector machines. In Expert Systems with Applications, Elsevier Ltd., vol. 31 (2), 231-240.

[19] Jackson, D. 2005. Evolving Defence Strategies by Genetic Programming. In Lecture Notes in Computer Science. Springer Berlin, Vol. 3447, 281-290.

[20] Keaveney, D. & O’Riordan, C. 2009. Evolving Robust Strategies for an Abstract Real-time Strategy Game. Proceedings of the 5th International Conference on Computational Intelligence and Games. 371-378

[21] Klank, U., Padoy, N., Feussner, H. and Navab, N. 2008. Automatic feature generation in endoscopic images. International Journal of Computer Assisted Radiology and Surgery. Springer, 3, 331-339

[22] Koza, J.R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press.

[23] Langdon, W. 2008. Evolving GeneChip correlation predictors on parallel graphics hardware. In WCCI, Hong Kong. IEEE.

[24] Langdon, W. & Harrison A. 2008. Evolving Regular Expressions for GeneChip Probe Performance Prediction. In Lecture Notes in Computer Science. Springer, Berlin, vol. 5199, 1061-1070.

[25] Langdon, W., Poli, R., McPhee, N. & Koza, J.R. 2008. Genetic Programming: An Introduction and Tutorial with a Survey of Techniques and Applications. In Studies in Computational Intelligence. Springer, Berlin, vol. 115, 927-1028.

[26] Langdon, W., Roswell, J. & Harrison, A. 2009. Creating regular expressions as mRNA motifs with GP to predict human exon splitting. Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (Montreal, Canada), Poster Session, Track 3, 1789 – 1790.

[27] Long, W., Yang, S., Min, Y. & Tong, S. 1997. Level-Oriented GA-based Test Generation of Logic Circuits. In Proceedings of the International Intelligent Processing Systems, vol 1, 563-567.

[28] O’Neill, M., Brabazon T., Ryan, C. & Collins J.J. 2001. Developing a Market Timing System using Grammatical Evolution. Proceedings of GECCO.

[29] Poli, R., Langdon, W. & McPhee. 2008. N. A Field Guide to Genetic Programming.

[30] Regular-Expressions.info 2009. http://www.regular-expressions.info/

[31] Sells, C. 2009. RegexDesigner.NET http://regexdesigner-net.findmysoft.com/

[32] Sun Microsystems. 2008. Lesson: Regular Expressions. http://java.sun.com/docs/books/tutorial/essential/regex/index.html

[33] Syswerda, G. 1989. Uniform Crossover in Genetic Algorithms. Proceedings of the 3rd International Conference on Genetic Algorithms. Morgan Kaufmann.

[34] Thompson, K. 1968. Programming techniques: Regular expression search algorithm. Commun. ACM, Vol 11(6), 419-422

[35] Withall, M.S., Hinde, C.J. & Stone, R.G. 2008. An improved representation for evolving programs. Journal of Genetic Programming and Evolvable Machines. Springer Netherlands, vol. 10(1), 37-70.

[36] Xhemali, D., Hinde, C.J. and Stone, R.G. 2007. Embarking on a Web Information Extraction Project. UKCI Conference on Computational Intelligence.

[37] Xhemali, D., Hinde, C.J. & Stone, R.G. 2009. Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. International Journal of Computer Science Issues, vol. 4(1), 16-23.

APPENDIX I: PAPER 4

Full Reference:

Xhemali, D., Hinde, C.J. & Stone, R.G. (2010-b). Domain-Independent Genotype to Phenotype Mapping through XML Rules. *International Journal of Computer Science Issues.* **7**(3), pp. 1-9.

Keywords: Genetic Evolution, Genotypes, Phenotypes, XML Mapping, Regular Expressions, Complete Software Structures.

Paper Type: Journal Paper (Published)

Domain-Independent Genotype to Phenotype Mapping through XML Rules

Daniela Xhemali[1], Christopher J. Hinde[2] and Roger G. Stone[3]

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

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

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

ABSTRACT

This paper discusses an innovative approach to mapping Genotypes to Phenotypes through XML rules. Specifically, it concentrates on the mapping process using two very different domains – Regular Expressions (REs) and Software Program Statements. The paper shows that our Genotype-Phenotype system can be applied to any domain that requires the use of REs and it can be adapted to work for any other domain with minimum effort.

1. INTRODUCTION

Genetic Programming (GP) research has attracted attention in various fields such as: game strategies [14], military defence [13], plant biology [7], electronics [20], railway platform allocation [4], spam filtering [5], feature extraction from media files [12], [16], automated web extraction [3], [31] etc. One can use different terminologies to define GP, but fundamentally: “At the most abstract level, GP is a systematic, domain independent method for getting computers to automatically solve problems starting from a high-level statement of what needs to be done” [18].

This paper focuses on a specific part of GP – the Genotype-Phenotype Mapping – thus other components of GP will not be covered here. Readers are encouraged to look at [31] for a discussion of the different GP components in relation to a real project, as well as [1], [8] and [9] for a large selection of GP papers and surveys of the history of evolutionary computing.

The Genotype-Phenotype Mapping relates to the way individuals in a population are represented, as this can have a significant effect on the performance of GP. A Genotype represents each individual in the search space, whereas its Phenotype represents the individual in the solution space [2]. Some research, particularly earlier GP research, do not make a distinction between Genotypes and Phenotypes [5], [17], [27]. Individuals in each genetic population remain the same throughout the evolution process. In these works the search space and the solution space are identical.

In 1994, Banzhaf [2] suggested the separation of the two spaces and introduced his work on the Genotype-Phenotype mapping. The separation involves the encoding of the individuals to a form known as the ‘Genotype’, which is later on decoded back to the corresponding program, referred to as the ‘Phenotype’. This separation simplifies and increases the efficiency of certain genetic operations such as: reproduction and mutation, because these would no longer be constrained by the parameters used in the program being evolved. In Genotype-Phenotype based GP, genetic operators such as Crossover and Mutation would be performed on the Genotype, whereas other processes, such as the Fitness scoring, would be performed on the Phenotype. Sections 3.1 and 3.4 explain this concept further through examples.

There are researchers who criticise the separation into Genotypes and Phenotypes [19]. The main concern expressed is that the conversion process of a mutated Genotype into the Phenotype may result in anomalies that could potentially lead to invalid solutions. A direct mapping between the encoded program and the actual program is therefore vital to ensure the validity of the solutions [23], [28].

In our research, the Genotypes are presented as strings of integers. The direct mapping of these integers to the corresponding structures is achieved through an innovative approach involving XML rules as described in section 3.4.

This research is part of a larger project in collaboration with an independent brokerage – Apricot Training Management (ATM) – which helps organisations to identify and analyse their training needs and recommend suitable courses for their employees. ATM currently uses time-consuming, labour-intensive, manual techniques to gather information related to training courses, however, this process often results in out-of-date courses being stored in the company’s database and many hours being wasted on Web browsing and data entry. Our overall project is to provide ATM with a system that will automate the extraction and storing of training course information into the company’s database, guaranteeing always up-to-date training data [29], [30]. Specifically, the research concentrates on the evolution of Regular Expressions (REs) for the extraction of pieces of information such as course names, prices, dates and locations from training web pages.

The following section focuses on research and techniques related to Genotype-Phenotype based GP.

2. RELATED RESEARCH

Many researchers have embraced the separation of the search space from the solution space through the use of Genotypes and Phenotypes [2], [15], [28]. This, however, adds an additional step to the genetic evolution process – the translation or mapping of the Genotypes to their corresponding Phenotypes. This step occurs after the genetic reproduction stage (i.e. the crossover and mutation) and before the Fitness test can take place.

The following concentrates on the different methods that have been used to achieve the mapping process.

Banzhaf [2] represented his Genotypes as linear binary strings. The mapping stage then processed these Genotypes from left to right in 5-bit sections, where each 5-bit code mapped to a pre-specified symbol. For example: 00000 mapped to PLUS, 00100 mapped to POW, 11000 mapped to variable X etc. The first bit indicated whether the code represented a function (PLUS, POW etc.) or a terminal (X, Y etc.). The research also discussed their concern about generating constant numbers. Koza [17] solved this problem by defining “random ephemeral constants” where constants are only generated once for a particular program and then reused wherever they are needed within that program.

Keller [15] continued in the footsteps of Banzhaf, concentrating on providing experimental evidence for choosing the Genotype-Phenotype approach instead of the normal GP approach. Keller’s system however, could only evolve programs in languages defined by the LALR (Look Ahead Look Recursive) grammar, as this was the grammar chosen for the repairing stage of the Genotype-Phenotype mapping process.

There was a certain amount of redundancy in the genetic coding in both Banzhaf’s and Keller’s works. They both admitted that, in their works, different binary strings could correspond to the same symbol, which could lead to inconsistencies e.g. 000 and 100 both mapping to ‘a’.

A slightly different Genotype representation is seen in the work of Withall et al. [28]. In here Genotypes are represented as linear blocks of integers. Each block comprises exactly four integers, each representing a different gene. Although both research works used fixed-length genomes, in [2] the resulting Phenotypes could vary in length, whereas in [28] they remained fixed. The first integer in Withall’s Genotype determines the type of function that follows. Similarly, the Genotype in our research is represented as a string of integers. There are no fixed length genomes determined however; instead the Genotype can contain any number of genes.

The unique feature of our research is the use of XML to define the necessary rules to achieve the Genotype-Phenotype mapping. The first gene in the string determines the XML rule to be followed, which in itself guides the mapping of the rest of the genes into a valid Phenotype. This is explained in detail in section 3.4.

3. GENOTYPE TO PHENOTYPE MAPPING

Before discussing our mapping approach, it is important to explain a few related topics in order for the reader to fully understand how the approach works. The following discusses the main representation chosen for the system, as well as the different domains on which this approach was tested. Specifically, the two domains relate to the Genotype-Phenotype mapping of: REs and Software Program Statements. The Software Program Statements domain was chosen because it is very different from that of REs and thus, it can truly illustrate the extent to which our system needs to be changed to apply to such a domain.

Section 3.4 explores the XML mapping process in detail.

