

# **Non-Invasive User Modelling to Support Browser Behaviour Reflection**

by

**Ann Marie Sexton,**

**Dissertation**

submitted to the

University of Dublin, Trinity College

in partial fulfillment of the requirements

for the Degree of

**Master of Science in Computer Science**

**University of Dublin, Trinity College**

September 2010

# Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

---

Ann Marie Sexton

September 12, 2010

## Permission to Lend and/or Copy

I agree that Trinity College Library may lend or copy this dissertation upon request.

---

Ann Marie Sexton

September 12, 2010

# Acknowledgments

A special thanks to my wonderful husband Peter and my beautiful girls Caroline, Lisa, Anne, Shannon, Caoimhe, Vanessa and Aiveen for all their support and patience during these past two years while I have been studying. This would not have been possible without that. To my sisters Jackie and Carol who are there through thick and thin and my wonderful friends and great classmates, too numerous to mention. To Eoin, who has been present for all the great and tough moments, for seeing this through to the end with the proof reading and encouragement.

A sincere thanks to my supervisor Owen Conlan for the amazing guidance, insight and enthusiasm, it was a pleasure to work with you. Also to Kevin Koidl, a wonderful guy, thank you for all your help. To Siobhan Clarke and all the lecturers of the NDS class, thank you.

This dissertation marks the end of a very tough year, a journey, to say the least, and to me it proves that ANYTHING is possible if you put your mind to it!

ANN MARIE SEXTON

*University of Dublin, Trinity College*

*September 2010*



# Non-Invasive User Modelling to Support Browser Behaviour Reflection

Ann Marie Sexton, M.Sc.

University of Dublin, Trinity College, 2010

Supervisor: Dr.Owen Conlan

While user modelling and personalisation is an ongoing area of research, it is also a mature field with work dating back more than twenty five years with no system having gained mass adoption. In this work we introduce AMS, a user modelling system that works silently in the background while users browse the internet, modelling browsing behaviour, collecting browsing data and analysing it with a view to inferring the user's interests.

Prevalent issues from similar systems, such as privacy concerns or intrusion to the user's browsing experience are nicely circumvented here as we engineer the data to being contained and stored at the user's browser while only using implicit methods to collect the data. Text analytics are used to extract key terms from the raw data which is collected from pages that the user visits and a rating is applied to these terms, taking into consideration the time spent actively viewing the page with respect to the length of the page.

We show how AMS is effective in surmising the user's interests, within the bounds of the evaluations that were carried out and we show how getting results from the linked data environment played a role in enhancing the user's overall experience.

This work is dedicated with love to my mother Annie and my mother-in-law Lily

# Contents

|                                                     |           |
|-----------------------------------------------------|-----------|
| <b>Acknowledgments</b>                              | <b>iv</b> |
| <b>List of Tables</b>                               | <b>x</b>  |
| <b>List of Figures</b>                              | <b>xi</b> |
| <b>Chapter 1 Introduction</b>                       | <b>1</b>  |
| 1.1 Motivation . . . . .                            | 4         |
| 1.2 Research Question . . . . .                     | 5         |
| 1.3 Objectives . . . . .                            | 5         |
| 1.4 Approach . . . . .                              | 6         |
| 1.5 Dissertation Outline . . . . .                  | 7         |
| <b>Chapter 2 State of the Art</b>                   | <b>9</b>  |
| 2.1 Introduction . . . . .                          | 9         |
| 2.2 Modelling based on browsing behaviour . . . . . | 9         |
| 2.3 User Modelling . . . . .                        | 15        |
| 2.3.1 Implicit User Modelling . . . . .             | 17        |
| 2.3.2 Explicit User Modelling . . . . .             | 18        |

|                                 |                                                   |           |
|---------------------------------|---------------------------------------------------|-----------|
| 2.3.3                           | Existing User Modelling Systems . . . . .         | 19        |
| 2.4                             | Linked Data . . . . .                             | 21        |
| 2.5                             | Text Analytics . . . . .                          | 23        |
| <b>Chapter 3 Design</b>         |                                                   | <b>25</b> |
| 3.1                             | Introduction . . . . .                            | 25        |
| 3.2                             | Design Influences from State of the Art . . . . . | 26        |
| 3.3                             | Requirements . . . . .                            | 26        |
| 3.3.1                           | Implicit User Modelling . . . . .                 | 26        |
| 3.3.2                           | User Centred Design . . . . .                     | 27        |
| 3.3.3                           | Text Analytics . . . . .                          | 32        |
| 3.3.4                           | Ratings Methodology . . . . .                     | 34        |
| 3.3.5                           | Data Storage . . . . .                            | 36        |
| 3.3.6                           | User Interface . . . . .                          | 37        |
| 3.3.7                           | Linked Data . . . . .                             | 38        |
| 3.3.8                           | Technology Choices . . . . .                      | 39        |
| 3.3.9                           | Processing platform . . . . .                     | 40        |
| 3.4                             | Architecture . . . . .                            | 41        |
| 3.5                             | Summary . . . . .                                 | 42        |
| <b>Chapter 4 Implementation</b> |                                                   | <b>43</b> |
| 4.1                             | Introduction . . . . .                            | 43        |
| 4.2                             | Development . . . . .                             | 43        |
| 4.3                             | Software Implementation . . . . .                 | 44        |
| 4.3.1                           | Google Chrome Extension . . . . .                 | 44        |
| 4.3.2                           | Google App Engine . . . . .                       | 50        |

|                                             |                                                 |           |
|---------------------------------------------|-------------------------------------------------|-----------|
| 4.3.3                                       | Java Service . . . . .                          | 52        |
| 4.3.4                                       | Data Storage . . . . .                          | 53        |
| 4.3.5                                       | User Reflection . . . . .                       | 54        |
| 4.3.6                                       | Linked Data . . . . .                           | 56        |
| 4.4                                         | Implementation Issues . . . . .                 | 56        |
| 4.5                                         | Conclusion . . . . .                            | 57        |
| <b>Chapter 5 Evaluation and Results</b>     |                                                 | <b>58</b> |
| 5.1                                         | Introduction . . . . .                          | 58        |
| 5.2                                         | Evaluation . . . . .                            | 59        |
| 5.2.1                                       | User Centred Design Evaluation . . . . .        | 59        |
| 5.3                                         | Results from Evaluation . . . . .               | 63        |
| 5.3.1                                       | Pre Evaluation Questionnaire Results . . . . .  | 64        |
| 5.3.2                                       | Post Evaluation Questionnaire Results . . . . . | 68        |
| 5.4                                         | Main Findings . . . . .                         | 75        |
| <b>Chapter 6 Conclusions</b>                |                                                 | <b>77</b> |
| 6.1                                         | Overview . . . . .                              | 77        |
| 6.2                                         | Motivation and Objectives . . . . .             | 77        |
| 6.3                                         | Contribution to State of the Art . . . . .      | 78        |
| 6.4                                         | Future Work . . . . .                           | 79        |
| <b>Bibliography</b>                         |                                                 | <b>81</b> |
| <b>Appendix A Java Service</b>              |                                                 | <b>87</b> |
| <b>Appendix B Evaluation Questionnaires</b> |                                                 | <b>89</b> |

# List of Tables

3.1 Ratings Table . . . . . 36

# List of Figures

|      |                                                         |    |
|------|---------------------------------------------------------|----|
| 2.1  | An archetypal system employing a user model [1]         | 16 |
| 3.1  | User Centred Design - Highlighted words of interest (1) | 29 |
| 3.2  | User Centred Design - Highlighted words of interest (2) | 30 |
| 3.3  | User Centred Design - Highlighted words of interest (3) | 31 |
| 3.4  | Hub and Spoke view of System Requirements               | 33 |
| 3.5  | Web browser developer tools                             | 38 |
| 3.6  | High Level Architecture                                 | 41 |
| 4.1  | High Level Overview of AMS                              | 44 |
| 4.2  | AMS - manifest.json                                     | 45 |
| 4.3  | Google Chrome Extension Installation Page               | 46 |
| 4.4  | AMS Google Chrome Extension                             | 47 |
| 4.5  | Web database setup                                      | 47 |
| 4.6  | Collecting text in bold lettering from web page         | 48 |
| 4.7  | Message passing with Google Chrome Extension            | 49 |
| 4.8  | Message received                                        | 49 |
| 4.9  | GAE Application Console                                 | 51 |
| 4.10 | AMSTextServlet.java                                     | 51 |

|      |                                                                                                 |    |
|------|-------------------------------------------------------------------------------------------------|----|
| 4.11 | Interaction between AMS and GAE . . . . .                                                       | 53 |
| 4.12 | Sample of SQL used to set up database tables and insert results<br>from text analysis . . . . . | 54 |
| 4.13 | Screenshot of AMS . . . . .                                                                     | 55 |
| 4.14 | CSS to create AMS User Interface . . . . .                                                      | 55 |
| 4.15 | SQL to select the top keywords and their ratings from the dataset .                             | 55 |
| 4.16 | Popup.html code to get and display the Top Keywords to the user .                               | 56 |
| 5.1  | User Centred Design - Preliminary Results from AMS . . . . .                                    | 60 |
| 5.2  | User Centred Design - Browsing History . . . . .                                                | 61 |
| 5.3  | User Centred Design - Results from AMS . . . . .                                                | 62 |
| 5.4  | User Centred Design -Results from Linked Data environment . . . .                               | 63 |
| 5.5  | Weekly time spent on internet . . . . .                                                         | 65 |
| 5.6  | Weekly time spent on repetitive tasks, social networking etc . . . .                            | 65 |
| 5.7  | Weekly time spent browsing/searching for information . . . . .                                  | 66 |
| 5.8  | Reaction to popups . . . . .                                                                    | 67 |
| 5.9  | Reaction to profiling for marketing purposes . . . . .                                          | 67 |
| 5.10 | Experiences of searching for information . . . . .                                              | 68 |
| 5.11 | Reaction to AMS website suggestions . . . . .                                                   | 69 |
| 5.12 | Reaction to AMS keyword suggestions . . . . .                                                   | 70 |
| 5.13 | Reaction to AMS content . . . . .                                                               | 71 |
| 5.14 | Results from Linked Data . . . . .                                                              | 72 |
| 5.15 | Number of unique terms collected for 6 users over a 30 minute period                            | 75 |
| B.1  | Pre Evaluation Questionnaire . . . . .                                                          | 91 |
| B.2  | Post Evaluation Questionnaire . . . . .                                                         | 93 |



# Chapter 1

## Introduction

“The Internet is the world’s largest library. It’s just that all the books are on the floor.” John Allen Paulos

In the civilised world today it is difficult to conceive a life without the internet. We constantly search the World Wide Web (WWW) at any time of the day or night to gain access to news, weather, email, travel, sport, entertainment etc. We tirelessly sift through vast quantities of information in order to find the snippet we are looking for. The WWW has revolutionised the way we think, the way we work, the way we do business. Businesses improve their productivity and competitiveness with instant access to information and lightning fast electronic processing ability. The WWW has become an invaluable resource to so many, from academic researchers to students to people with disabilities. It has brought the world closer to us and has played a huge role in globalisation as it integrates and amalgamates the people of the world.

With the exponential growth of the WWW [2], there is an exciting area of research, to investigate and develop ways to capture how users browse the web with a view to using this information to profile the user and infer the users' interests. If this information can be captured and utilised correctly, and that is a challenge to say the least, it could then be used in a number of ways, e.g. to develop a user model which could work in unison with the user, being continuously updated, which could present personalised and relevant suggestions and information to the user to improve their browsing experience as they browse the web. Making systems more useful and more usable is a fundamental goal of human-computer interaction and also to provide users with a browsing experience which is appropriate for their specific knowledge and interests.

As the internet evolves there is a constant desire to get more relevant information and to get it faster than ever before. We are inclined to browse blindly, using search engines without really knowing how best to formulate a query that will represent what we are looking for. Lazonder et al. [3] compare the work of finding useful information on the WWW to that of the work of a detective, always trying to ask the right questions, to the right sources, and then having to piece the information together to reach a result. We are presented with millions of results for every query we make and waste time trying to sift out the items which are relevant to us. Most likely we will see something else that we weren't looking for and go off on a deviation away from the task at hand. Current personalised search tools like iGoogle<sup>1</sup> or My Yahoo<sup>2</sup> strive to make our browsing experience better

---

<sup>1</sup><http://www.google.com/ig>

<sup>2</sup><http://my.yahoo.com/>

by presenting a web page that we can custom tailor to display information such as the local weather or our daily horoscope etc.