3.1 GP Representation

There are two main representations in GP: the Tree-Based GP and the Linear GP [28]. The Tree-Based GP represents programs as syntax trees (Fig. 1), where program variables and constants make up the tree leaves (x, 1, 5, y, 2), whereas the program operators (*, +, –, /) are the internal nodes. The tree leaves are also known as the terminals, whereas the internal nodes are known as the functions. Fig. 1 shows the tree representation of program "(x+1)*(5-(y/2))". For complex programs, the main tree may contain many sub trees.

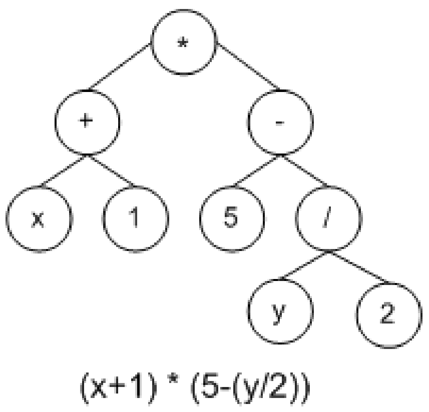


Fig. 1 Tree-based GP Representation

Linear GP represents programs as a linear sequence of instructions (Fig. 2). Based on biological evolution, the sets of instructions are known as Genomes and each individual instruction is called a gene. All the available Genomes for a particular program form the Genotype. Linear GP representations may have either fixed length Genomes i.e. the same number of genes for every instruction, or variable ones. This depends on the problem to be solved.

1st Instruction	2nd Instruction	...	nth Instruction

Fig. 2 Linear GP Representation

We have chosen the linear approach for our research, because it is more appropriate for the representation of non-standard models like REs. Furthermore, linear GP runs faster than tree-based GP [21]. This is because almost all computer architectures represent computer programs in a linear fashion – as a sequence of commands to execute. The execution of tree-shaped programs is not natural for computers, thus interpreters or compilers would need to be used as part of the tree-based GP [21], which is computationally expensive.

3.2 Regular Expressions (REs)

REs [26], [10], [22], [25] are powerful tools used to detect patterns in data. They can range from basic to very complex, matching from just literal text to very specific instances of text based on certain criteria. For example: ^[A-Z][a-z]+ matches all instances that begin (^) with an uppercase letter ([A-Z]) followed by one or more (+) lowercase letters ([a-z]) such as "Regular" or "Expression" but not "RE" or "REs".

REs are very well known, particularly in the UNIX community and they feature largely in some programming languages such as Perl, PHP or AWK. However, the manual generation of REs can be a difficult, error-prone and time consuming undertaking, especially for complex patterns. This is due to the fact that although REs are built up from small building blocks, where each block is fairly simple; all the available blocks can be combined in an infinite number of ways [10], which may result in a highly complex RE. Tools have been developed to evaluate the validity of REs [11], [24], however, very little, if anything has been done towards the automatic generation of REs.

Our research focuses on the automatic generation of REs through the use of genetic programming principles in order to automate web extraction. This paper concentrates on the mapping of Genotypes (strings of meaningless numbers) to Phenotypes (REs) through XML rules as explained in section 3.4.

3.3 Software Program Statements

The research in this paper is based on the work of Withall et al. [28]. The examples are kept as close to the original as possible in order to ensure their integrity. The only difference is that the Software Program Statements in [28] were evolved to be valid in Perl, whereas in this paper their validity is ensured against VB.NET 2008. Fig. 3 shows an example of the syntax differences in both languages.

```
PERL:     if ($x != $y) {
              $z = $z + 1;
          }

VB.NET:   IF x <> y THEN
              z = z + 1
          END IF
```

Fig. 3 PERL vs. VB.NET

The Software Program Statements that are considered in this paper include simple structures that are commonly used in programming such as: FOR … NEXT; IF … THEN … ELSE, WHILE … END WHILE as well as useful statements such as Addition (x = y + z), Subtraction (x = y - z), Multiplication (x = y * z) and Division (x = y / z).

The following section will explain in detail how Genotypes are converted to either REs or Software Program Statements using XML rules.

3.4 XML Mapping – REs

It is very important for the Genotypes and Phenotypes to have a direct relationship between them, in order for essential characteristics to be preserved from the parents and inherited by the offspring [28]. Koza [17] did not encounter this problem, because the Genotypes were not separated from the Phenotypes in his work, however, representations such as grammatical evolution [20] could be in danger of creating offspring that do not inherit important qualities from their parents, due to a lack of direct mapping between the parent and offspring Phenotypes.

Our Genotype to Phenotype mapping is similar to the work of Withall et al. [28], whereby the Genotypes are used for the genetic manipulation, whereas the Phenotypes are used for the Fitness Test. One of the differences, however, is the length of the Genomes. Withall et al. use fixed-length gene blocks or Genomes, where each block comprises four genes, however, they allow for variable-length Genomes through padding – in cases of shorter program structures or statements, left over genes are ignored. In this research, we only use variable-length Genomes. This is because different REs may contain a varying number of components (i.e. tags, RE structures, keywords etc.). One can create a regular expression that is a centimetre long or one that covers a whole A4 page. Withall et al., however, deal with the evolution of structures, as discussed in section 3.3, which are more rigid when it comes to the number of components they contain.

As mentioned previously, a unique attribute of this research is that it achieves the Genotype to Phenotype conversion through XML rules (Fig. 4). This technique has many advantages including: improved readability, compatibility with many programming languages, portability and extendibility (XML is not restricted to a limited set of keywords defined by the proprietary vendors, which aids the process of creating rules of different levels of complexity).

The initial XML rules were created manually after an extensive analysis of a number of web pages. However, in the future, new rules will be able to be added to the XML file automatically (Hinde, Stone and Siau are currently working towards implementing this functionality).

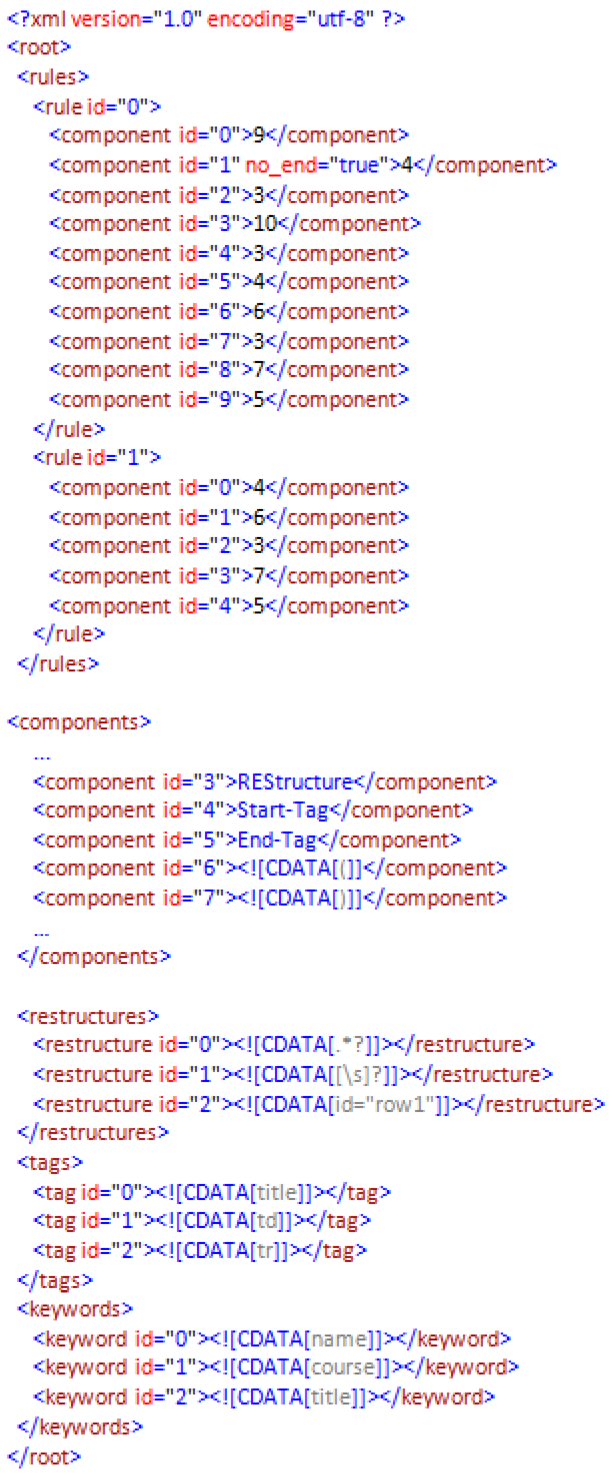
The rest of this section explains the way XML is used to guide the mapping of Genotypes to valid Phenotypes. The Genotype-Phenotype mapping process for the REs domain is shown below:

Pseudo-code: Genotype-Phenotype Mapping Process

```
1) Determine the XML rule to follow
2) Follow the chosen XML rule to the end
3) IF the Genotype has fewer genes than the rule requires
     a. Follow the rule for the number of genes available
     b. Repair outcome to create a valid partial solution.
4) IF the Genotype has enough genes for the XML rule
     a. Follow all the components within the rule
     b. Repair outcome (if necessary) to create a valid and complete solution.
5) IF the Genotype has more genes than the  rule requires
     a. Follow the same steps as above (4a and 4b)
     b. Ignore the rest of the genes in the Genotype
```

Note that this is not a character by character evolution, because this would increase the search space and dramatically increase the execution time. Instead REs are divided into three collections: HTML tags (e.g. “title”, “tr” etc.), keywords (e.g. “course”, “title” etc.) and RE substructures (e.g. “.*?” or “[\s]?” etc.). Each evolved gene will be translated to an element of one of these collections. Each collection is further divided into a number of components e.g. the Start-Tag component determines an opening tag, the End-Tag determines a closing tag, the Start-REStructure determines an opening RE structure, an RE-Structure determines a normal structure that does not need closing etc.

These components are important for ensuring consistency and accuracy in the Phenotypes created. For example once a Start-Tag has been determined, the system automatically knows that it needs to reuse this tag as an End-Tag when told so by the rules.



```xml
<?xml version="1.0" encoding="utf-8" ?>
<root>
 <rules>
   <rule id="0">
     <component id="0">9</component>
     <component id="1" no_end="true">4</component>
     <component id="2">3</component>
     <component id="3">10</component>
     <component id="4">3</component>
     <component id="5">4</component>
     <component id="6">6</component>
     <component id="7">3</component>
     <component id="8">7</component>
     <component id="9">5</component>
   </rule>
   <rule id="1">
     <component id="0">4</component>
     <component id="1">6</component>
     <component id="2">3</component>
     <component id="3">7</component>
     <component id="4">5</component>
   </rule>
 </rules>

<components>
   ...
   <component id="3">REStructure</component>
   <component id="4">Start-Tag</component>
   <component id="5">End-Tag</component>
   <component id="6"><![CDATA[(]]</component>
   <component id="7"><![CDATA[)]]</component>
   ...
 </components>

 <restructures>
   <restructure id="0"><![CDATA[.*?]]></restructure>
   <restructure id="1"><![CDATA[[\s]?]]></restructure>
   <restructure id="2"><![CDATA[id="row1"]]></restructure>
 </restructures>
 <tags>
   <tag id="0"><![CDATA[title]]></tag>
   <tag id="1"><![CDATA[td]]></tag>
   <tag id="2"><![CDATA[tr]]></tag>
 </tags>
 <keywords>
   <keyword id="0"><![CDATA[name]]></keyword>
   <keyword id="1"><![CDATA[course]]></keyword>
   <keyword id="2"><![CDATA[title]]></keyword>
 </keywords>
</root>
```

Fig. 4 Sample of XML Rules for REs

We have also introduced some additional components that are not related to the specified collections e.g. the Start-Capture component translates to the symbol ‘(‘ and indicates the beginning of a capturing group i.e. the part of the RE, which will capture the part of the results needed; the End-Capture component indicates the end of the capturing group etc. These additional components do not use any genes from the Genotype. They were introduced simply to help the mapping process to translate Genotypes into syntactically correct REs. Fig. 4 shows a partial list of the components needed for the Genotype-Phenotype Mapping process and the order in which they are used within the XML rules.

Each XML rule (Fig. 4) determines the necessary components and the order in which they are to be chosen by the mapping stage in order to create a valid and efficient RE. The first gene in the Genotype is always associated with the RE rule choice. The remaining string of integers in the Genotype maps to the different components within that RE rule.

The modulo function is used for this purpose, e.g. Table 1 shows the Genotype to be translated using the information in Fig. 4. The value of the first gene is 5. This represents the RE rule to be used. In this case, there are two different rules in the XML file, so 5 mod 2 = 1, which means that the second rule (id = 1) is chosen. This rule contains five different components. The first component number is 4 (Table 2). This corresponds to the Start-Tag component (Fig. 4), which means the next gene in the Genotype (gene 7) needs to be mapped to one of the tags in the ‘tags’ collection. In this case, the collection has three tags, thus 7 mod 3 = 1 gives the ‘td’ tag. The following component number is 6. This corresponds to the Start-Capture component, which maps to the symbol “(“ without using any genes from the Genotype.

The remaining components are dealt with in the same manner (see Table 2). Note that in this example, only the first three genes of the Genotype were needed; the remaining two are simply ignored.

Table 1: Genotype

5	7	15	22	43

Table 2: Genotype to RE Mapping

Component No.	*Component*	*Gene Used*	*Modulo*	*Translation*
-	-	5	1	Rule 2
4	Start-Tag	7	1	td
6	Start-Capture	-	-	(
3	RE-Structure	15	0	.*?
7	End-Capture	-	-	)
5	End-Tag	-	-	td

<td.*?>(.*?)</td>

Fig. 5 Phenotype

Once all the required genes have been decoded, the resulting RE is repaired to ensure its validity and efficiency. Fig. 5 shows the complete RE (Phenotype) for the above example. The additional symbols added by the repairing function are shown encircled.

The repairing function is an independent function which scrutinises the Phenotype created in order to guarantee the validity of the RE[12]. This function is in charge of tasks like: closing opening tags, closing opening parentheses, adding/removing RE structures in cases when there are fewer genes in the Genotype than required by the XML rule, adding variable declarations at the beginning of the program, adding 'footer' information where necessary e.g. when returning the RE to a calling function etc. All this is achieved through the use of a STACK programming structure, which works in a LIFO (Last In First Out) manner. This is particularly helpful when closing nested tags and RE structures, for example: Tag1 + Tag2 + RE-Structure + *C-RE* + *C-Tag2* + *C-Tag1* (where C stands for Closing. The Italic part shows the part of the RE added by the repairing function.)

The XML rules can also determine whether or not a closing tag is actually needed. This is useful in cases where the inclusion of a closing tag results in different results being extracted to the ones needed. For example, the RE: <tr[\s]?id="row1".*?><td.*?>.*?</td> contains two opening tags ('tr' and 'td'), however, only the 'td' tag needs to be closed for this to work as intended. This is achieved by including no_end="true" in the rule (Fig. 4).

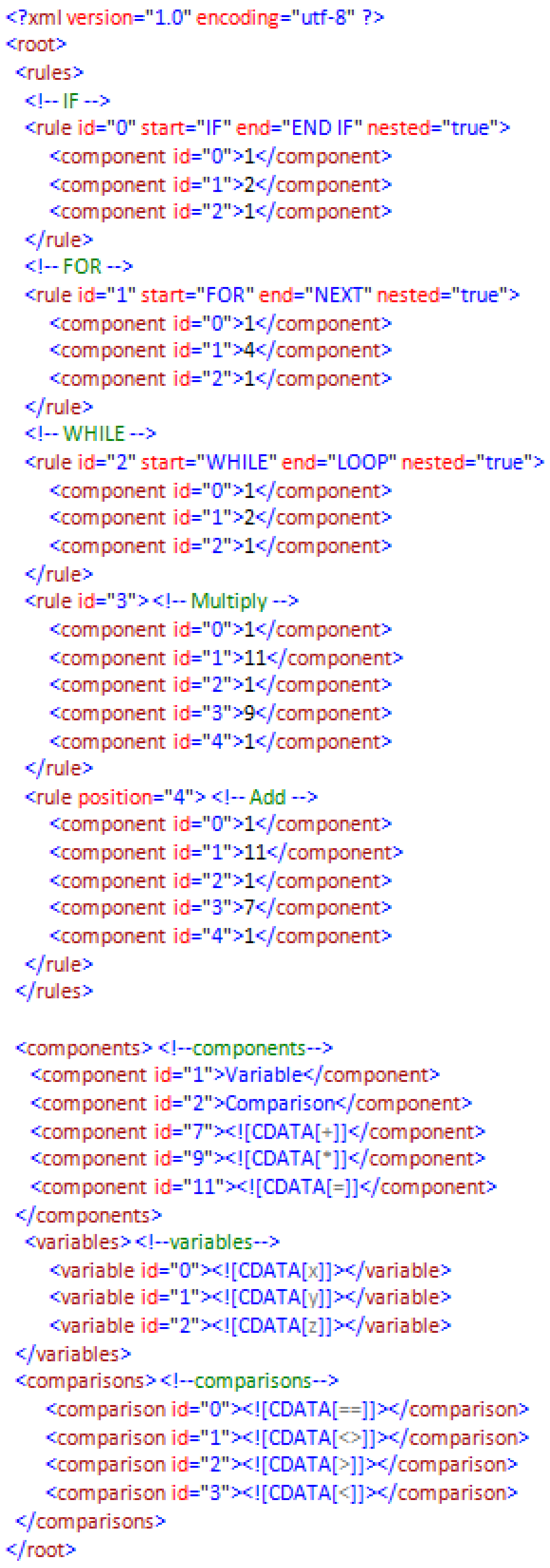
3.5 XML Mapping – Software Program Statements

This section discusses the changes that needed to be made to the system for it to work with Software Program Statements instead of REs. The areas changed were: the XML rules and the repairing function. The following explains the changes involved in order to show the minimum effort required to adapt our system to such an entirely different domain.

Fig. 7 shows a sample of the XML rules and components needed to guide the Genotype-Phenotype stage of the genetic evolution of software program statements. When compared to the Genotype-Phenotype Mapping system for REs (Fig. 4), it is clear that the overall logic remains the same, with each XML rule using the available collections (in this case: 'variables' and 'comparisons') to guide the system through the different components needed for each rule. Identically, there are a number of components that have been introduced to simplify and ensure the consistency of the mapping process for Software Program Statements, but which do not use any genes from the Genotype since they do not need to be evolved. Such components include: the 'Assign (=) operator, the 'Addition' (+) operator, etc.

For example, Table 3 shows the Genotype to be translated using the information in Fig. 7. The value of the first gene is 10. This represents the statement type to be used. In this case, there are five different rules (statements) in the XML file, so 10 mod 5 = 0 means that the statement is an 'If'. This statement contains three different components each requiring the use of a gene to choose from the 'variables' and 'comparisons' collections. The 'variables' collection has three elements, whereas the 'comparisons' has four, therefore, 27 mod 3 = 0, 7 mod 4 = 3 and 13 mod 3 = 1 give elements 'x', '<' and 'y' respectively.