Accurate personalisation has become a Holy Grail of the WWW [4]. We have come to expect to be welcomed by name on websites that we visit frequently and we expect results of our queries to be tailored to our local area. Social websites strive to learn as much as possible about us in order to use that information to present more relevant information back to us. Most websites that use any form of personalisation do so by utilising a log-in system which enables them to segregate unique information by user. Facebook takes note of who we are friends with and who our friends are friends with and makes suggestions as to who we might know. Ecommerce websites such as Amazon.com store our shopping history, what we purchase and what we look at, and use this information to make relevant suggestions of items we might be interested in, based on what people with similar purchase/browsing history also bought/looked at. This is a powerful sales tool and no doubt one that has promoted huge revenue over the past number of years.

It is possible to envision a world where there is an individual repository of information about each person, their characteristics, their likes and dislikes, their interests, their area of work, their patterns of behaviour etc. Basically one log-in area for the whole WWW where a user profile, which can be easily accessed by web applications, including search engines, to be used to adapt information on a person by person basis as they search or browse the web. Imagine being able to go onto any search engine and type a query such as 'what movie will I watch tonight?' and be presented with only movies that you will like and of course

not have to sift through ones that you have already seen. The main obstacle preventing this idea from becoming a reality at present is how to deal with the issue of privacy surrounding the collecting and storage of personal information, but the other hindrance is the ability to capture users' interests accurately in a non invasive way. To best deal with the issue of privacy, the system developed in this work is a client side web browser plug-in which stores data on each person's individual machine and is not accessible by any means outside of that machine.

## **1.1 Motivation**

While user modelling and personalisation is an ongoing area of research, it is also a mature field with work dating back more than twenty five years [5] and as you will read about in the State of the Art (Chapter 2), no system has yet managed to gain mass adoption. There appears to be no quick fix, one for all solution that can magically interpret each individual's requirements. The motivation for this work is a desire to find a simple, effective means of modelling a user's browsing behaviour with a view to reflecting the perceived interests of the user back to them. If this system can accurately model the user, it could be used in a variety of future work in the area of user modelling. The decision was made to build a non-invasive, implicit system due to the annoyance to users [6] and the time consuming nature of explicit questionnaires or nuisance pop-ups of any kind when attempting to acquire information from the web. The performance of the system is evaluated within specific domains and results are compared to the system's performance in the open domain.

## **1.2 Research Question**

This dissertation investigates to what degree a user's interests can be inferred by gathering text, performing text analysis and applying a rating, on information from the pages that he/she visits while both actively searching for information and casually browsing for information.

## **1.3 Objectives**

One goal of user modelling is to realise the possibility of providing users with their own unique view of information, personally tailored in the best possible way to suit their individual needs. When a user is provided with personally tailored information in this way, as opposed to being presented with the same information that everyone else with their own unique requirements are getting, they have the ability to process the information more rapidly and ultimately save the time and energy that it would otherwise take to sift through the mountains of information available on the WWW.

The primary goal of this work is to identify the key areas of interest to a user when browsing for information and to do this in a non-invasive way. We will then utilise the most dominant elements of our findings to develop a system for capturing, analysing and utilising this information in order to create a unique individual user model which contains a surmised account of the user's interests. This model could then be used in future work, e.g. to build a web browser plug-in that makes informed and relevant suggestions to the user as he/she browses the

WWW, perhaps by extracting information from a linked data set.

## 1.4 Approach

After analysing each of the web browsing and user modelling experiments that are outlined in the state of the art Chapter 2, for what was successful and what wasn't, a plan was drawn up of how an effective user modelling system could be built. Chapter three will outline the motivation for the design of AMS and outline the steps that were taken to arrive at the final design. On deciding that a web browser plug-in was a suitable fit for the requirements, we began experimenting with both Firefox add-ons and Google Chrome extensions. The choice to use Google Chrome is discussed in Chapter 3. The next step was to decide how best to collect the data and make it work towards the goal of the project. Researching the document object model (DOM) which is a cross platform, language independent means of accessing and interacting with HTML objects, we discovered the possibilities that the DOM allowed and experimented with the collecting of any text that was represented by tags or id's on a web page. The DOM works by allowing access to any element by using the Javascript language, `getElementById` or `getElementsByTagName` calls. The next step was to decide what type of text analysis to use on the collected text, how the data would be rated, followed by where it would be stored. Finally the means of presenting the data back to the user was decided on. An in-depth discussion of the process outlined here can be found in Chapter 3.

## 1.5 Dissertation Outline

This dissertation is divided into a theoretical part, chapters 1- 2, and a practical part, chapters 3-7. This **Introduction Chapter** outlines the motivation for this work, the research question that will be addressed by this dissertation, and talks about user modelling and personalisation, and what the benefits of an accurate user model can be.

**Chapter 2** presents the State of the Art in the area of user modelling and web browsing behaviour and discusses how people browse the WWW and the types of user models that have been developed to date, the motivation for these, what has worked well and what hasn't worked well and the reasons why.

The design chapter, **Chapter 3**, assesses the State of the Art and discusses the key points that inspired the design of our user modelling system AMS. This chapter introduces AMS, a non-invasive user modelling system that we have developed. There follows a description of the design issues, the motivation for the design based on the User Centred Modelling approach, the problems that were encountered along the way and how they were overcome.

**Chapter 4** discusses the implementation of AMS and provides an outline of the technological architecture of the system, the technologies which were used to develop the system and the underlying web service that provides text analysis to the system.

The evaluation chapter, **Chapter 5** talks about the evaluation, the domains that the experiments were carried out in, how the experiments were carried out, what evaluation means were used and outlines the results.

Finally **Chapter 6** concludes with the main points of the project, what was achieved and learned and includes an outline of the contribution to the State of the Art. Future work which could be carried out in this area is also discussed here.



# Chapter 2

## State of the Art

### 2.1 Introduction

This chapter presents the state of the art in web browsing behaviour and user modelling. Section 2.2 discusses the various types of web browsing behaviour, what is most popular and what are the trends. In 2.3 the recent research in user modelling is discussed where both client side methods and server side methods are looked at. There is also a discussion on implicit versus explicit user modelling. In 2.4 linked data is introduced and shown how it will be included in Chapter 4's evaluation of AMS. Finally in 2.5 we briefly discuss text analytics and how it can play a role in user modelling systems.

### 2.2 Modelling based on browsing behaviour

Two well known browsing methods to obtain information from the WWW are by entering search terms into a query based search engine e.g. Google, or by using

hyperlinks to navigate from one page to another. Half of the evaluators of AMS, as can be seen in Chapter 4, cite the difficulty of finding relevant information quickly and easily, the majority found popups to be annoying and none of the evaluators ever fill out surveys trying to build a profile of them for marketing purposes.

There are a number of ways of characterising web surfing data, whereby we can analyse browsing data at the client, proxy or server side. In this study, we are interested in capturing browsing behaviour at the client side, more specifically from within the browser, as we believe that this is where we can best capture and analyse the user's browsing behaviour and in turn use this analysis to formulate a method to find and present more applicable information.

A study of user behaviour on the WWW was carried out by Catledge and Pitkow [7] in 1994 and was one of the first studies done. 107 people were studied over 21 days, with each participant using the XMosaic browser. The authors enabled a client-side trace file to be generated by configuring the browser, and this trace file detailed user interface selections and user navigation patterns. They recorded 31,134 navigation commands over the 21 days, which approximates to 14 page requests each day by each user. The results show that the majority of user activity was in following hyperlinks (52%) and in using the 'Back' button (41%). The other categories such as using the 'Forward' button, bookmarking, etc., evenly accounted for the remaining 7%. Another widely cited study on web browsing behaviour was carried out by Tauscher and Greenberg [8] in 1995. They surveyed browsing information for 23 users over a six-week period, with approximately 19,000 navigation commands, which equates to approximately 21 page

requests each day by each user. The study results show the most prominent web browsing activity (50%) to be that of clicking on hyperlinks to navigate between pages, with 43% using the 'Back' button as the second most popular means of web browsing. These results are in line with the findings of Catledge and Pitkow [7].

The Tauscher and Greenberg study [8] also investigated how users revisit web pages. They calculated the probability of a user revisiting a page, i.e. a user requesting an URL that they had previously requested at some other time in the study, to be approximately 0.58. This would suggest that the actual figure for revisiting links is much higher, as the number of times that these users requested the same URL previous to the study is not taken into consideration in these figures. It is common practice in browsing the web to request the same URL again and again, as often the content is updated but the URL is still the same (e.g. [www.aertel.ie](http://www.aertel.ie) giving frequent news updates)

Cockburn et al. [9] noted some shortcomings in the earlier [8] studies and their work extends these earlier works to give a more up to date view of browsing behaviour. Among the shortcomings listed, were the substantial growth of the web and development of navigational tools since the earlier studies. They also felt that the duration of the previous studies was not long enough to capture an accurate account of the number of page re-visits by users. Furthermore, since users were working on XMosaic, which was not their browser of choice in most cases, their behaviour could be somewhat different than their behaviour while using their favourite browser. In the study, Cockburn et al. [9] analysed the history.dat files of 17 users working on the Netscape browser, on dates from October 1999 to January

2000. Users consisted of faculty and staff of the University of Canterbury in New Zealand. 84,841 page requests were made over the duration of the study, which is approximately 42 page requests by each user every day, which is significantly higher than the earlier studies [7][8] and is an indication of a substantial increase in daily use of the WWW. The results of the study found the re-visitation rate to be 0.81, a sizeable increase over the Tauscher and Greenburg [8] figure of 0.58. Part of this increase can be explained by the difference in length of the studies, four months versus six weeks, giving a more accurate picture. Furthermore with the evolution of the WWW and the introduction of new technologies such as Ajax, people are more likely to re-visit pages as content is often continuously updated.

The next significant study was conducted by Huberman et al. [10] They found through studying the surfing data of America Online users, over a five day period in December 1997, that there are strong consistencies with users surfing patterns, especially with the amount of times that users click within websites. They study the number of links a user clicks on within a Website and devise a mathematical probability of the depth of search. Each page visited is given a value and if the user clicks on the next page it is an indication that this page is also of value. It is difficult to tell the value of the next page so they assume that it is randomly related to the current page and assign it a random value plus the value of the page just left. When the value of continuing to browse is determined to be lower than the expected value of the information to be found on the next page, the user ceases. From these assumptions, Huberman et al. [11] determine the probability of the amount of links that a user is likely to follow within a website.

A pattern of web browsing behaviour is beginning to emerge from these experiments, whereby users are more inclined to navigate their way through hyperlinks while browsing, or by using the 'Back' button, than by any other means. It is important also to keep in mind that there is a strong tendency towards re-visiting pages previously viewed.

When we focus on the different ways of seeking information we find that Wilson [12] categorises four types of search and acquisition; 'passive attention, passive search, active search and ongoing search'. Passive attention is inadvertently obtaining information while we are listening to the radio or watching television. Passive search is when we are searching for information and discover some useful information that we hadn't actively been searching for. Active search is when an individual is intentionally searching for information. Ongoing search is building upon the information base that is created from active searching. Building these four categories into web browsing behaviour, we can identify passive attention with browsing through links with no intent or direction, passive search with bookmarking, active search with entering search terms or directly entering URL's to navigate to a required page and ongoing search with re-visiting pages and bookmarks and entering explicit search terms.

There are many specific actions that a user can perform when browsing the WWW, such as clicking on hyperlinks to follow a chosen path, dragging the sliding bar, or using a mouse wheel to scroll up and down a page to read or scan the information. Some users tend to highlight text with a mouse as they read it or use the mouse pointer to underline each word being read. Using the forward and back

buttons are also a popular method of moving between pages. Bookmarking a web page is probably one of the strongest indications of current interest in a particular topic. It is important to bear in mind however, the results in [9] indicate that although users tend to keep a large numbers of bookmarks, it is not often that they get deleted, meaning that a high percentage of bookmarks are most likely invalid or interest in them has expired. What a user is downloading is also a direct implication of interest. Some of the more modern browsers, e.g. Google Chrome, offer a page of thumbnail screenshots which are links to the user's most frequently used websites. As shown earlier, 81% of users tend to revisit a webpage, which implies that four out of every five pages have been viewed by the user previously, and this would justify the reasoning behind the Google Chrome thumbnail page. The history page is also a good source for connecting users with previously visited websites.