[12] Many HTML tags come in pairs. The system ensures that the same closing tag is chosen based on the opening tag evolved, and the nesting of the tags in the RE hierarchy is preserved.



```xml
<?xml version="1.0" encoding="utf-8" ?>
<root>
 <rules>
  <!-- IF -->
  <rule id="0" start="IF" end="END IF" nested="true">
     <component id="0">1</component>
     <component id="1">2</component>
     <component id="2">1</component>
  </rule>
  <!-- FOR -->
  <rule id="1" start="FOR" end="NEXT" nested="true">
     <component id="0">1</component>
     <component id="1">4</component>
     <component id="2">1</component>
  </rule>
  <!-- WHILE -->
  <rule id="2" start="WHILE" end="LOOP" nested="true">
     <component id="0">1</component>
     <component id="1">2</component>
     <component id="2">1</component>
  </rule>
  <rule id="3"> <!-- Multiply -->
     <component id="0">1</component>
     <component id="1">11</component>
     <component id="2">1</component>
     <component id="3">9</component>
     <component id="4">1</component>
  </rule>
  <rule position="4"> <!-- Add -->
     <component id="0">1</component>
     <component id="1">11</component>
     <component id="2">1</component>
     <component id="3">7</component>
     <component id="4">1</component>
  </rule>
 </rules>

 <components> <!--components-->
   <component id="1">Variable</component>
   <component id="2">Comparison</component>
   <component id="7"><![CDATA[+]]</component>
   <component id="9"><![CDATA[*]]</component>
   <component id="11"><![CDATA[=]]</component>
 </components>
  <variables><!--variables-->
     <variable id="0"><![CDATA[x]]></variable>
     <variable id="1"><![CDATA[y]]></variable>
     <variable id="2"><![CDATA[z]]></variable>
 </variables>
 <comparisons><!--comparisons-->
     <comparison id="0"><![CDATA[==]]></comparison>
     <comparison id="1"><![CDATA[<>]]></comparison>
     <comparison id="2"><![CDATA[>]]></comparison>
     <comparison id="3"><![CDATA[<]]></comparison>
 </comparisons>
</root>
```

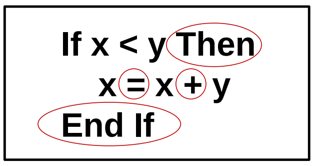
Fig. 7 Sample of XML Rules for Software Statements

Table 3: Genotype

10	27	7	13	19	9	63	4

Table 4: Genotype to Software Statement Mapping

Component No.	*Component*	*Gene Used*	*Modulo*	*Translation*
-	-	10	0	If
1	Variable	27	0	x
2	Comparison	7	3	<
1	Variable	13	1	y
-	-	19	4	Add
1	Variable	9	0	x
11	Assign	-	-	=
1	Variable	63	0	x
7	Addition	-	-	+
1	Variable	4	1	y



```
If x < y Then
   x = x + y
End If
```

Fig. 6 Phenotype

This is the point where the logic in the Genotype-Phenotype mapping of REs changes slightly. In this case, the system knows that although the first statement has been translated in full, the mapping cannot end here. This is because, the XML rule for the IF statement includes an extra attribute: nested = “true” (Fig. 7) which indicates that the IF statement expects another statement inside. The following gene (gene 19) in the Genotype is therefore used to determine the next statement type. Therefore, 19 mod 5 = 4 means that the next statement is ‘Add’. The remaining components are dealt with in the same manner as before (see Table 4).

Once all the required genes have been decoded, the resulting Phenotype is repaired to ensure it is syntactically correct. Fig. 6 shows the complete software structure (Phenotype) for the above example. The additional symbols and programming keywords added by the repairing function are shown encircled. The ‘Assign’ and ‘Addition’ operators are also dealt with in the repairing function, as these are static attributes associated with the Add statement, thus they do not need to be evolved. The repairing function retains the same STACK programming structure and responsibilities as in the REs domain.

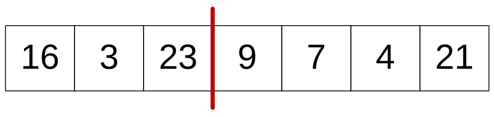
A difference noticed between the two domains is that whilst each rule in Fig. 7 belongs to a different software statement or structure, all of the rules in Fig. 4 are used for the extraction of the same piece of information from the web (the course title in this example). This is because REs are very diverse and as such an unlimited number of REs can be written to extract the same piece of information from a web page. This means that in relation to updating the XML rules once written, the Software Program Statements would require much less attention than REs because these are more rigid structures that need a certain number of components, in a particular order, at all times. For example, the Add statement mentioned above may sum up more than two variables, however, there will always be need for one variable to which this sum is assigned, one ‘Assign’ operator and one or more ‘Addition’ operators.

4. DISCUSSION

There is a significant difference between the two domains chosen for this paper. REs are diverse and sometimes unpredictable, whereas Software Program Statements are structured and rigid. They do not have any components in common. It is in fact difficult to find any similarities between them.

One main difference between the two domains, in relation to their GP representation, is that each RE is represented as a linear, string of integers, where all the integers (genes) are part of the same Genome. A Software Program Statement, on the other hand, can be represented as either a string of integers or as a set of strings of integers. The latter is the case with program structures such as: IF ... ELSE ... THEN, or FOR ... NEXT etc. which incorporate other independent software statements or structures within them. Fig. 3 showed an example of an ‘Add’ statement being nested within the IF structure.

Withall et al. [28] deals with these cases by having separate Genomes for each individual statement. Updating our system to deal with multiple Genomes, would potentially require a considerable change, thus, all the Genomes were instead joined to form one single Genome. Similarly to the example discussed in section 3.5, Fig. 8 shows an example of the Genotype-Phenotype mapping process treating two Genomes (separated with the vertical red line) as one and mapping them to the FOR ... Add ... NEXT structure below. This example works with the data shown in Fig. 7.



```
FOR x = 0 To z
   y = y + x
NEXT
```

Fig. 8 Single Genome Mapping

The only change required to map joined Genomes to a valid Phenotype was to alter the system to recognise the end of one statement and the beginning of the other. In the above example, the system knows that FOR is a nested structure; additionally the XML rule tells the system that the FOR structure needs only two variables, thus the system deduces that the fourth gene (gene 9) must belong to a different statement or structure nested within the FOR loop. Gene 9 corresponds to the ‘Add’ statement, thus the genes following this one will be mapped according to the XML rule for ‘Add’. The repairing function is executed at the end to tidy up the solution by closing the FOR structure.

The above change took the longest to complete – nearly three hours – as it involved altering message calls within two different classes in our Object Oriented System. The changes stated in section 3.5 were all fairly simple to implement and took under 30 minutes to change. Thus overall, this has been a 3.5 hour effort, which considering the significant differences in both domains, we consider this to be a minimal effort.

Results from the execution of our full genetic system in the evolution of REs for the extraction of course names from web pages can be found in [31]. This system would not have to change to work with any other domain that requires the use of REs. However, running the full system on the Software Program Statements domain, or other RE-unrelated domains, would require an additional change – a different Fitness test.

5. CONCLUSION

This paper has discussed an innovative approach to mapping Genotypes to Phenotypes through XML rules. The different steps involved in the process are explained through examples. Two entirely different domains were considered – REs and Software Program Statements - in order to show the amount of effort and time required to adapt the original system to work with a new domain.

Results show that the effort involved in the alterations was minimal, with all the changes taking under 3.5 hours. More time, however, needs to be spent to optimise the translation of constant variables.

The next stage of the research will concentrate on using our Genotype-Phenotype mapping process together with the rest of the GP system to evolve REs for the extraction of other training course details such as prices, dates and locations. Analysis of some preliminary web pages has indicated a partial dependence amongst this data, which will need to be reflected in the Fitness test produced.

ACKNOWLEDGEMENTS

We would like to thank the whole team at ATM for their support and help. Also, thank you to ATM, the Centre for Innovative and Collaborative Engineering (CICE) and Loughborough University for funding our work. We would also like to thank Daniel Sills for his help with some technical .NET concepts.

REFERENCES

[1] Back, T., Hammel, U. & Schwefel, H-P. 1997. Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation.

[2] Banzhaf, W. 1994. Genotype-Phenotype-Mapping and Neutral Variation – A case study in Genetic Programming. Proceedings of the International Conference on Evolutionary Computation. Springer-Verlag, 322-332.

[3] Barrero, D., Camacho, D. & R-Moreno, M. 2009. Automatic Web Data Extraction Based on Genetic Algorithms and Regular Expressions. Data Mining and Multi-agent Integration. ISBN 978-1-4419-0523-9, Springer-Verlag US, 143.

[4] Clarke, M., Hinde, C.J., Whithall, M.S., Jackson, T.W., Phillips, I.W., Brown, S & Watson, R. 2009. Allocating Railway Platforms using a Genetic Algorithm. Research and Development in Intelligent Systems XXVI, Springer London, 421-434.

[5] Conrad, E. 2007. Detecting Spam with Genetic Regular Expressions. SANS Institute Reading Room. Available online at: http://www.giac.org/certified_professionals/practicals/GCIA/00793.php

[6] De Kunder, M. 2010. The size of the World Wide Web (Tilburg University). Retrieved February 23rd, 2010 from http://www.worldwidewebsize.com

[7] Dyer, J. & Bentley, P. 2002. PLANTWORLD: Population Dynamics in Contrasting Environments. In Cantu-Paz E., GECCO, 122-129.

[8] Fogel, D.B. 1994. An Introduction to Simulated Evolutionary Optimisation. IEEE Transactions on Neural Networks.

[9] Fogel, D.B. 1998. The Fossil Record. Fogel, D.B., Ed., IEEE Press

[10] Friedl, J. 2006. Mastering Regular Expressions, Third Edition. O'Reilly & Associates (Aug 2006)

[11] Goyvaerts, J. RegexBuddy. http://www.regular-expressions.info/regexbuddy.html

[12] Hsu, P-H. 2007. Feature extraction of hyperspectral images using wavelet and matching pursuit. ISPRS Journal of Photogrammetry and Remote Sensing. Elsevier Science, Amsterdam, vol. 62 (2), 78-92.

[13] Jackson, D. 2005. Evolving Defence Strategies by Genetic Programming. In Lecture Notes in Computer Science. Springer Berlin, Vol. 3447, 281-290.

[14] Keaveney, D. & O'Riordan, C. 2009. Evolving Robust Strategies for an Abstract Real-time Strategy Game. Proceedings of the 5th International Conference on Computational Intelligence and Games. 371-378.

[15] Keller, R.E. & Banzhaf, W. 1996. Genetic Programming using Genotype-Phenotype Mapping from Linear Genomes into Linear Phenotypes. Proceedings of the First Annual Conference on Genetic Programming, California. 116-122.

[16] Klank, U., Padoy, N., Feussner, H. and Navab, N. 2008. Automatic feature generation in endoscopic images. International Journal of Computer Assisted Radiology and Surgery. Springer, 3, 331-339

[17] Koza, J.R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press.

[18] Langdon, W., Poli, R., McPhee, N. & Koza, J.R. 2008. Genetic Programming: An Introduction and Tutorial with a Survey of Techniques and Applications. In Studies in Computational Intelligence. Springer, Berlin, vol. 115, 927-1028.

[19] Moore, J.P. 2000. Exploring and Exploiting Models of the Fitness Landscape: A Case against Evolutionary Optimization. PhD Thesis, University of Plymouth.