A typical web browsing session consists of reading or scanning the current webpage, deciding whether to follow any links, returning to a previously visited page, going to a bookmarked page or bookmarking the current page. Following a link indicates possible interest in the destination page but it is more indicative of interest in the page containing the link. If the user navigates away from the destination page in a very short time, this is an implication that the link was of little value to the user. Similar to reading a book, we tend to read web pages from left to right and from top to bottom. When a user skips past a link without exploring it, we can assume that the user is not interested in this link, at least at the present time.

This research focuses on collecting and analysing information from web pages that are visited, such as the title text, the headings text, keywords from the meta tags, text in bold or strong lettering, text entered into a search bar and text highlighted with a mouse as a user browses the web page. The motivation behind what information to collect is outlined in Chapter 3 and is largely based on the results of our User Centred Design process. The collected data is analysed, processed and reflected back to the user. The goal is to determine whether the analysis of these methods alone will provide us with enough data about the user's interests, to develop an accurate user model of web browsing behaviour.

## **2.3 User Modelling**

In traditional information retrieval systems producing static hypermedia content, there has been a missing link whereby the user is inundated with information, much of which is of no relevance to the specific user. Take for example a ten year old looking up information for a school project on the subject of the human body. While the query will return a list of documents ranked in overall interest of the users of the WWW, the ten year old will be presented with the same information as a Biology PhD student entering the same search term. Situations such as this example is, no doubt what originally inspired user-modelling research, to find a way to profile the individual user in order to present a personalised browsing experience while using the WWW.

The core element of this research is the user modelling. A user model is a model of how the computer represents the user's interests, information needs,

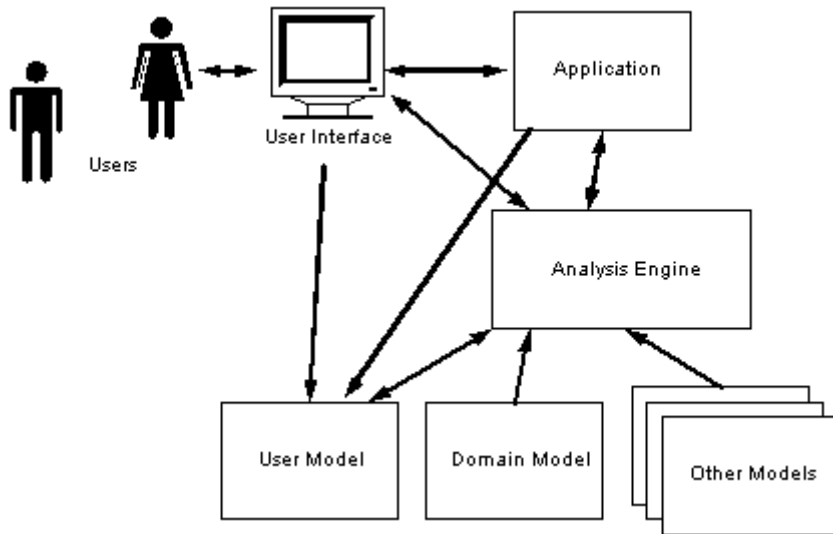


Figure 2.1: An archetypal system employing a user model [1]

expectations and goals. Kules et al. in [1] express their view of a typical user modelling system in Figure 2.1. While this is not specific to any one user modelling system, it is indicative of the components which are used in user modelling systems in general and indeed comprises some of the components of the system which is presented in this work as will be seen in Chapter 3.

In this research our aim is to infer what the general interests of the user are, what the user possibly knows about the subject in question, and what cognitive processes the user presents which might affect his/her use of the system. We must, first of all, determine what information is available to us and what information we are going to capture to leave us best equipped to model the user's interests. The user's query to a search engine provides the ultimate indication of intent and interest and provides a good base point to begin our modelling. From there we will capture key terms from the text of the page, perform some text analysis and



develop an algorithm to extract the key points to use in formulating the user model.

One thing which is important to consider at this point is that along with having specific long term interests, the user may also be looking for impromptu information which may represent a fleeting interest, and once the information is obtained, the user will most likely have no further need for this type of information. Take for example, a user looking for information about renting a house and may scroll through many house renting websites. Once the house is rented the user has most likely lost interest in those websites. In these situations, information about the user collected over an extensive period of time is most likely ineffective, whereas the current search situation such as which search results the user is viewing currently, can be expected to be much more useful.

### **2.3.1 Implicit User Modelling**

There are basically two methods of user modelling, implicit and explicit, and we will endeavour to explain these now. Implicit user modelling is a means of building a model by collecting information that is implied by the user. There are no direct questions asked of the user, there are no boxes to tick, basically there is no direct user input involved. An implicit model tends to work in silence, taking and analysing information as the user browses, perhaps noting keystrokes, search terms etc. This method is possibly more favourable to the user as it requires no interaction and is not annoying. [13] shows that implicit modelling can be as effective and sometimes improve on explicit modelling in terms of accurate

feedback.

Sometimes however, implicit user models can be annoying in a different manner. One of the widest known implicit user modelling projects is the Lumiere project [14]. The Lumiere project was developed at Microsoft Research and uses Bayesian algorithms to capture the uncertainty between the user's needs and goals from their observed actions and queries. It was from this project that the Microsoft Office Assistant was born, and made its first appearance in Office 97. While the technology worked really well, the downside of the Office Assistant was its invasiveness to the user and it was found to be annoying and disruptive to the work at hand [15]. Consequently it was turned off by default in the Office XP edition. It is this project that inspired the idea of a non-invasive user model to be used in this work

### **2.3.2 Explicit User Modelling**

In direct contrast to implicit user modelling methods, an explicit user model has a basis of direct input by the user whereby the modelling is done based on pre-defined methods of gathering information from the user, be it through questionnaires or other means of directly gaining the user's input. Users can also be requested to provide explicit feedback on the relevance of the results produced by the system in question. This is called relevance feedback and had been proven to be an effective means of obtaining accurate user preferences [16].

A good example of an explicit user modelling system is the Avanti Project [17] [18] which is an information system centred around a metropolitan area for use

by a variety of people with different needs and different backgrounds, from the elderly to the handicapped to the tourist to the resident. Avanti uses an initial interview to build a profile of the user and then uses stereotypes to place users into subgroups. The information presented to the user is then customised based on the profile and the subgroup, for example users with disabilities could be given information on accessibility in the area.

The explicit model however, although having the ability to collect more relevant information and collect it efficiently, can be frustrating and off-putting to the user, causing a distraction to the user's thinking pattern and perhaps sending them off in another direction or inspiring them to give up what they were doing. Or as aptly put in [19] 'Since the cost to the user is high and the benefits are not always apparent, it can be difficult to collect the necessary data and the effectiveness of explicit techniques can be limited'.

### **2.3.3 Existing User Modelling Systems**

The following is a brief account of a selection of the work done to date on user modelling. The User Modelling Tool (UMT) proposed by Brajnik et al. [20] specifies stereotypes, containing user type descriptions in the form of attribute-value pairs. The stereotypes are arranged in random hierarchies and sub-stereotypes can inherit information from the main stereotypes. Every stereotype has triggers and these can indicate when a stereotype is applicable to the current user. A rule interpreter provided by UMT allows the defining of user model rules. Assertions about the user are stored by UMT, and are generated by the application system. If

these assertions are deemed reliable or unreliable, they are considered as constant or as assumptions to be deleted later.

Although there has been much work done on user modelling and many different types are in use, the focus has shifted in recent years towards ontology based user models which ties in with the vision of a semantic web and linked data. Ontology is defined as 'an explicit specification of the conceptualisation of a domain' [21] or in other words it is a description of a common understanding of a domain that can be used by both machines and users. In [22] Kay outlines the advantage of using ontologies in user modelling. Kay states that a user model "needs an agreed ontology and representation so it can be used by different application programs".

The idea of a shared user model is discussed in [23]. Here, the Personis server is presented, which bases its user model on component-evidence-source triplets. Each application can define its own triplets without having to regard the others, which limits its reusability. OntobUM (Ontology based User Model) [24] is an ontology-based user modelling framework which was developed mainly for knowledge management systems. In OntobUM there are two parts to the user model, an explicit and an implicit part. The implicit part is concerned with system usage where the authors characterise users as 'readers, writers or lurkers', while the explicit part holds qualities such as the profile, preferences and identity information. OntobUM is a user-modelling server, which uses the RDF/RDFS format, the same format used in linked data. A broad overview of commercially available systems is outlined by Kobsa in [25].

We are interested in client-side customisation, where a user has their own distinct user model that has learned about their interests and can customise the WWW browsing experience accordingly. Letizia [26] scans the WWW ahead of the user, investigating links which are on the current webpage, and in turn using this information for the user model to recommend pages that it has determined will be of interest to the user. Letizia observes the user behaviour and builds the model based on these observations. Similarly in Syskill and Webert [27] users are encouraged to rate Web pages and then a profile is generated of each user's specific interests. The AiA project [28] adds a presentation agent which directs the user's attention to probable areas of interest. The agent has a model which uses individual user's preferences, interests and needs and this model is used to decide what information to present and how best to present it. The presentation agent is on the client side, similar to this research which also focuses on the client side.

## **2.4 Linked Data**

In 1998, Tim Berners Lee, who is attributed to being the founder of the WWW, shared his vision for the future of the web. He coined the term 'semantic web' [29] to portray a WWW where computers can understand, analyse and act upon data, without the need for human intervention.

With the rapid growth of available information on the World Wide Web, any of the established search engines have difficulty meeting the information needs of users, in terms of effectiveness. Users are presented with information overload,

millions of possible websites to answer one simple query. For example, searching on Google for the word 'Amazon' will return 648 million results and will present all types of completely unrelated information, ranging from the Amazon rainforest to the Amazon eCommerce website.

In 2006, Tim Berners Lee introduced the idea of 'linked data' [30] with the aim being to link related data and to allow users to share data in much the same way that they share documents now. Linked Data refers to data available on the Web that it is machine-readable, clearly defined, 'linked to external data, and can be linked to from external data [9]. A brief overview of the design principles for linked data is that every resource has a unique URI and these URI's can be looked up using HTTP [30]. There should be links to other resources and information about a resource is expressed as a set of simple subject-predicate-object triples in the Resource Description Framework (RDF) [31]. In his 2009 address at the Ted Conference, Berners-Lee encourages users to put their data on the web and more importantly to define relationships within that data.

Linked data is a subset of the semantic web where data that is related is linked together and the semantics of the relationship clearly defined, with a goal to making web search more effective. We envisage the linked data set to play a key role in future work around the field of user modelling and personalisation.

In this work we are interested in comparing the effectiveness of our user model in a traditional browsing situation to that in the linked data environment. If our user experiment proves successful, it could be further enhanced by technologically

implementing a Sparql query to query a linked data set such as DBpedia with a view to presenting relevant information to suit the user's individual needs.

## 2.5 Text Analytics

Text analytics is the analysis of text with the goal of changing text into usable data by studying word frequency distributions and applying pattern recognition and text mining algorithms [32]. Text analytics is a similar concept to text mining. Tan et al. [33] define two areas of text mining, text refinement and knowledge distillation. Text refinement transforms documents into a pre chosen intermediate form and knowledge distillation extracts patterns or knowledge from the intermediate form. Nasukawa et al. [34] define the very basics of text mining to be concept extraction which in this case are simple keywords or features which summarise the content of a document. Similar to this work where keywords are extracted, the same problems were encountered by Nasukawa et al. whereby not every word in the document defines the document concept. Therefore the challenge is how to extract the meaningful words and how to cluster them into useful groups.

Text analytics are used by many companies to analyse data such as customer satisfaction questionnaires, customer complaints etc. In the medical field natural language processing systems such as MedLEE [35] and GENIES [36], have been developed to assist in text mining for specific clinical information. Hearst et al. [37] distinguish between text mining and data extraction and describes situations such as a computer successfully extracting key information such as name, address, job skills etc from a CV as data extraction while their criteria for text mining is

that there must be some new information produced.