[20] O'Neill, M., Brabazon T., Ryan, C. & Collins J.J. 2001. Developing a Market Timing System using Grammatical Evolution. Proceedings of GECCO.

[21] Poli, R., Langdon, W. & McPhee. 2008. N. A Field Guide to Genetic Programming. Published via http://lulu.com (With contributions from J. R. Koza)

[22] Regular-Expressions.info (Last Update: Jun 2009). http://www.regular-expressions.info/

[23] Rothlauf, F. 2006. Representations for Genetic and Evolutionary Algorithms. Springer-Verlag New York.

[24] Sells, C. 2009. RegexDesigner.NET http://regexdesigner-net.findmysoft.com/

[25] Sun Microsystems. (Last updated: Feb 2008). Lesson: Regular Expressions. http://java.sun.com/docs/books/tutorial/essential/regex/index.html

[26] Thompson, K. 1968. Programming techniques: Regular expression search algorithm. Commun. ACM, Vol 11(6), 419-422

[27] Whigham, P.A. 1995. Grammatically-based Genetic Programming. Workshop on Genetic Programming.

[28] Withall, M.S., Hinde, C.J. & Stone, R.G. 2008. An improved representation for evolving programs. Journal of Genetic Programming and Evolvable Machines. Springer Netherlands, vol. 10(1), 37-70.

[29] Xhemali, D., Hinde, C.J. and Stone, R.G. 2007. Embarking on a Web Information Extraction Project. UKCI Conference on Computational Intelligence (London, UK, Jul 02-04, 2007)

[30] Xhemali, D., Hinde, C.J. & Stone, R.G. 2009. Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. International Journal of Computer Science Issues, vol. 4(1), 16-23.

[31] Xhemali, D., Hinde, C.J. & Stone, R.G. 2010. Genetic Evolution of Regular Expressions for the Automated Extraction of Course Names from the Web. Internal Report. Loughborough University, UK.

APPENDIX J: PAPER 5

Full Reference:

Xhemali, D., Hinde, C.J. & Stone, R.G. (2010-c). Genetic Evolution of ‘Sorting’ Programs through a Novel Genotype-Phenotype Mapping. *Proceedings of the 2010 International Conference on Evolutionary Computation*. Valencia, Spain.

Keywords: Genetic Programming, Genotype-Phenotype mapping, XML, Regular Expressions, Software Programs.

Paper Type: Conference Paper (Accepted for Publishing)

GENETIC EVOLUTION OF 'SORTING' PROGRAMS THROUGH A NOVEL GENOTYPE-PHENOTYPE MAPPING

Daniela Xhemali, Christopher J. Hinde, Roger G. Stone
Department of Computer Science, Loughborough University, Loughborough, United Kingdom
D.Xhemali@lboro.ac.uk, C.J.Hinde@ lboro.ac.uk, R.G.Stone@ lboro.ac.uk

ABSTRACT

This paper presents an adaptable genetic evolutionary system, which includes an innovative approach to mapping genotypes to phenotypes through XML rules. The evolutionary system was originally created to evolve Regular Expressions (REs) to automate the extraction of web information. However, the system has been adapted to work with a completely different domain – Complete Software Programs – to demonstrate the flexibility of this approach. Specifically, the paper concentrates on the evolution of 'Sorting' programs. Experiments show that our evolutionary system is successful and can be adapted to work for challenging domains with minimum effort.

1. INTRODUCTION

Genetic Programming (GP) can be defined as "a systematic, domain-independent method for getting computers to automatically solve problems starting from a high-level statement of what needs to be done" (Langdon et al., 2008). GP research has attracted attention in various fields such as: game strategies (Keaveney & O'Riordan, 2009), military defence (Jackson, 2005), plant biology (Dyer & Bentley, 2002), electronics (O'Neill et al., 2001), railway platform allocation (Clarke et al., 2009), spam filtering (Conrad, 2007), feature extraction from media files (Hsu, 2007; Klank et al., 2008) etc.

An area that has also managed to secure the attention of GP is the automation of Web Information Extraction (WIE) (Atkinson-Abutridy et al., 2004; Barrero et al., 2009; Xhemali et al., 2010-b). The research presented in this paper was originally set up to evolve Regular Expressions (REs) to automate the extraction of web information, specifically training course information such as: course names, dates, locations and prices. The details of this part of the research, including experimental results, were covered previously (Xhemali et al., 2010-b), thus they will not be covered again in this paper.

This paper focuses on a specific part of GP – the genotype to phenotype mapping. Previous work (Xhemali et al., 2010-a) gave details of our innovative approach in relation to its application to REs and to Complete Software Statements (Withall et al., 2008) such as FOR loops, IF statements etc. This paper concentrates on the genotype to phenotype mapping process in relation to Complete Software Programs. Specifically, experiments are carried out to test the complete GP process for the evolution of 'Sorting' programs. 'Sorting' programs were chosen in order to demonstrate the flexibility of this approach, as such programs represent an entirely different domain to REs and a challenging problem for GP systems.

2. RELATED RESEARCH

The genotype-phenotype mapping relates to the way individuals in a population are represented, as this can have a significant effect on the performance of GP. A genotype represents each individual in the search space, whereas its phenotype represents the individual in the solution space (Banzhaf, 1994). Some research, particularly earlier GP research, does not make a distinction between genotypes and phenotypes (Koza, 1992; Whigham, 1995; Conrad, 2007, Snajder et al., 2008 etc.). Individuals in each genetic population remain the same throughout the evolution process. In these works the search space and the solution space are identical.

In 1994, Banzhaf suggested the separation of the two spaces and introduced his work on the genotype-phenotype mapping. The separation involves the encoding of the individuals to a form known as the genotype, which is later on decoded back to the corresponding program, referred to as the phenotype. Since then, many other researchers have embraced the separation into genotypes and phenotypes (Keller & Banzhaf, 1996; Withall et al., 2008; Clarke et al., 2009 etc.). This separation simplifies and increases the efficiency of certain genetic operations such as: reproduction and mutation, because these would no longer be constrained by the parameters used in the program being evolved. In genotype-phenotype based GP, genetic operators such as Crossover and Mutation would be performed on the genotype, whereas other processes, such as the Fitness scoring, would be performed on the phenotype. Sections 3.2 and 3.3 explain this concept further through examples.

On the downside, however, this adds an additional step to the genetic evolution process – the translation or mapping of the genotypes to their corresponding phenotypes. This step occurs after the genetic reproduction stage (i.e. the crossover and mutation) and before the Fitness test can take place. There are researchers who criticise the separation into genotypes and phenotypes (Moore, 2000). The main concern expressed is that the conversion process of a mutated genotype into the phenotype may result in anomalies that could potentially lead to invalid solutions. A direct mapping between the encoded program (genotype) and the actual program (phenotype) is therefore vital to ensure the validity of the solutions (Rothlauf, 2006; Withall et al., 2008).

Banzhaf (1994; 2006) represented his genotypes as linear binary strings. The mapping stage then processed these genotypes from left to right in 5-bit sections, where each 5-bit code mapped to a pre-specified symbol. For example: 00000 mapped to PLUS, 00100 mapped to POW, 11000 mapped to variable X etc. The first bit indicated whether the code represented a function (PLUS, POW etc.) or a terminal (X, Y etc.). The research also discussed their concern about generating constant numbers. Koza (1992) had solved this problem by defining "random ephemeral constants" where constants are only generated once for a particular program and then reused wherever they are needed within that program.

Keller (1996) continued in the footsteps of Banzhaf, concentrating on providing experimental evidence for choosing the genotype-phenotype approach instead of the normal GP approach. Keller used the LALR (Look Ahead LR) parser for the repairing stage of the genotype-phenotype mapping process. LALR parsers scan the input from left to right and construct a rightmost derivation in reverse (Aho and Ullman, 1979). There was a certain amount of redundancy in the genetic coding in both Banzhaf's and Keller's works. They both admitted that, in their works, different binary strings could correspond to the same symbol, which could lead to inconsistencies e.g. 000 and 100 both mapping to ‘a’.

A slightly different genotype representation is seen in the work of Withall et al. (2008). In here genotypes are represented as linear blocks of integers. Each block consists of exactly four integers, each integer representing a different gene. Although both research works used fixed-length genomes, in the work of Banzhaf (1994; 2006) the resulting phenotypes could vary in length, whereas in (Withall et al., 2008) they remained fixed. However, Withall allowed for variable-length genomes through padding, whereby shorter program structures or statements ignored outstanding genes. The first integer in Withall’s genotype always determines the type of function that follows.

Grosan and Abraham (2009) worked with multi-chromosome genotypes. The number of chromosomes was varied. However, the number of genes per chromosome was fixed. In this research, each gene corresponded to either a terminal: T = {a, b, c, d} or a function: F = {+, *}. A function gene also included pointers towards the function parameters to tell the system which terminals were to be manipulated by the function. Also, the first gene of the chromosome was always a terminal. This was to ensure that only syntactically correct programs are evolved. Very differently from above, Yosif et al. (2010) introduced the novel approach of adapting a support vector machine to predict phenotypes from genotype data.

Similarly to Withall et al. (2008), the genotype in our research is represented as a string of integers. There are no fixed length genomes determined however; instead the genotype can contain any number of genes. The direct mapping of these integers to the corresponding structures is achieved through an innovative approach involving XML rules. The first gene in the string determines the XML rule to be followed, which in itself guides the mapping of the rest of the genes into a valid phenotype. This is explained in detail in section 3.2.

3. GENOTYPES TO PHENOTYPES

As previously mentioned, some details about this research, including the GP representation chosen and a thorough explanation of our novel genotype-phenotype mapping approach used on the REs domain, was published in a previous paper (Xhemali et al., 2010-a), thus they will not be repeated here. The rest of this paper concentrates on the application of the genotype-phenotype approach to a completely different domain – that of Software Programs, specifically ‘Sorting’ programs – in order to illustrate the flexibility of our approach. Additionally, the fitness function for the ‘Sorting’ programs is presented and experimental results are discussed.

The examples in this paper are based on the work of Withall et al. (2008). They are kept as close to the original as possible to ensure their integrity. One main difference however, is that in Whithall’s work the evolved programs were in Perl, whereas in this paper their validity is ensured against VB script. Although the evolutionary system is developed in VB.NET with a MS SQL Server database backend, VB Script was used to extend the application to execute each evolved program, as these are obviously not compiled into the application. The .NET framework allows for the execution of dynamic code through the System.CodeDom.Compiler and the appropriate namespaces, however, this is more complicated and slower during execution than utilising the Microsoft Script Control from VB.NET.

3.1 ‘Sorting’ Programs

The ‘Sorting’ program was chosen, because it is a popular, well known program and a standard Computer Science problem due to its higher complexity over other small software programs such as: ‘Sum finder’, ‘Maximum value finder’ etc. The aim of a ‘Sorting’ program is to order a list of integers or characters in ascending order. The output of a ‘Sorting’ program is therefore another list, rather than a single value.

$$
\boxed{
\begin{array}{l}
sort\colon \mathbb{Z}^* \to \mathbb{Z}^* \\[6pt]
pre\_sort\,(L) \triangleq True \\[6pt]
post\_sort\,(L,N) \triangleq bag\_of(N) = bag\_of(L) \wedge ascending(N) \\[6pt]
\quad where\ bag\_of(\langle\rangle) \neq \emptyset \\[6pt]
\qquad bag\_of(\langle x \rangle) \neq \{|x|\} \\[6pt]
\qquad bag\_of(L_1 \frown L_2) \neq bag\_of(L_1) \uplus bag\_of(L_2) \\[6pt]
\quad ascending(N) \triangleq (\forall x, y\colon \mathbb{Z})(x\ before\ y\ in\ N \Rightarrow x \leq y) \\[6pt]
and \\[6pt]
\quad x\ before\ y\ in\ N \triangleq (\exists N_1, N_2, N_3\colon \mathbb{Z}^*)(N = N_1 \frown \langle x \rangle \frown N_2 \frown \langle y \rangle \frown N_3)
\end{array}
}
$$

Figure 1: Specification of ‘Sort’ (Withall et al., 2008)

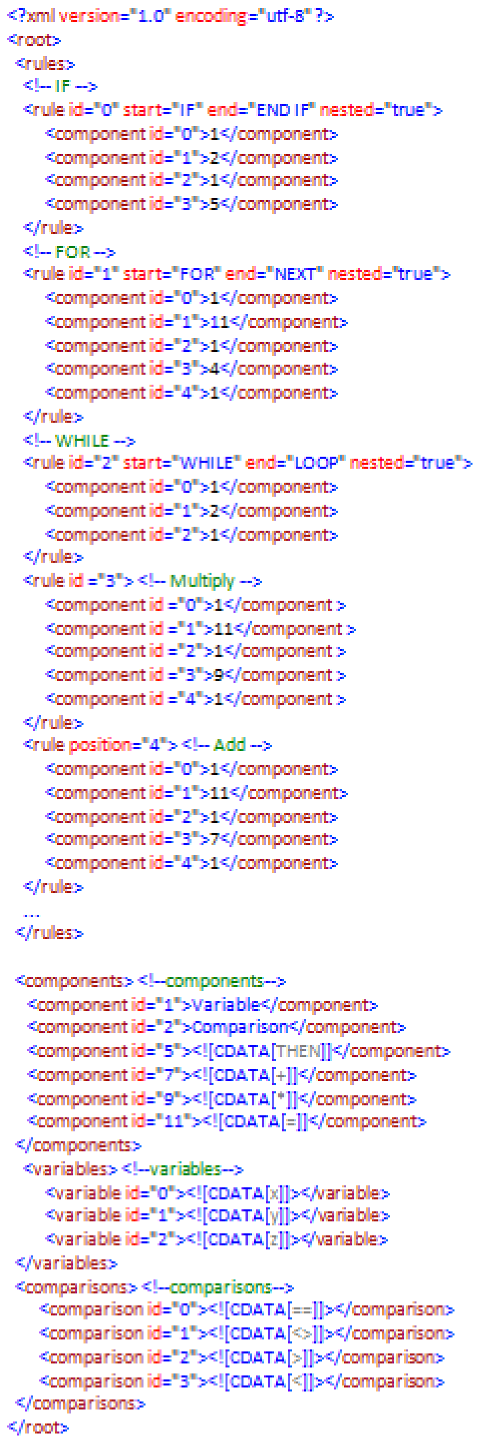
The ‘Sorting’ programs evolved in this research are concerned with the sorting of lists of integers. Figure 1 shows a specification of the ‘Sorting’ problem. The following explains the general details behind the genotype-phenotype mapping for software structures, as well as the additional statements and genes needed for the ‘Sorting’ program.

3.2 XML Mapping

Our genotype-phenotype mapping consists of two main components: the XML rules, which guide the system through the mapping process and the Repairing function, which makes sure that the evolved programs are syntactically correct. The genotype-phenotype mapping process is as follows:

Pseudo-code: Genotype-Phenotype Mapping

```
1) Determine the XML rule to follow
2) Follow the chosen XML rule to the end
3) IF the Genotype has fewer genes than the rule requires
      a. Follow the rule for the number of genes available
      b. Repair outcome to create a valid partial solution.
4) IF the Genotype has enough genes for the XML rule
      a. Follow all the components in the rule
      b. Repair outcome (if necessary) to create a valid and complete solution.
5) IF the Genotype has more genes than the  rule requires
      a. Follow the same steps as above (4a and 4b)
      b. Ignore the rest of the genes in the Genotype
```



```xml
<?xml version="1.0" encoding="utf-8" ?>
<root>
 <rules>
  <!-- IF -->
  <rule id="0" start="IF" end="END IF" nested="true">
     <component id="0">1</component>
     <component id="1">2</component>
     <component id="2">1</component>
     <component id="3">5</component>
  </rule>
  <!-- FOR -->
  <rule id="1" start="FOR" end="NEXT" nested="true">
     <component id="0">1</component>
     <component id="1">11</component>
     <component id="2">1</component>
     <component id="3">4</component>
     <component id="4">1</component>
  </rule>
  <!-- WHILE -->
  <rule id="2" start="WHILE" end="LOOP" nested="true">
     <component id="0">1</component>
     <component id="1">2</component>
     <component id="2">1</component>
  </rule>
  <rule id ="3"> <!-- Multiply -->
     <component id ="0">1</component >
     <component id ="1">11</component >
     <component id ="2">1</component >
     <component id ="3">9</component >
     <component id ="4">1</component >
  </rule>
  <rule position="4"> <!-- Add -->
     <component id="0">1</component>
     <component id="1">11</component>
     <component id="2">1</component>
     <component id="3">7</component>
     <component id="4">1</component>
  </rule>
  ...
 </rules>

 <components> <!--components-->
   <component id="1">Variable</component>
   <component id="2">Comparison</component>
   <component id="5"><![CDATA[THEN]]</component>
   <component id="7"><![CDATA[+]]</component>
   <component id="9"><![CDATA[*]]</component>
   <component id="11"><![CDATA[=]]</component>
 </components>
  <variables> <!--variables-->
     <variable id="0"><![CDATA[x]]></variable>
     <variable id="1"><![CDATA[y]]></variable>
     <variable id="2"><![CDATA[z]]></variable>
 </variables>
 <comparisons> <!--comparisons-->
    <comparison id="0"><![CDATA[==]]></comparison>
    <comparison id="1"><![CDATA[<>]]></comparison>
    <comparison id="2"><![CDATA[>]]></comparison>
    <comparison id="3"><![CDATA[<]]></comparison>
 </comparisons>
</root>
```

Figure 2: Sample of XML Rules

Note that this is not a character by character evolution, because this would increase the search space and dramatically increase the execution time. Instead, programs are divided into two collections: Variables (e.g. “tmp1”, “tmp2”, “tmp3”) and Comparisons (e.g. “>”, “<”, “!=” etc.). Each evolved gene is translated to an element of one of these collections. There is a separate XML rule for each software structure (e.g. “FOR”, “IF ... THEN ... ELSE”, “ADD”, “ASSIGN” etc.). Each rule is composed of a number of components, which guide the system through the translation of each gene to the corresponding software structure (Figure 2). For example, the IF ... THEN structure, of format (IF variable1 comparison variable2 THEN), requires three evolvable genes: two variables and one comparison. Table 1, Table 2 and Figure 3 illustrate this scenario by giving an example of the genotype-phenotype mapping process.

Each component refers to either the elements in the above two collections, or to other predefined elements that do not need to be evolved and as such do not require the use of any extra genes, such as: the different operators associated with each programming structure (e.g. "+" is always associated with "Add"; "=" is always associated with "Assign", thus these do not need to be evolved) or the keywords required by the programming language chosen – in this case VBScript – in order to create syntactically correct code (e.g. "THEN", "TO" etc).

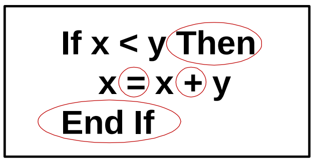
The Repairing function is responsible for making sure that all the different software structures are combined correctly to create a syntactically correct and complete software program. Figure 2 shows a sample of the XML rules and components needed to guide the genotype-phenotype stage of the genetic evolution of software programs. Table 1 shows a sample Genotype to be translated using the information in Figure 2.

Table 1: Genotype

10	27	7	13	19	9	63	4

Table 2: Genotype to Software Statement Mapping

Component No.	Component	Gene Used	Modulo	Translation
-	-	10	0	If
1	Variable	27	0	x
2	Comparison	7	3	<
1	Variable	13	1	y
5	Then-Keyword	-	-	Then
-	-	19	4	Add
1	Variable	9	0	x
11	Assign	-	-	=
1	Variable	63	0	x
7	Addition	-	-	+
1	Variable	4	1	y



```
If x < y Then
   x = x + y
End If
```

Figure 3: Phenotype

Note that the first gene in the genotype is always associated with the XML rule choice. The modulo function is used for this reason. In the above example (Table 1), the value of the first gene is 10. This represents the software structure to be used. In this case, there are five different rules in the XML file, so 10 mod 5 = 0 means that the structure chosen is an 'IF'.

This structure contains four different components (Figure 2). The first three components require the use of a gene to choose from either the ‘variables’ or the ‘comparisons’ collections. The ‘variables’ collection has three elements, whereas the ‘comparisons’ has four, therefore, 27 mod 3 = 0, 7 mod 4 = 3 and 13 mod 3 = 1 give elements ‘x’, ‘<’ and ‘y’ respectively.