AMS uses text analytics, albeit a lightweight variety, to produce interesting data from search terms entered into a web search query engine, from text selected as we scroll, keywords from meta tags, text which makes up the title of the page, text in bold and text in headings on the page. Techniques such as the porter stemmer algorithm [38] are used to transform words into their root form. This reduces the repetitiveness of like words and groups words in a more useful way.

One issue related to text mining that is gaining momentum as the WWW grows is the issue of privacy. For example, as social networking sites gain more knowledge about us, from knowing who are friends are to having photographs of our families in their possession, we are becoming aware of the importance of privacy and users are beginning to question the integrity of the websites they are using and are reluctant to divulge personal information without first knowing how this information will be protected.

In recent years methods have been proposed to conceal sensitive information when text mining in order to represent the information without loss of privacy. Some important techniques include methods such as l-diversity [39], k-anonymity [40], perturbation [41] and condensation [42]. AMS addresses privacy effectively by collecting and storing data only at the client side, within the web browser and therefore no masking of data is necessary. The text that goes out to the hosted text service is anonymous and holds no association to the user.



# Chapter 3

## Design

### 3.1 Introduction

The purpose of this project is to design a non-invasive user modelling system that works in the background as users browse the web. This system should gather key information from each page that a user visits, perform text analysis on the information, and subject it to some form of rating mechanism, in order to transform it into usable data that is a representation of the user's interests. At this point the user can decide to view the information if he so desires and decide whether the information is relevant to him/her. The outcome of the evaluations performed by users will determine if the methods used are adequate enough to build a feasible user model that infers the users' interests.

## **3.2 Design Influences from State of the Art**

The principal lesson learned from analysing the state of the art is that it is difficult to model a user's browsing behaviour accurately by implicit means. However we have also learned that modelling by implicit means is more favourable to the user as it doesn't negatively impact on their browsing experience. Another noteworthy message that comes across from the state of the art is that there is no 'one size fits all' when it comes to anticipating users browsing behaviour. Attempting to build a user model by explicit means, e.g by asking direct questions, while it proves more efficient and more accurate, can be very off-putting and annoying to users. Anything unexpected appearing on the screen or any indication to say that there is something 'watching' every move the user makes is also a major disincentive and can inspire users to cease browsing. Further analysis of the state of the art and the issues surrounding privacy of user's data steered us away from the choice of developing a user modelling system that is housed at the server side. A more favourable option is to capture and store the data at the client side. This method should prove to be popular with users as it eliminates the possibility of exploitation of their browsing information.

## **3.3 Requirements**

### **3.3.1 Implicit User Modelling**

The points discussed in 3.2 inspired the creation of a system that completely works in the background, naturally collecting information and modelling the user's browsing behaviour without the user being constantly reminded that it is there, in

other words an implicit user modelling system. [43] uses implicit user modelling techniques, similar to AMS but different in the types of information that are gathered to build the model, to improve users search accuracy with the UCAIR web search tool which is a client side web browser plug in.

There are a number of factors that must be taken into consideration in order to build an accurate model. [13] shows that the total time spent scrolling on a web page is a good indication of user interest and therefore this is one of the aspects of the user's browsing that AMS will take into consideration. It also makes sense to take the length of the web page into consideration since a user spending the same amount of time on a page that is 1200 pixels long compared to a page that is 3600 pixels long should indicate a higher level of interest. These factors, the time spent on the page with respect to the length of the page are the basis of our rating system which is discussed in 3.3.4.

### **3.3.2 User Centred Design**

The concept of user centred design and the discussion in [44] of the dangers of neglecting to include the user in the design process, inspired us to choose one person to follow this project from start to finish. This user's input was used throughout the entire design and evaluation stages.

User centred design is a popular concept in user modelling and was the basis of the HyperAudio system [45]. HyperAudio is a portable adaptive guide for museum visitors. As the visitor moves around the museum building, their location is used to navigate and guide their visit. Users are categorised by answering an initial

questionnaire and the electronic portable device is programmed specifically per category, (age group, interests, levels of expertise etc)

The person we chose for our User Centred Design process is a qualified NLP Life Coach, we will call her 'Carol' for the purposes of this project. Carol readily agreed to help out in the process and a time was decided for the initial meeting. When asked to choose a domain to use for evaluation purposes Carol chose the area of life coaching which includes NLP and hypnosis.

In order to decide what information is most useful to capture from a webpage to initiate our user modelling system we carried out the following experiment. At the outset Carol was asked to browse the web to find some websites that interested her. The search terms that she entered were noted. When she found a page that she liked she was asked to print it and highlight any single words on the page that she felt were indicative of her interests or that caught her attention. She was asked to repeat this task until she had three pages of interest printed and highlighted. Figure 1, 2 and 3 were what resulted from this experiment

It is clear to see from the highlighted words that it is not necessarily the words of the body of the page that are of interest, in fact that is rarely the case. Instead the most important words consistently appear either in the headings or bold lettering. We also note that a lot of the key words that were highlighted also happen to appear in the title. We continued our experiment for another five pages and found that the results were consistent, the majority of useful words were either in the heading text or bold lettering and some of the highlighted key words that may



Creating Your Future®  
The Secret of Creating Your Future®  
Time Line Therapy®  
An acknowledged methodology created by Tad James



HOME GENERAL INFORMATION PRODUCT INFORMATION & ORDERING TRAININGS & SEMINARS AFFILIATED SITES



• Dr. Tad James - Creator of Time Line Therapy® Trainings and Techniques® and of "The Secret of Creating Your Future" seminar and coaching techniques.

TIME LINE THERAPY® TRAINING

- Free Info Pack
- Free Downloads
- Schedule of Events
- Enroll Now
- Contact Us

Time Line Therapy® Training  
The original and only  
one of its kind in the world



The latest development in NLP, Time Line Therapy® techniques are a unique and unmatched method for creating powerful change in business, education, and therapy.

Time Line Therapy® utilizes a person's own internal "Time Line" to work with their unconscious minds in a variety of ways; including, healing emotional traumas and eradicating unwanted thoughts, emotions and behaviors.



To learn Time Line  
ENROLL NOW! [CLICK HERE](#)  
Therapy

Time Line Therapy® has been so successful in its results that it has been utilized for over a decade by thousands of people including psychiatrists, psychologists, marriage and family counselors, social workers, life and business coaches, and even athletic coaches.

Be free from your past, create your future! Make it so .... with Time Line Therapy®.

Bothered by Negative Emotions? Want to get rid of some emotional baggage? Then watch this Time Line Therapy® Video. Get rid of Negative Emotions:

- download the Video Clip for Windows Media Player: Time Line Therapy® Video
- Download the Audio Clip for MP3: Time Line Therapy® Audio

Time Line Therapy® is so highly regarded, in fact, that the Council of Psychotherapy in Croatia asked to be trained by "The James" in "Time Line Therapy" so they could help many of the victims of the war suffering from "post traumatic stress disorder". The specific collection of techniques called Time Line Therapy® produces long-lasting transformation very quickly and easily. The process is faster than what is currently called Brief Therapy. These powerful Time Line Therapy™ techniques are becoming the method of choice to make fast, effective, long-term changes in behavior.

Your "Time Line" is how you unconsciously store your memories or how you unconsciously know the difference between a memory from the past and a projection of the future. Behavioral change in an individual takes place at an unconscious level. People don't change consciously. The Time Line Therapy™ techniques allow you to work at the unconscious level and release the effects of past negative experiences and change "inappropriate" programming in minutes rather than days, months or years.

Time Line Therapy® training will teach you a collection of techniques that allow you to gain emotional control over your life. Inappropriate emotional reactions, such as bursts of anger, periods of apathy, depression, sadness, anxiety, and chronic fear, are responsible for preventing people from achieving the quality of life they desire. Limiting decisions, such as "I'm not good enough," "I'll never be rich," or "I don't deserve a great marriage," create false limitations and hamper your ability to create reachable and attainable goals and outcomes. Created by Dr. Tad James, Time Line Therapy™ techniques enable you to eliminate many types of issues in your past, thus allowing you to move forward toward your goals and desires.

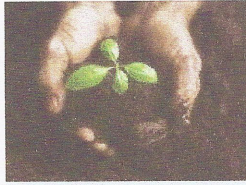
Applications of NLP and Time Line Therapy®:

Achieving Professional Excellence: Whether you're already succeeding in your profession, having some difficulties, or if you're transitioning into a new position, NLP and Time Line Therapy® techniques can help you achieve, maintain and enhance excellence.

What our Graduates are saying about Time Line Therapy®:

Figure 3.1: User Centred Design - Highlighted words of interest (1)





**Culture change & Business Development:**

*Creates a sympathetic environment for people, encourages risk taking and nurtures growth. EI facilitates change faster as it synthesises cognitive & technical skills to engage the collective wisdom.*

We need to examine what impact EI leadership has to offer for our personal performance, emotionally, physically and spiritually. There is now widespread support from pioneering researches like that of Dr. Alex [Concord of The Concorde Initiative](#). Her understanding of how the brain functions has led her to identify that the limbic system is in fact the powerhouse and *Chief Executive* of the human-mind/human body system. Linear thinking is cognitive, but to create transformational change the limbic system has to be involved. Transformational change (limbic) is to change fundamentally how we see things and how you do things as a consequence.

Cognitive change is about providing more guidelines and rules that are more restricting. Limbic change is about changing the individual *intention* by addressing emotional intelligence that is not really in alignment with who they are now, their current purpose, and that creates more choices.

Dr. Alex Concord call *high performance* low stress systems. What is stress? **Stress** arises when the demand on the whole human system exceeds resources in any give time frame. Group work will increase the resources as will coaching increase the resources for the client. In summary to reduce stress, that aim is clearly to ensure that demands are minimised and resource maximised. The greatest demands are internal are much more important than external demands.

### Principles of Emotional Intelligence

These eight principles are taken from Applied EI by Tim Sparrow & Amanda Knight

1. We are each of us in **control** of, and responsible for, our **actions**.
2. No one else can control out feelings.
3. People are different, they experience the world differently, they feel different things and the want different things.
4. However you, and they, are OK.
5. Feelings and behaviour are separate. *Being out of touch with our feelings does not mean being out of control of ourselves and behaviour.*
6. *All feelings are self-justified, to be accepted, and important*
7. *Change is possible*
8. *All people have a natural tendency towards growth and health.*

### Measuring Individual Effectiveness

Underpinning *ie is* the JCA theory of Emotional Intelligence (EI) EI is a combination of skills, attitudes and habits that distinguish superior performance from run-of-the-mill performance both in life as a whole and at work. EI is made up of two parts:

**Intrapersonal intelligence** being intelligent in picking up what is going on inside of us (Self awareness) and doing what we need to do about it (Self management)

**Interpersonal intelligence** Being intelligent in picking up what is going on in other people and between people (Other awareness) and doing what we need to do about that (Relationship management).



Figure 3.2: User Centred Design - Highlighted words of interest (2)

of your home

**ABNLP**

More Info

www.ColinNLP.com

100% Online NLP Certification From the comfort of your home

**ABNLP**

More Info

www.ColinNLP.com

**Affirmations - The Linguistic Technique**

Making use of Affirmations means taking some particular message, such as "I am Confident!" and saying it over and over again, day after day, several times per day - at least three times per day - repeating the phrase endlessly to yourself.

What eventually happens is that it eventually sinks down into your sub-conscious mind and finally your inner self accepts that you are confident. When that happens your life will begin to change because confident people are simply people who believe they are confident. You will begin to act confidently and you will acquire the confidence you seek.

**Hypnosis - The Original NLP Technique**

Using hypnosis effectively bypasses the conscious mind with your message. You get yourself into a nice relaxed state and the message you want to internalize is spoken directly to your sub-conscious mind. The idea is basically the same as with affirmations but it can be a quicker method of accomplishing what you want.

**Natural Programming - NLP the Easy Way**

By carefully vetting and selecting the movies you watch it is possible to engage in some Natural Programming of your own; it's a kind-of NLP but by taking the easy route! You can ensure you pass positive, healthy messages to your subconscious mind if you choose to watch the right movies.

If you like this idea, **The Spiritual Cinema Circle** can help you with its excellent selection of positive and uplifting movies.