The fourth component (id=“5”) tells the Repairing function that the THEN keyword is required next. The ‘THEN’ keyword is a mandatory requirement for IF statements in VB.NET or VBScript, thus this component does not need to be evolved and as such does not require the use of an extra gene from the genotype.

The nested = “true” attribute seen in Figure 2 indicates that the IF structure, differently from one-line statements, such as ‘Add’ or ‘Multiply’, expects other statement(s) inside. The following gene (gene 19) in the genotype is therefore used to determine the statement type to be nested in this IF. Therefore, 19 mod 5 = 4 means that the next statement is ‘Add’. The remaining components are dealt with in the same manner (see Table 2).

Once all the required genes have been decoded, the resulting phenotype is repaired to ensure it is syntactically correct. The Repairing function is an independent function, which scrutinises the phenotype created in order to guarantee the syntactic validity of the solution. This function is in charge of tasks like: closing software structures appropriately (e.g. FOR loops in VB.NET need to end in NEXT for them to be valid); adding the necessary operators, which do not need to be evolved (e.g. the ‘Add’ (“+”) and the ‘Assign’ (“=”) operators); tidying up the phenotype in cases when there are fewer genes available than required by the XML rule; adding header and footer information about a solution such as: variable declaration or variable return etc. All this is achieved through the use of a STACK programming structure, which works in a LIFO (Last In First Out) manner.

Figure 3 shows the complete software structure (phenotype) for the above example. The additional symbols and programming keywords added by the repairing function are shown encircled.

Updating the XML rules, once written, would require little effort, because software statements and structures are rigid in the number of components needed and the order in which they are needed. For example, the Add statement mentioned above may sum up more than two variables, however, there will always be need for one variable to which this sum is assigned, one ‘Assign’ operator and one or more ‘Addition’ operators. Adding new XML rules for new statements or structures would be equally as effortless, because it would only require the addition of the different components for that structure to the XML rules as well as any additional variables or comparisons that may be required.
The changes that were made to our Genotype-Phenotype mapping process for the ‘Sorting’ software programs involved the simple addition of one more structure and a few more variables and comparison values. Specifically, a special nested FOR loop was added to the current software structures, similarly to Withall et al. (2008), which includes two nested FOR loops in the following format:

```
FOR ($var_1$ = 0 TO $var_2$)
    FOR ($var_3$ = $var_1$+1 TO $var_2$)
        ...
```

The above could have been achieved through the existing FOR loop structure (Figure 2), however, we chose to add the nested FOR loop, because this is a standard structure used when comparing elements in a list and it results in faster execution of the evolution process.
Due to the 'Sorting' program dealing with lists or arrays of integers, a few more variable types needed to be added to distinguish between normal variables and array variables (since a normal variable x is entirely different from the variable array(x)). This is to maintain the accuracy of the evolved solutions.

The Experimental Results section presents and discusses the results of our experiment.

3.3 Fitness Function

Similarly to Withall et al. (2008), the fitness function in this research has been simplified to only compare adjacent items in the list of integers rather than all the possible pairs in the list. This decision was made in order to speed up the evolution process.

Code: Fitness Function for 'Sorting'

```
If list.Length > 0 Then
   For i As Int32=0 To list.Length - 2
      If list(i) <= list(i + 1) Then
         fitness += 1
      End If
   Next
End If

If list.length > 1 Then
   fitness = fitness/(list.Length-1)
Else 'The list has only got one integer
   fitness = 1
End If
```

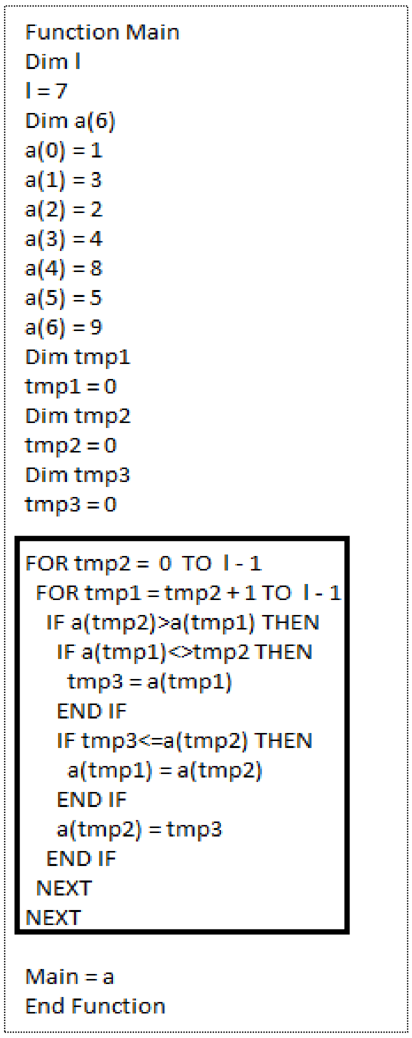
The code for the 'Sorting' program is presented above to show its simplicity. Note that the necessary header information such as variable declaration and initialisation are left out for clarity. Also note that this function is only called if the post-evolution list of integers contains the same elements as the pre-evolution list of integers.

4. EXPERIMENTAL RESULTS

Details about the evolutionary system used in this research were discussed in a previous paper (Xhemali et al., 2010-b) however, the main parameters used by the system are summarised in here to help the reader fully appreciate the experimental results presented:

- Population size: 10
- Tournament size: 40%
- Fitness Target: 1
- Uniform Crossover Rate: 50%
- Mutation Rate: 1 gene per genotype
- Initial Population: Random

The following gives the results from the experiments set up for the evolution of ‘Sorting’ programs. In order to maintain the similarity with the work of Withall et al. (2008), there were ten runs of the experiment, each with a maximum number of generations of 50,000. None of the experiments needed this many generations however, and each run produced a valid ‘Sorting’ program.



```
Function Main
Dim l
l = 7
Dim a(6)
a(0) = 1
a(1) = 3
a(2) = 2
a(3) = 4
a(4) = 8
a(5) = 5
a(6) = 9
Dim tmp1
tmp1 = 0
Dim tmp2
tmp2 = 0
Dim tmp3
tmp3 = 0

FOR tmp2 =  0  TO  l - 1
 FOR tmp1 = tmp2 + 1 TO  l - 1
  IF a(tmp2)>a(tmp1) THEN
   IF a(tmp1)<>tmp2 THEN
    tmp3 = a(tmp1)
   END IF
   IF tmp3<=a(tmp2) THEN
    a(tmp1) = a(tmp2)
   END IF
   a(tmp2) = tmp3
  END IF
 NEXT
NEXT

Main = a
End Function
```

Figure 4: Complete ‘Sorting’ Solution: (1)

Figure 4 shows one of the ‘Sorting’ programs produced by the system. Note that this example shows the full VBScript code executed from within VB.NET 2008. The header and footer of the solution (shown in Figure 4 within the dotted rectangle) was added to the solution by the Repairing function. The part of the code shown within the solid rectangle was evolved by the system (and tidied up by the Repairing function, as described in section 3.2).

All experiments presented in this paper were based on the sorting process of a fixed list of seven integers – a(1, 3, 2, 4, 8, 5, 9) – however, further experiments were carried out with different integer lists to ensure that the results in this paper were not somehow influenced by the integers chosen.

Table 3 gives information about five of the ‘Sorting’ solutions evolved (Fitness Score: 1), which were also generated by Withall et al. (2008). The information includes the number of generations (Gens) taken for the solution to be generated and the amount of time taken to arrive to this solution. The remaining five solutions (6-10), also with a Fitness Score of 1, refer to additional ‘Sorting’ solutions evolved during the experiments for this paper.

Table 3: ‘Sorting’ Results

Solution No.	**Generations**	**Generations (Withall)**	**Time**	**Time (Withall)**
1	35467	47975	57’33”	1h04’28”
2	16200	14189	25’25”	19’22”
3	8950	8219	12’08”	10’22”
4	15982	16312	23’12”	21’57”
5	762	5834	2’36”	8’00”
6	20050	-	27’31”	-
7	739	-	2’19”	-
8	1690	-	4’48”	-
9	559	-	1’45”	-
10	93	-	0’15”	-

The solutions themselves (without the header or footer information), including the five that were different from the solutions generated by Withall et al. (2008) are listed in Appendix A.

In relation to the results in Table 3, it is important to note that comparing the results of different evolutionary systems is not straightforward and by no means definitive. Despite experiments being carried out on the same domain, a touch of luck is also involved in getting to a perfect solution quickly from evolutionary experiments. This is because, depending on the crossover of the genes and particularly on the (random) gene that gets chosen for mutation and the outcome of the mutation itself, a perfect solution may not be reached at the same time by different systems. Furthermore, even experiments carried out on the same system at different times, may not arrive at the same solution at the exact same generation. Keeping this in mind, Table 3 shows that Withall et al. (2008) need fewer generations than this research for two of the above solutions (2 and 3). This research needs fewer generations for solutions 1, 4 and 5.

One thing is evident however, that Withall’s timings (time needed per generation) are lower than those in this research. As previously mentioned Withall’s genetic evolutionary system was all written in Perl, thus there was no need for converting the evolved code to a Scripting language first to achieve dynamic execution, since Perl is already a scripting language. The system in this research however, was written in VB.NET and includes the additional step of forcing the execution of the solution as VBScript from within the VB.NET application. This affects the overall execution speed. Furthermore, it was observed during the experiment that VBScript displayed a message box to the user each time the script took longer than normal to execute. The user was then asked to choose whether to allow the script to continue running or end it and allow the system to move onto the next script.

We managed to change the VBScript control to make the decision by itself, without involving the user. Although this has sped up the execution, it is still an extra step that VBScript has to do behind the scenes, which increases the overall execution time for the evolutionary system.

A potential solution may be to change the system to execute the dynamic programs in Perl instead of VBScript and see if this makes a difference to the above timing issue. Another solution may be to allow .NET itself to execute the dynamic code through the inbuilt System.CodeDom.Compiler, however, this is a more complicated solution, which may still result in time wastage due to the additional manipulation that VB.NET will have to make to itself to compile the dynamic code at runtime.