**Resources Related to This Subject**

- The NLP Toolbox by Colin G Smith
- 7 Part Affirmations Course
- Program Your Mind for Success

**Articles Related to This Subject**

- NLP Techniques
- How to Program Happiness
- Perception - How We See Things
- Anchoring

**Search White Dove Books**

Find what you want Quickly & Easily



Google Search

Web Site

- ★ Watch Our Inspirational Videos
- ★ Great Free Downloads
- ★ The Master Key System eCourse
- ★ The Personal Development Blog
- ★ Online Personal Development Store



Click on a flag to Translate This Article

Home > Articles > NLP Techniques

Copyright © White Dove Books

Figure 3.3: User Centred Design - Highlighted words of interest (3)

not have appeared in either headings or bold typically happened to be present in the page title. Another point that was noted while observing Carol's browsing behaviour was that she tended to, albeit very occasionally, highlight text on the page as she browsed. When asked if the text highlighted held any significance to her, she responded that she hadn't realised at the time that she was doing that but yes it was definitely text that held slightly more interest than other parts of the page.

From our User Centred Design approach we can show a visual representation of the design requirements for AMS in Figure 3.4

### **3.3.3 Text Analytics**

Requirement: A text analysis system to generate usable data from the information collected as the user browses the web.

A decision has been made from the User Centred Design process about what data would be most beneficial for AMS to collect from the pages browsed, namely headings text, bold text, text from the page title and text which is highlighted as the user reads the page. In addition to collecting this information we learned in the State of the Art that text used to formulate search engine queries is a direct indication of interest by the user and therefore we will also collect this text for our system. Furthermore because the keywords from the meta tags on a page are what the author of the page has written to summarise the page content, we will also use these words.



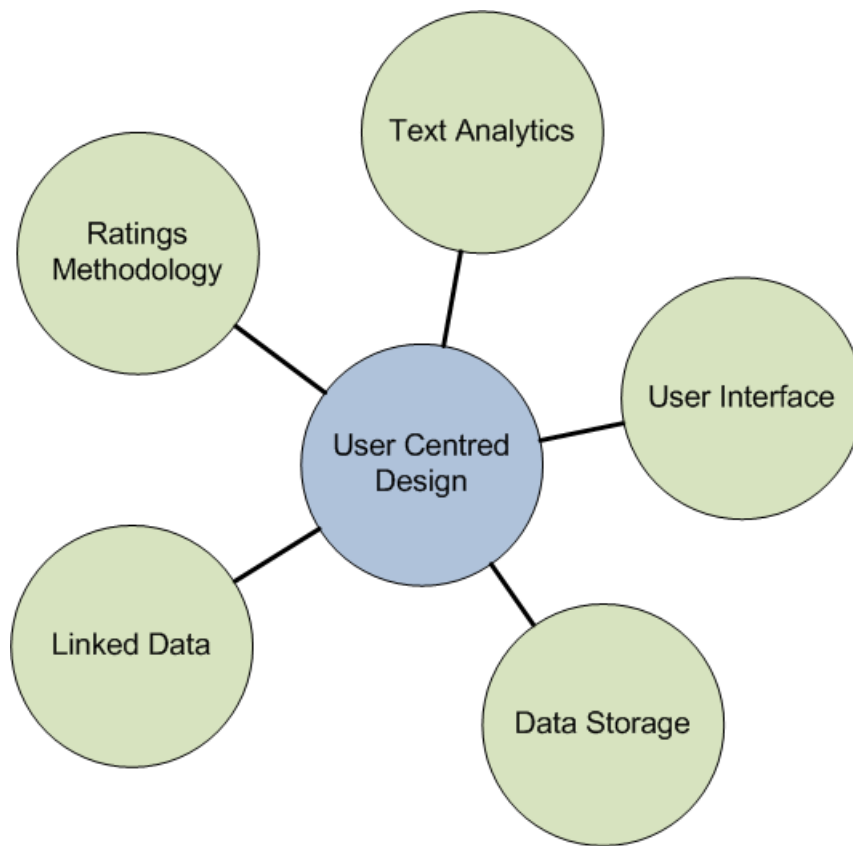


Figure 3.4: Hub and Spoke view of System Requirements

At this point, another issue came to mind that needed to be dealt with and that is the issue of how to deal with websites that the user visits frequently such as social networking sites, news sites etc. We seen from the State of the Art that users tend to revisit websites at a rate of 81%, meaning that four out of five websites have been previously viewed by the user. Therefore, these websites would have the ability to inundate the system with repetitive information and deem the results ineffective. Because of this we will introduce a 'blacklist' of frequently used websites and programmatically exclude these from the updating the user model.

It is necessary to decide how the text collected will be turned into usable data that represents the user's interests. The initial most obvious thing to do is to strip out all the common words, or 'stop words' as they are often referred to. All text should then be changed to lower case and the porter stemmer algorithm [38] used to bring words back to their root form, e.g running becomes run. The frequency of the occurrence of each word should then be counted and the word and the frequency multiplied by the calculated rating as outlined in the next section and Table 1.

### **3.3.4 Ratings Methodology**

Requirement: To formulate and apply a suitable ratings system that can be applied to the data generated by the system and which will generate the most accurate user model possible. This rating system should have the ability to be easily adjusted until the most accurate user model is presented.

With the text gathered from the viewed web pages now turned into usable key terms the next step is to establish some type of rating on these terms. It would be unfair to give the same rating, indicating the same level of interest, to a term that was generated from a page which was only viewed for a few seconds as opposed to a page that was scrutinised over a much longer period of time. Therefore, a method was developed to give an accurate representation of time spent on each page. A high level overview of this calculation is that the clock is started when a user opens a web page by any means available such as clicking a link or entering a URL. The system starts 'listening' for mouse movements of any kind, polling

for five second periods of inactivity. For each five second period of inactivity an 'idle time' counter is incremented by five seconds. When the user exits the page by any exit methods, for example by using the 'x' at the top right corner of the page, by clicking on a link, or by using the 'forward' or 'back' buttons, the clock is immediately stopped and the 'idle time' is subtracted from the total time spent on the page (stop time minus start time) which is calculated in seconds. This method also takes care of such cases where tabbed browsing is being used and the user opens a second tab. The period of inactivity on the first tab will be accurately recorded and used for the calculation of total time on the page giving a more accurate measurement of actual time spent viewing the page.

A second concept needs to be taken into consideration along with the time spent on the page and that is the length of the page in question. It would be unfair to give the same rating to data generated from a 'short' page that was viewed for an equal period of time as a 'long' page. Therefore we perform another calculation, dividing the total length of the page by the available screen height (the visible area of the screen). The total time spent on the page is then divided by the result and this is now deemed to be a fair representation of time across all web pages regardless of size. A pseudocode algorithm to display this calculation is as follows:

```
((Page exit time - Page enter time) - idle time)/  
(Page length/Available screen size)
```

From this point forward we will refer to this result as the 'Time spent on page'.

| Time Spent on page (seconds) | Rating |
|------------------------------|--------|
| Less than 5                  | 1      |
| Between 5 and 10             | 2      |
| Between 10 and 20            | 3      |
| More than 20                 | 4      |

Table 3.1: Ratings Table

Now we take time calculated above and apply the rating. The ratings from Table 3.1 were applied at the outset for the initial evaluations and these could be throttled at any time to work towards the most accurate results possible. The rating figure is multiplied by the frequency of the occurrence of the key words after the text analytics is performed on the data. The rating process is key to the overall success of the system.

### 3.3.5 Data Storage

Requirement: A means of storing data at the client side that facilitates efficient insertion and retrieval and can handle a sizable amount of data. A further requirement is to effectively deal with the issue of maintaining the privacy and integrity of the data.

Looking for a solution to store data the options were a local flat file system or MySQL database which would have been technically difficult for each user to implement, or else to utilise the data storage options offered by the web browser. There is of course one other option and that is the possibility of storing the information on a hosted database service such as the Google datastore. However the data privacy concerns discussed earlier deemed this an option that couldn't even be considered. The local browser data storage is what was decided on and

each of the storage options was investigated. HTML5 are now offering a local web SQL database at the browser and after exploring the alternative options of session storage which wouldn't be capable of holding the data for the necessary period of time, or local storage which is limited to key-value pairs, it was the local web SQL database storage that was decided upon. This gives a familiar but not all-inclusive SQL approach with familiar insert, delete and update statements a possibility.

The option of storing the data at the web browser facilitates the following:

- Fastest possible access as it is local to the user.
- No single point of failure of the overall storage system for AMS.
- Low load requirement per database.
- Solves the privacy issue as data is stored on each user's own computer.

Figure 3 shows a screenshot of the developers tools that are available to administer the different types of web browser storage available, database, local storage, session storage and cookies. These tools can also be used to view the underlying code of the system and any errors which have occurred. On the Google Chrome browser this console can be found by clicking the page icon on the top right hand side of the browser window, followed by developer and developer tools.

### **3.3.6 User Interface**

Requirement: A user interface for displaying information generated by the system back to the user.

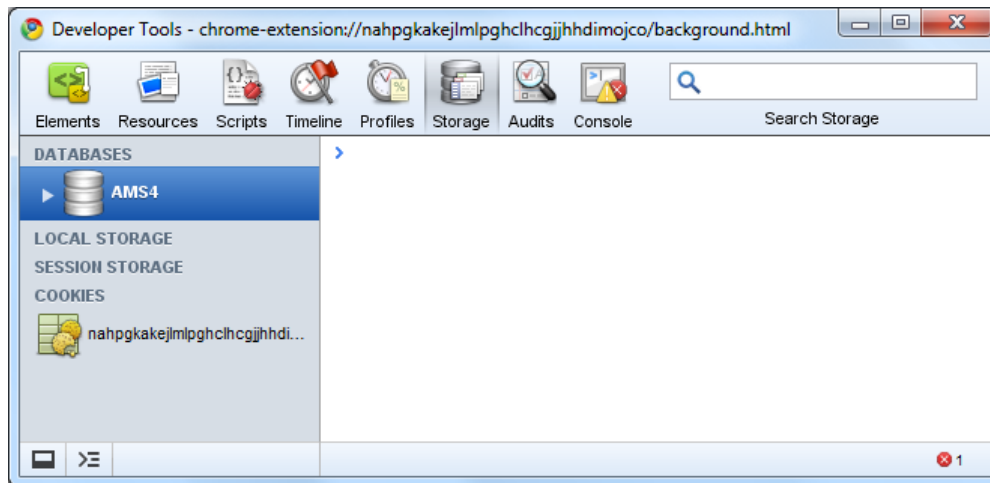


Figure 3.5: Web browser developer tools

Because of the non-invasive nature of the system and the limited amount of time that the user interface will be used, the decision was made to keep it simple and easy to read. After experimenting with a number of different colours we chose a light grey background with a red border as this was conducive to ease of reading and is visually appealing. The user interface will prove to have a much bigger role when the future work is complete as is outlined in the concluding chapter.

### 3.3.7 Linked Data

Requirement: To show the evaluators the possibilities that linked data have to offer in conjunction with AMS

Due to the time constraints of this project the linked data aspect of AMS is not technologically built into the system at present and will be implemented using a sparql query with the aid of the jena library within a java class, as part of the

future work discussed in Chapter 6. However, for the purposes of this dissertation, to have the ability to show the evaluators of the system the results that could be extracted from the linked data set, the key terms will be entered into the 'Precision Search and Find' tool<sup>1</sup> which is part of the Dbpedia linked data set. The linked data results will form part of the material for evaluation.

### 3.3.8 Technology Choices

Requirement: A computer application that can operate in the background as the user browses the web, yet must have some means for the user to interact with the application when so desired.

The decision has been made due to privacy concerns that the application must reside at the client side. The choices for the technology to use were fairly limited with the options being something that works as a proxy between the user and the internet or else some form of add-on or plug in at the browser. A web browser plug in was decided on due to the desire for the end product to be something that the user could see but that doesn't interfere in any way with their browsing experience. This aim was achieved in the final version of AMS as the only indication of the system is a small logo at the top of the screen which can be clicked to view the output of the analysis performed. Two types of plug-in technology was looked at and experimented with, Google Chrome extensions and Firefox Add-ons. Ease of development, smooth performance, the instant installation versus having to restart the browser are what inspired the choice of Google Chrome as the one to use and despite the fact that there isn't as extensive a documentation set for

---

<sup>1</sup><http://lod.openlinksw.com>

Google Chrome extensions as there are for Firefox Add-ons, due to the newness of the Google Chrome extension technology, it was relatively simple to follow the instructions to set up a basic extension. Google Chrome extensions are written using various web technologies such as HTML and CSS. In addition to its own API, they support the full use of the JavaScript language and all of the various JavaScript libraries.