5. CONCLUSION

This paper has discussed an innovative approach to mapping genotypes to phenotypes through XML rules in relation to software statements and structures as well as complete software programs. Utilising XML gives this technique many advantages including: improved readability, compatibility with many programming languages, portability and extendibility (XML is not restricted to a limited set of keywords defined by the proprietary vendors, which aids the process of creating rules of different levels of complexity).

Experiments were set up to test the complete evolutionary system on the evolution of ‘Sorting’ programs. The ‘Sorting’ program was chosen because it represents a well known, standard Computer Science problem, which is complex enough to really test the evolutionary system.

Our evolutionary system was originally created to deal with the evolution of Regular Expressions. However, it was discovered, that it was easily adaptable to other domains, including that of Complete Software Structures. In fact, the only two areas that needed to be changed to make the system work for the new domain, were the XML rules and the Fitness function.

The experimental results from the evolution of ‘Sorting’ programs highlight the efficacy and flexibility of our system, despite the complexity of the ‘Sorting’ problem for GP systems. However, further testing needs to be done to ensure the reliability of this approach for other complex programs.

ACKNOWLEDGEMENTS

We would like to thank the team at ATM for their support. Also, thank you to ATM, CICE and Loughborough University for funding our work. We would also like to thank Daniel Sills for his help with some technical .NET concepts.

REFERENCES

Aho, A.V. & Ullman, J.D. 1979. Principles of Compiler Design. Addison Wesley, ISBN 0-201-00022-9.

Atkinson-Abutridy, J., Mellish, C., Aitken, S. 2004. Combining information extraction with genetic algorithms for text mining. IEEE Intelligent Systems. 19(3), pp. 22-30.

Banzhaf, W. 1994. Genotype-Phenotype-Mapping and Neutral Variation – A case study in Genetic Programming. Proceedings of the International Conference on Evolutionary Computation. Springer-Verlag, pp.322-332.

Banzhaf, W. 2006. genotype-phenotype-mapping and neutral variation – A case study in Genetic Programming. Lecture Notes in Computer Science, Springer Berlin, 866, pp. 322-332.

Barrero, D., Camacho, D. & R-Moreno, M. 2009. Automatic Web Data Extraction Based on Genetic Algorithms and Regular Expressions. Data Mining and Multi-agent Integration. Springer-Verlag, pp. 143.

Clarke, M., Hinde, C.J., Withall, M.S., Jackson, T.W., Philips, I.W., Brown, S. & Watson, R. 2009. Allocating Railway Platforms using a Genetic Algorithm. Research and Development in Intelligent Systems XXVI. Springer London, pp. 421-434.

Conrad, E. 2007. Detecting Spam with Genetic Regular Expressions. SANS Institute Reading Room. Available: http://www.giac.org/certified_professionals/practicals/GCIA/00793.php

Dyer, J. & Bentley, P. 2002. PLANTWORLD: Population Dynamics in Contrasting Environments. In Cantu-Paz E., GECCO, pp. 122-129.

Grosan, C. & Abraham, A. 2008. Evolving Computer Programs for Knowledge Discovery. Social Science Research Network (SSRN).

Hsu, P-H. 2007. Feature extraction of hyperspectral images using wavelet and matching pursuit. ISPRS Journal of Photogrammetry and Remote Sensing. Elsevier Science, Amsterdam, 62 (2), pp. 78-92.

Jackson, D. 2005. Evolving Defence Strategies by Genetic Programming. In Lecture Notes in Computer Science. Springer Berlin, 3447, 281-290.

Keaveney, D. & O'Riordan, C. 2009. Evolving Robust Strategies for an Abstract Real-time Strategy Game. Proceedings of the 5th International Conference on Computational Intelligence and Games. pp. 371-378.

Keller, R.E. & Banzhaf, W. 1996. Genetic Programming using Genotype-Phenotype Mapping from Linear Genomes into Linear Phenotypes. Proceedings of the First Annual Conference on Genetic Programming, California. pp. 116-122.

Klank, U., Padoy, N., Feussner, H. and Navab, N. 2008. Automatic feature generation in endoscopic images. International Journal of Computer Assisted Radiology and Surgery. Springer, 3, pp. 331-339.

Koza, J.R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press.

Langdon, W., Poli, R., McPhee, N. & Koza, J.R. 2008. Genetic Programming: An Introduction and Tutorial with a Survey of Techniques and Applications. In Studies in Computational Intelligence. Springer, Berlin, 115, pp. 927-1028.

Moore, J.P. 2000. Exploring and Exploiting Models of the Fitness Landscape: A Case against Evolutionary Optimization. PhD Thesis, University of Plymouth.

O'Neill, M., Brabazon T., Ryan, C. & Collins J.J. 2001. Developing a Market Timing System using Grammatical Evolution. Proceedings of GECCO.

Rothlauf, F. 2006. Representations for Genetic and Evolutionary Algorithms. Springer-Verlag New York.

Snajder, J., Basic, B.D., Petrovic, S. & Sikiric, I. 2008. Evolving new lexical association measures using genetic programming. Proceedings of the Association for Computational Linguistics. Ohio, pp. 181-184.

Whigham, P.A. 1995. Grammatically-based Genetic Programming. Workshop on Genetic Programming.

Withall, M.S., Hinde, C.J. & Stone, R.G. 2008. An improved representation for evolving programs. Journal of Genetic Programming and Evolvable Machines. Springer Netherlands, 10(1), pp. 37-70.

Xhemali, D., Hinde, C.J. & Stone, R.G. 2009-a. Domain-Independent Genotype to Phenotype Mapping through XML Rules. International Journal of Computer Science Issues, 7(3).

Xhemali, D., Hinde, C.J. & Stone, R.G. 2010-b. Genetic Evolution of Regular Expressions for the Automated Extraction of Course Names from the Web. Proceedings of the International Conference on Genetic and Evolutionary Methods. Las Vegas.

Yosif, N., Gramm, J., Wang, Q., Noble, W., Karp, R. & Sharan, R. 2010. Prediction of Phenotype Information from Genotype Data. Communications in Information and systems. 10(2), pp. 99-114.

APPENDIX A

These are the ‘Sorting’ programs generated in this research. Solution (1) was shown in Figure 4. Solutions (1-5) match those obtained by Withall et al. (2008), whereas the remaining (6-10) are new.

Note that parts of the following programs may look different to ‘hand written’ code for ‘Sorting’ programs. Solutions (3) and (9) are the closest to the conventional ‘hand coded’ version.

<u>‘Sorting’ Solution: (2)</u>

```
FOR tmp2 =  0  TO  l - 1
  FOR tmp1 = tmp2 + 1 TO  l - 1
    tmp4 = a(tmp1)
    tmp3 = a(tmp2)
    IF a(tmp2)>a(tmp1) THEN
      a(tmp1) = a(tmp2)
      FOR tmp1 = 0 to l - 1
        IF tmp4<>tmp3 THEN
          a(tmp2) = tmp4
        END IF
      NEXT
    END IF
  NEXT
NEXT
```

<u>‘Sorting’ Solution: (3)</u>

```
FOR tmp2 =  0  TO  l - 1
  FOR tmp1 = tmp2 + 1 TO  l - 1
    IF a(tmp1)<=a(tmp2) THEN
      tmp3 = a(tmp1)
      a(tmp1) = a(tmp2)
      a(tmp2) = tmp3
    END IF
  NEXT
NEXT
```

‘Sorting’ Solution: (4)

```
FOR tmp2 =  0  TO  l - 1
  FOR tmp1 = tmp2 + 1 TO  l - 1
    IF a(tmp1)<a(tmp2) THEN
      tmp3 = a(tmp1)
      a(tmp1) = a(tmp2)
      FOR tmp1 = 0 to l - 1
        a(tmp2) = tmp3
      NEXT
    END IF
  NEXT
NEXT
```

‘Sorting’ Solution: (5)

```
FOR tmp2 =  0  TO  l - 1
  FOR tmp1 = 0 TO  l - 1
    IF a(tmp2)<=a(tmp1) THEN
      tmp4 = a(tmp1)
      a(tmp1) = a(tmp2)
      a(tmp2) = tmp4
    END IF
  NEXT
NEXT
```

‘Sorting’ Solution: (6)

```
For tmp2 = 0 To l - 1
  For tmp5 = tmp2 + 1 To l - 1
    For tmp1 = 0 To l - 1
      tmp4 = a(tmp1)
      If tmp3 <= a(tmp1) Then
        If tmp4 >= a(tmp5) Then
          tmp3 = a(tmp5)
          a(tmp1) = tmp3
          a(tmp1) = a(tmp5)
          a(tmp5) = tmp4
        End If
      End If
    Next
  Next
Next
```

‘Sorting’ Solution: (7)

```
FOR tmp1 =  0  TO  l - 1
  FOR tmp2 = 0 TO  l - 1
    IF a(tmp1)<=a(tmp2) THEN
      tmp3 = a(tmp2)
      a(tmp2) = a(tmp1)
      a(tmp1) = tmp3
    END IF
  NEXT
NEXT
```

‘Sorting’ Solution: (8)

```
For tmp2 = 0 To l - 1
  For tmp1 = tmp2 + 1 To l - 1
    If a(tmp2) >= a(tmp1) Then
      tmp3 = a(tmp1)
      a(tmp1) = a(tmp2)
      If a(tmp2) <= a(tmp1) Then
        For tmp4 = 0 To l - 1
          a(tmp2) = tmp3
        Next
      End If
    End If
  Next
Next
```

‘Sorting’ Solution: (9)

```
FOR tmp1 =  0  TO  l - 1
  FOR tmp2 = tmp1 + 1 TO  l - 1
    IF a(tmp1)>a(tmp2) THEN
      tmp3 = a(tmp1)
      a(tmp1) = a(tmp2)
      a(tmp2) = tmp3
    END IF
  NEXT
NEXT
```

‘Sorting’ Solution: (10)

```
For tmp2 = 0 To l - 1
  For tmp1 = tmp2 + 1 To l - 1
    tmp3 = a(tmp2)
    If tmp3 >= a(tmp1) Then
      tmp4 = a(tmp1)
      tmp3 = tmp4
      a(tmp1) = a(tmp2)
      a(tmp2) = tmp3
    End If
  Next
Next
```