### **3.3.9 Processing platform**

Requirement: We required a processing platform to run the java web application which performs the text analytics for the system. After extensive research we chose the Google App Engine for the following reasons:

1. It provided functionality required.
2. It is a stable platform.
3. It is intuitive and relatively simple to implement
4. It provides scalability.
5. It is managed.
6. Administration is easy.
7. It is the most cost effective.

After reading the documentation and working through some tutorials and examples we had a good grasp of how to implement the service. A full description of the implementation can be found in the next chapter.



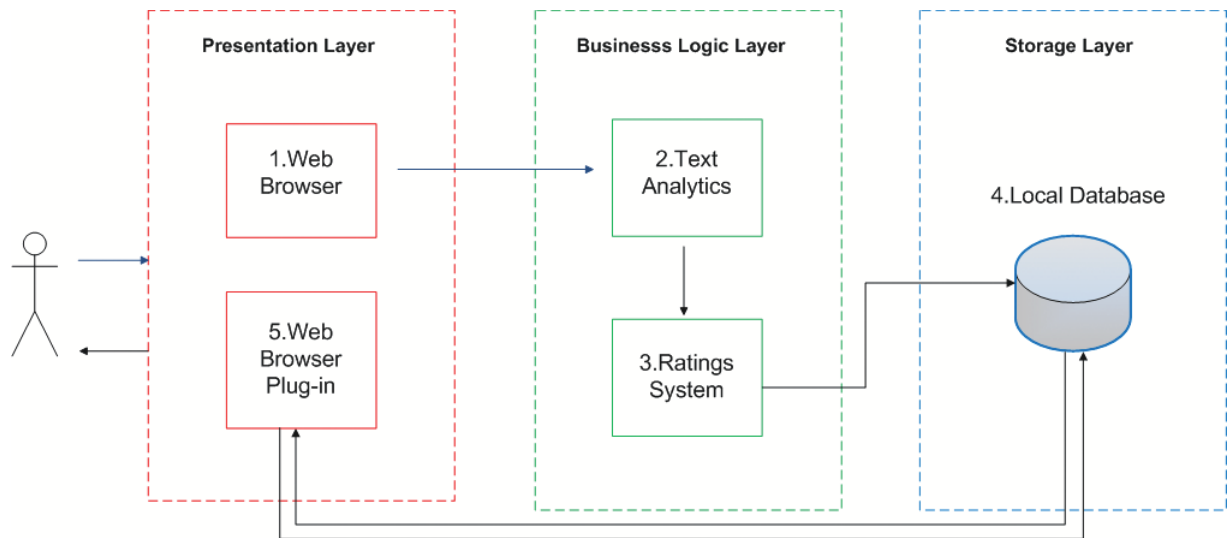


Figure 3.6: High Level Architecture

## 3.4 Architecture

The high level architecture can be seen in Figure 3.6 This is followed by a step by step walk through of the system. Each component is discussed in detail in the next chapter.

Following the process from start to finish in conjunction with the numbering on the diagram.

1. User opens web page, starts clock in background. User closes web page, clock stops.
2. Text is collected from the web page and goes through the text analytics program and is transformed into usable data.
3. Data is subjected to the ratings system as outlined in 3.3.4
4. Rated data is stored in the local web storage database

5. User clicks on web browser plug-in, the database is queried and the top rated data is displayed to the user form the popup of the plug-in.
6. The top rated keywords are entered into linked data for evaluation purposes. This step is not shown in the diagram due to it not being technologically included in this version.

### **3.5 Summary**

In this chapter the reader is introduced to the User Centred Design process that was used to feed the design of the system. The requirements of the system were profiled and the thought process that was used to tackle each one. There is a detailed account of how all the requirements were met in the overall design. The architecture of the system is outlined and there is a graphical representation of the flow of information with respect to the user.

# Chapter 4

## Implementation

### 4.1 Introduction

This chapter outlines the implementation of AMS, and how the design choices from chapter 3 were integrated into the final system. The low level architecture and components of the system from the point at which the user opens the browser is discussed and the path taken by the information as it makes its way from collection at the presentation layer, to processing at the business logic layer, on to the storage layer and finally back to the user for scrutinising. Figure 4.1 shows a high level overview of the system.

### 4.2 Development

The development of AMS was carried out on a HP Pavilion m7360n PC with a Pentium D processor, 3.0 GB of RAM installed and 2.8 GHz processing speed.

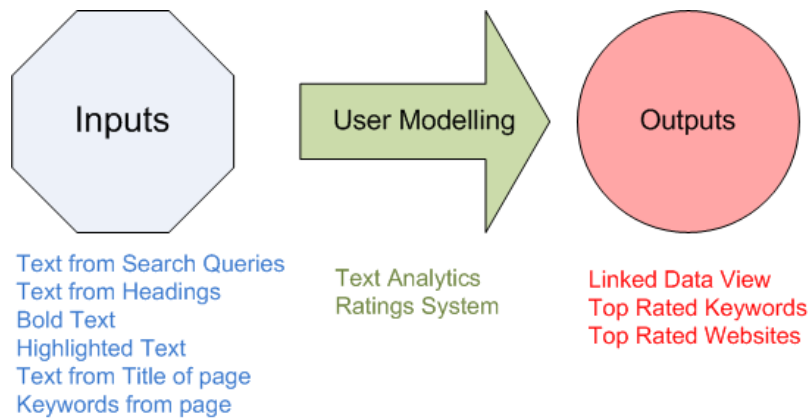


Figure 4.1: High Level Overview of AMS

The technologies used in the implementation are Java 1.6 supported through the Eclipse IDE, JavaScript, Ajax, JSON, HTML, CSS, SQL, HTTP, Google App Engine, Google Chrome Extension API and the Google Chrome Web Browser.

## 4.3 Software Implementation

### 4.3.1 Google Chrome Extension

At the outset the initial challenge was to get a simple Google Chrome extension up and running. Starting off with the Hello World example from Google's extension tutorials, we learned that there are two key files that make up the framework of an extension. A manifest json file is at the core of all Google chrome extensions along with a HTML file. In addition to these two mandatory files there can be any number of other files such as JavaScript or images etc. The manifest.json file holds all the information about the extension. Figure 4.2 is the manifest file for the most recent version of AMS.

```

{
  "name": "AMS",
  "version": "0.4",
  "background_page": "background.html",
  "permissions": ["history", "tabs", "http://**/*", "https://**/*",
  "http://irish59575.appspot.com/"],
  "icons": { "16": "icon16.png",
             "48": "icon48.png",
             "128": "icon128.png" },
  "browser_action": {
    "default_title": "AMS User Model4",
    "default_icon": "icon.png",
    "popup": "popup.html"
  },
  "content_scripts": [
    {
      "matches": ["http://**/*"],
      "js": ["contentscript.js"]
    }
  ]
}

```

Figure 4.2: AMS - manifest.json

To give a brief outline of Figure 4.2 from the top, the manifest file holds the name and version number of the extension and this information is shown to the user on the screen that shows which extensions are installed and these can be seen at <chrome://extensions/> using the Google Chrome browser, Figure 4.3

The background page is a key part of the extension and will be discussed in the next section, this indicates what the background page is called and where it is located. Permissions, as the name implies, allow the extension to do things such as access the user's web browsing history, inject JavaScript code programmatically into the viewed pages and make cross-origin XMLHttpRequests. The manifest file also contains the location of the various sized icons which are used to identify the

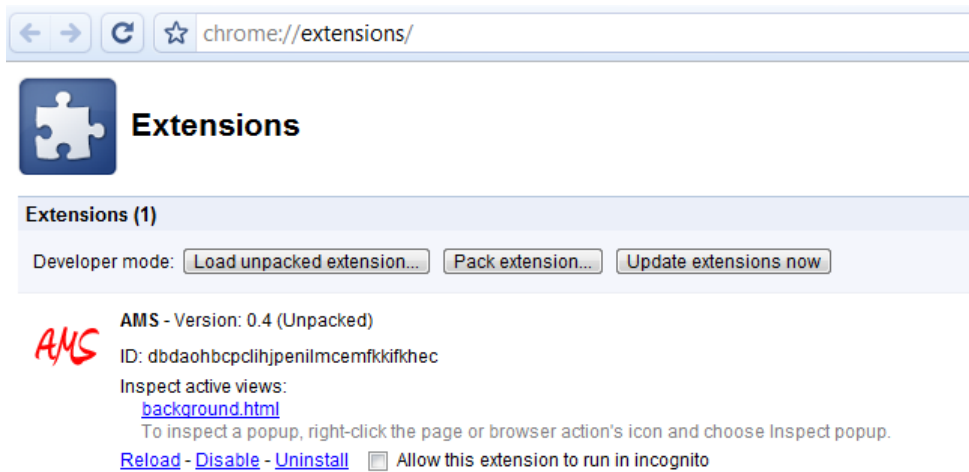


Figure 4.3: Google Chrome Extension Installation Page

extension in the browser bar and also on the installation page. The browser action indicates the name of the extension to be shown when a mouse is hovered over the extension icon on the browser bar and what HTML code to display when the icon is clicked, in AMS this is 'popup.html'. Finally the content script is necessary in cases where it is necessary to run a JavaScript file in conjunction with the background page. AMS runs the file 'contentscript.js' every time the extension is activated and this will be discussed in the next section. Figure 4.4 is a visual representation of AMS Google Chrome Extension

With some trial and error we got the 'hello world' extension up and running. The next step was to get the extension to interact with the HTML5 web SQL database. Initially we mistakenly placed the following code in the contentscript.js file

This was setting up the database correctly but we found that there was a new

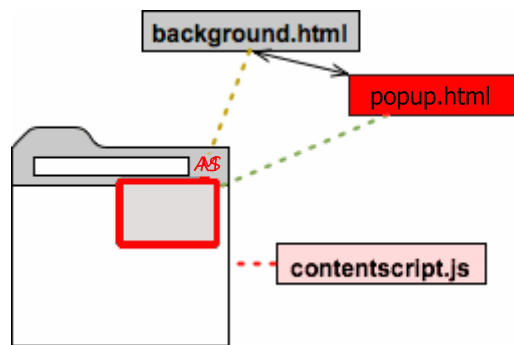


Figure 4.4: AMS Google Chrome Extension

```
function connectToDB()
{
dbObj = openDatabase('AMS4', '4.0', 'AMS Database', 2 * 1024 * 1024);
}
connectToDB();
```

Figure 4.5: Web database setup

database being set up for each web page that was visited instead of a one overall database for the entire extension. After some reading and investigating we realised that the code from Figure 4.5 should have been in the background.html file. The purpose of the background.html file is to solely work in the background and provide a base for the extension and a place to put code which is needed to interact with the other components of the extension.

Now with the database set up for the extension the next step was to attempt to potentially interact with the web page being viewed by the user. On researching the methods available from JavaScript and the Document Object Model (DOM) API's we experimented with a simple 'getElementByTagName' call. Recalling from Chapter 3 that AMS will collect the following from each web page visited:

- Text from the title of the page
- Keywords of the page
- Text that forms a search query
- Text in bold font
- Text from all headings on a page
- Any text that is highlighted as the user browses.

For the purposes of this walk through of how AMS was implemented we will follow one aspect of the information that is collected and so we focus on how the bold text is processed. Figure 4.6 shows a snippet of code from `contentscript.js` that is used to collect any text in bold lettering from the viewed web page. Because the background page cannot interact with the web page directly, this is where the `contentscript.js` file is needed.

```
//Get all the 'bold' text from page
var bold = new Array();
var bolditems = new Array();
function getBold(){
    bold = document.getElementsByTagName('b');
    for (j=0; j<bold.length; j++)
        bolditems = (bold[j].innerText);
    return(bolditems);
}
```

Figure 4.6: Collecting text in bold lettering from web page

Once the text has been collected, it is now necessary to send the result over to the `background.html` page since it is from there that access to the database is



facilitated and any other processing that is required is also instigated from here. Passing the data to the background page is done via the message passing API. Following on with our example in Figure 4.6 the code to pass the bold text from contentscript.js to background.html and be seen in Figure 4.7

```
//send info over to background.html
chrome.extension.connect().postMessage(
    {
        "boldtext" : getBold(),//calls the getBold() function
    });
}
```

Figure 4.7: Message passing with Google Chrome Extension

On the receiver side, in background.html the code to receive the message is shown in Figure 4.8

```
//receive info from contentscript.js
chrome.extension.onConnect.addListener(function(port){
port.onMessage.addListener(function(msg)
{
    boldtext = msg.boldtext;
}
});
```

Figure 4.8: Message received

At this point it is time to send the text to the java service for processing. The java service is made up of three java classes, Text.java which creates a text object which is made up of a word and a frequency, TextService.java where the text processing happens and AMSTextServlet.java which is used to interact with the Google App Engine application server which will be discussed in the next section.

### 4.3.2 Google App Engine

The text analytics java service for AMS is hosted by the Google App Engine (GAE) which is a cloud computing infrastructure for creating and hosting web applications. GAE supports the Java Servlet API which is what AMS uses to interact with the server. GAE also offers a nice feature called the datastore for storing data. At one point during the development of AMS we investigated the possibility of storing the collective data generated from all users in one location and we implemented an instance of the Google datastore to realise what possibilities were on offer. The datastore is not a relational database and all data is retrieved and stored as entities. The datastore offers great potential in terms of scalability, performance, replication and load balancing but was a little tricky to implement using the Java Persistence API and GQL for querying. Due to privacy concerns, having an overall datastore available in the final version is not an option for this research but it should be known that it is a definite possibility and can be looked at for future work possibly within some larger study in this field.

It is relatively easy to implement GAE using the Eclipse IDE and the Google Plugin for Eclipse. Users with a Gmail username and password can sign up quickly by accessing <https://appengine.google.com/> and registering a new application which will generate the application id. The application console Figure 4.9 provides a means to view usage data, quota details, logs, access the datastore etc.

Once the java classes and the java servlet classes were built, the application was deployed to GAE by means of a convenient deployment option within Eclipse

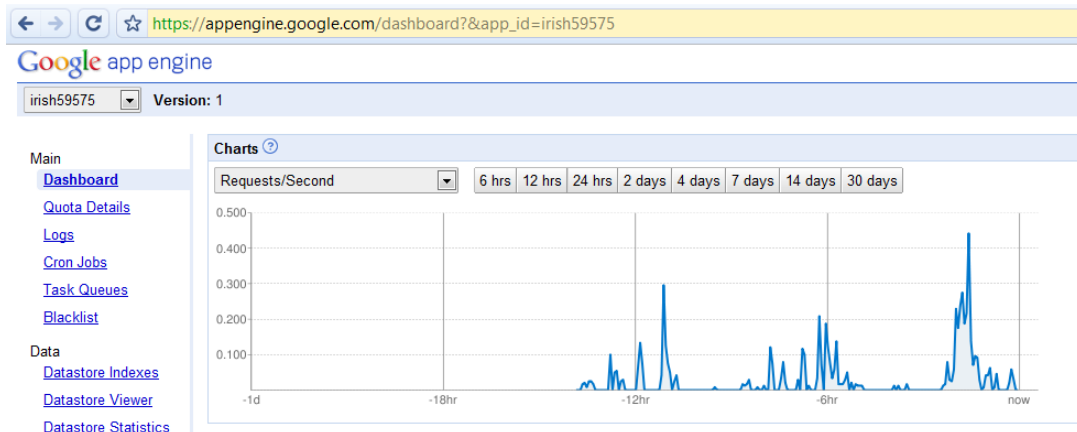


Figure 4.9: GAE Application Console

to upload the application. The java servlet class, `AMSTextServlet.java`, interacts with GAE by means of a `doGet` method. Figure 4.10 shows an extract from `AMSTextServlet.java` and shows how the servlet interacts with `TextService.java`

```
public class AMSTextServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws IOException {
        resp.setContentType("text/xml");

        String textString = (String)req.getParameter("text");
        ArrayList<Text> _results = null;
        if (textString != null) {
            _results = TextService.getInstance().analyseText(textString);
        }
    }
}
```

Figure 4.10: `AMSTextServlet.java`

### 4.3.3 Java Service

Along with `AMSTextServlet.java`, as mentioned earlier the java service is also made up of `Text.java` and `TextService.java`. `Text.java` create an object of type `Text` which is made up of a word and a frequency. `TextService.java` contains all of the text processing code for the application and we will explore that now.

`TextService.java` (Appendix A) comprises a constructor and some methods. The first method `analyseText` takes in the words to be analysed as a parameter, splits them into tokens, changes them to lower case and immediately passes to the next method `wordFilter` which strips out all the 'stopwords' or commonly used words, by means of pattern matching each word against the words in `stopwords.txt`, a text file of common words that we put together for the system, which is read in line by line, and deletes any matches found. These words are kept in an external file, as opposed to an array, for ease of manipulation of the file at a later time if words need to be added or taken away. The remaining words are sent back to the `analyseText` method where they go through the stemming process using the Porter Stemmer algorithm [38] to change them to their root form. The next step is to place each word in a hash map, then at each iteration, check if the word is already there and if found the frequency is incremented by one. The words and their frequencies are added to an array list of `Text` objects and these are the results which are sent back to the google chrome extension when the `XMLHttpRequest` is made. Figure 4.11

Meanwhile, returning to `background.html` which now has received the pro-

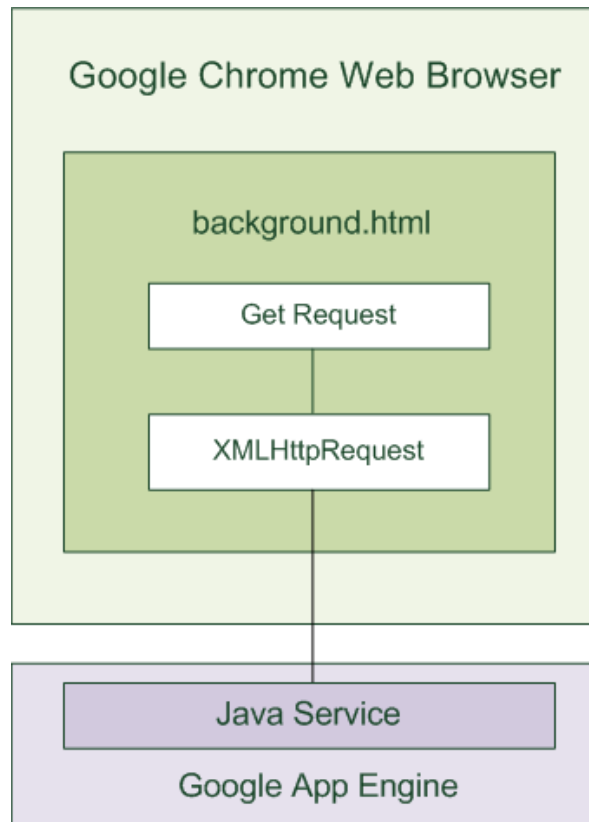


Figure 4.11: Interaction between AMS and GAE

cessed text and the corresponding frequency it is now time to apply the ratings methodology as outlined in 3.3.4. The frequency of each word is multiplied by the calculated rating corresponding to how much time the user spent actively looking at the page that the word originated from, while taking into consideration the length of the page.

#### 4.3.4 Data Storage

We showed in 4.3.1 how the HTML 5 web storage database was set up, now it is time to set up our schema and enter the results of our findings into the database.

We chose a simple schema with just two tables, one to hold the keywords and their frequencies and another to hold the web page title and the related rating, from 3.3.4, corresponding to how long the user spent viewing the page. Creating the tables and inserting the results into the database is done by means of SQL statements as in Figure 4.12

```
tx.executeSql('CREATE TABLE IF NOT EXISTS keywords
(word TEXT PRIMARY KEY ON CONFLICT IGNORE, frequency INTEGER)');

tx.executeSql('CREATE TABLE IF NOT EXISTS sites
(site TEXT PRIMARY KEY ON CONFLICT IGNORE, visits)');

tx.executeSql('INSERT INTO keywords(word, frequency) VALUES
('+parsedResults[0]+'", 0)');

tx.executeSql('UPDATE keywords SET frequency = frequency +
'+parsedResults[1]*confidence+' WHERE word LIKE "'
+parsedResults[0]+'");
```

Figure 4.12: Sample of SQL used to set up database tables and insert results from text analysis

### 4.3.5 User Reflection

The final part of the implementation is to discuss how the data is reflected back to the user. Recalling in 4.3.1 the manifest.json file holds a value for 'browser action', this is what happens when the user clicks the AMS icon in the browser bar, see Figure 4.13. In AMS this is set to display popup.html which contains the CSS layout for the extension, Figure 4.14 which creates the grey background and the red border.

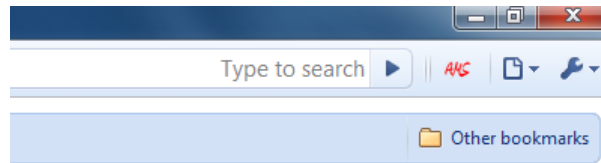


Figure 4.13: Screenshot of AMS

```
<style>
body {
min-width:400px;
overflow-x:hidden;
}
#content {border: #FF0000 solid 6px; border-radius: 6px; background:
#eeeeee; margin-left:auto; margin-right:auto; padding: 10px;}
</style>
```

Figure 4.14: CSS to create AMS User Interface

Popup.html also controls what is displayed on the popup. In AMS we decided to show the user the top three websites, and their ratings and the top five keywords, and their ratings, that AMS deemed to be of most interest. This requires a simple SQL call to the database. Figure 4.15

```
tx.executeQuery('SELECT DISTINCT word, frequency FROM keywords
ORDER by frequency DESC LIMIT 5', [], function (tx, results){
    for (var i = 0; i < results.rows.length; i++){
        keys[i]=(results.rows.item(i).word);
        numKeys[i] = (results.rows.item(i).frequency);
    }
}
```

Figure 4.15: SQL to select the top keywords and their ratings from the dataset

However it is not possible for popup.html to directly communicate with the database as the database is setup and administered from background.html, and therefore it is necessary for popup.html to communicate with the database via

background.html. This can be done using the Google Chrome Extension API which allows direct communication with background.html from anywhere within the extension, with a 'chrome.extension.getBackgroundPage()' call as can be seen in Figure 4.16

```
document.write("<h2>Top Keywords That Best Describe  
Your Interests</h2>");  
for (var j = 0; j < chrome.extension.getBackgroundPage()  
.keys.length; j++){  
document.write("<p>" + chrome.extension.getBackgroundPage()  
.keys[j] + " :  
<b>Earned a rating of " + chrome.extension.getBackgroundPage().  
numKeys[j] + "</b></p>");  
}
```

Figure 4.16: Popup.html code to get and display the Top Keywords to the user

### 4.3.6 Linked Data

Now that the data is reflected back to the user, as a further step for the evaluation, we wish to view what would be returned from the Dbpedia linked data set for our top keywords. We experimented with using a Sparql query using Java and the Jena Library but due to time constraints this part of the project is not technologically implemented. Instead, we use the Dbpedia Precision Search and Find tool to display results from linked data as can be seen in the Chapter 5.

## 4.4 Implementation Issues

All in all the implementation of AMS went relatively smoothly. In addition to those mentioned already, some of the issues that earlier versions highlighted were:



- The difficulty of capturing 'clean' data. When attempting to scrape text from a web page we are at the mercy of the quality of the html that is used to author the page. There tended to be incidents of non-letter character such as colons, semi-colons, question marks etc appearing in the database. This issue was addressed in the latest version by using regular expression pattern matching to replace all non-letter characters with an empty space ("");
- Another similar problem arose with trying to capture the keywords from the page. Sometimes the keywords are separated by a comma and sometimes by a space. Using the same code to capture both of these possibilities was resulting in words being bunched together in the database deeming those words unusable.

## 4.5 Conclusion

In this chapter we gave an overview of AMS and the challenges that were faced during the development and implementation stages. We outlined how the AMS user modelling system works and how it was implemented and we showed snippets of code throughout to give the reader a sense of how all the pieces fit together.

# Chapter 5

## Evaluation and Results

### 5.1 Introduction

The key to the success of building an accurate user modelling system is the satisfaction of the user and most importantly to what degree their interests have been inferred. The purpose of this evaluation is to test the functionality, usability and the overall appeal of the implemented Google Chrome extension as well as to investigate the potential benefits of utilising the linked data set to return more relevant results to the user. Six users were chosen to perform the main evaluation of the system and three of them were asked to choose a domain to work within. The three domains chosen were Life Coaching, Sport and Fashion. The remaining three evaluators will work in the open domain.

## 5.2 Evaluation

### 5.2.1 User Centred Design Evaluation

For the first part of the evaluation we returned to our User Centred Evaluation Subject (as described in Chapter 3). This allowed for continuity of the evaluation process. In the Design phase our subject 'Carol' assisted with which aspects of the visited web pages were deemed most valuable to capture and analyse, which were the title text, the text from headings, text in bold lettering and text which is highlighted as the user browses the web page. In addition to this, as seen in the State of the Art, the text from web search queries are also a direct expression of interest by the user and therefore these are also captured. Furthermore we also capture the keywords from the meta tags on the page as these are the words that the author of the page uses to summarise the content of the page and therefore these words should be of direct interest to the user if in fact the user expresses interest in that page. For this next part of the evaluation, AMS was installed on Carol's computer and she was asked to re-visit the three web pages that she had previously highlighted at the outset of the experiment. Figures 3.1,3.2, and 3.3. After browsing these pages as normal AMS produced a reflection of Carol's browsing as seen in Figure 5.1.

Looking back at figures 3.1, 3.2 and 3.3, we can see that each of the words that rated highest in AMS were words that Carol highlighted as being of interest to her. When she was asked if the rating of the websites were a reflection of both her interest and of the time spent reading the page, she very much agreed that this was the case. For the next part of the evaluation, we uninstalled and re-installed

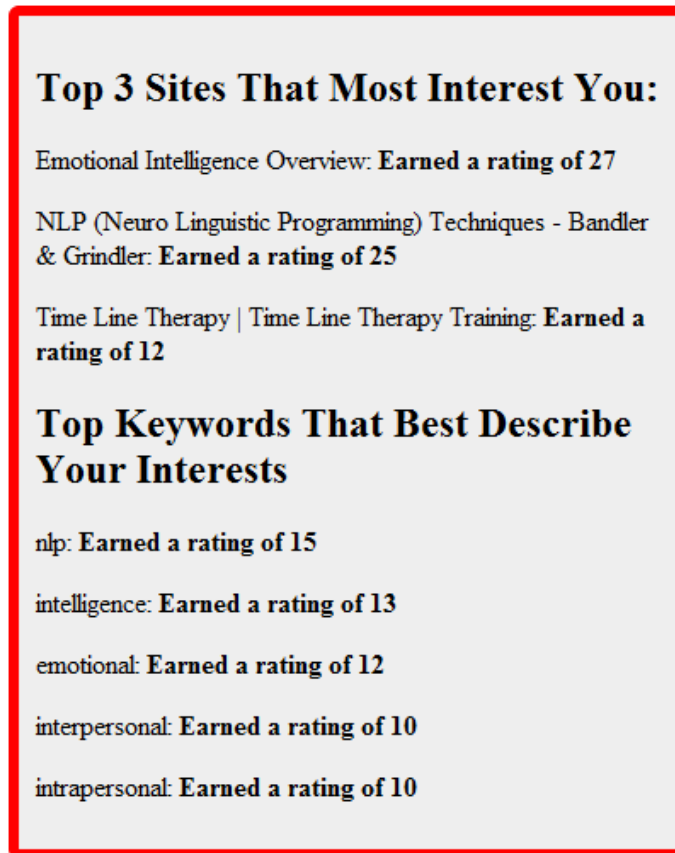


Figure 5.1: User Centred Design - Preliminary Results from AMS

AMS to clear down the database and we asked Carol to browse the web for 30 minutes inside her chosen domain. Figure 5.2 shows her browsing history for that 30 minute period.

At the end of the 30 minutes AMS produced the results as in Figure 5.3. We asked Carol to fill out the evaluation questionnaire and the results are included with the other evaluations carried out and can be seen in the next section.

























- 3:49 PM  [Hypnosis Scripts Library | Hypnosis Scripts Downloads](#)
- 3:48 PM  [Transform lives with the Break Bad Habits Hypnosis Script](#)
- 3:47 PM  [Self Hypnosis | Subliminal Self Help | Hypnosis Downloads](#)
- 3:47 PM  [Develop Your Personal Power](#)
- 3:47 PM  [Amazon.com: Awaken the Giant Within : How to Take Immediate Control of Your Mental, Emotional](#)
- 3:46 PM  [self hypnosis - Google Search](#)
- 3:46 PM  [nlp hypnosis - Google Search](#)
- 3:46 PM  [nlp seduction - Google Search](#)
- 3:45 PM  [HYPNOSIS.COM](#)
- 3:45 PM  [HYPNOSIS.COM](#)
- 3:45 PM  [HYPNOSIS.COM](#)
- 3:44 PM  [nlp books - Google Search](#)
- 3:43 PM  [The Law of Attraction Visualization by Darryl Howrie](#)
- 3:42 PM  [Alter Egos Can be a State of Mind to Cope with Massive Success:](#)
- 3:42 PM  [Dr Rick Wins ARIA Music Award](#)
- 3:42 PM  [Dr Rick Collingwood's - Hypnosis Certificate Online](#)
- 3:40 PM  [6 Questions to Tell if they're Lying](#)
- 3:40 PM  [Overcoming Fear Strategies II](#)
- 3:35 PM  [A Powerful Public Speaking Intro by D Howrie](#)
- 3:27 PM  [NLP Training Courses Online](#)
- 3:26 PM  [nlp coaching video - Google Search](#)
- 3:26 PM  [NLP Video](#)
- 3:21 PM  [Time Line Therapy | Time Line Therapy Training](#)
- 3:18 PM  [spirituality - Google Search](#)

Figure 5.2: User Centred Design - Browsing History

The top two rated keywords ('hypnosis' and 'nlp'), followed by the second pair of top rated keywords ('intelligence' and 'emotional') were then manually entered into the 'Precision Search and Find' tool<sup>1</sup> which is part of the Dbpedia linked data set. The two sets of results in Figure 5.4 is what was presented to Carol and her reaction was one of incredulity, stating that this was the information that she had been looking for all along and that all of the non related results that she was

---

<sup>1</sup><http://lod.openlinksw.com>

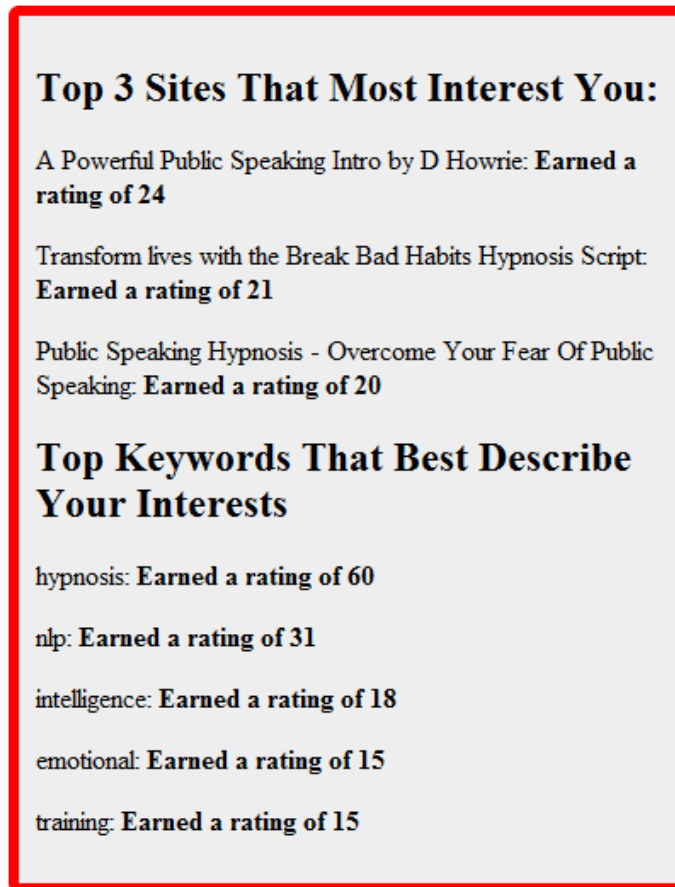


Figure 5.3: User Centred Design - Results from AMS

accustomed to getting from regular search engines were not there.

This concludes the User Centred Design process for this work. All in all, it was a successful experiment which fed the design process to promote which aspects of a user's browsing were important to capture.

The image displays two screenshots of a query interface from OpenLink Software. Both screenshots show a table of results for a specific query. The top screenshot is for the query "Entity1 has any Attribute with Value 'hypnosis nlp'". It shows six results, each with a URI, a name, and a summary. The bottom screenshot is for the query "Entity1 has any Attribute with Value 'emotional intelligence'". It shows 20 results, each with a URI, a name, and a summary. The interface includes a header with the OpenLink Software logo, a query description, a link to view the query as SPARQL, and a pagination indicator.

Figure 5.4: User Centred Design -Results from Linked Data environment

## 5.3 Results from Evaluation

As stated earlier in this chapter, we used six users to evaluate the user modelling system. There were two male and four females from a variety of backgrounds, one from IT, two from Business, one Life Coach (our User Centred Design subject), one student and a stay at home Dad. Four of the users fall in the 35-45 age bracket, one in the 18-25 bracket and one in the over 45 bracket. All users are known to the author and donated their time to evaluate the project free of charge. The users

carried out their evaluation separately and they were each given a brief summary of the system ahead of the evaluation with an emphasis placed on what type of data is collected and how it is only stored at the browser side and will be used for the purposes of the evaluation only.

The first three users to carry out their evaluation were asked to choose a domain to work within. The domains chosen were Life Coaching, Sport and Fashion. All evaluations were carried out on the user's own computer on which AMS was installed, after the briefing of the system was carried out. Users were asked to browse the internet within their chosen domain, or in the open domain if they were not initially asked to choose a domain, for thirty continuous minutes while AMS worked silently in the background. Users were asked to simulate their normal web browsing behaviour.

### **5.3.1 Pre Evaluation Questionnaire Results**

For clarity of outlining the results of the evaluation we will adopt the method used by Conlan et al. in [46] as an easy to comprehend method of displaying the results. Prior to commencing with the study, each user was given a pre-evaluation questionnaire (Appendix B) and the questions and answers are shown in Figures 5.5 to Figure 5.10 .

It might be of interest to note from Figure 5.5 that the users who spend more than twenty hours a week on the internet are the IT person, the student and the stay at home Dad.



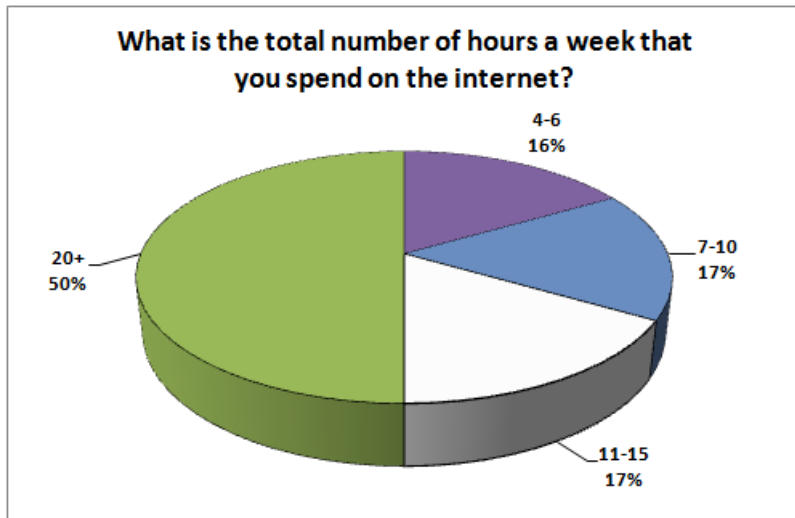


Figure 5.5: Weekly time spent on internet

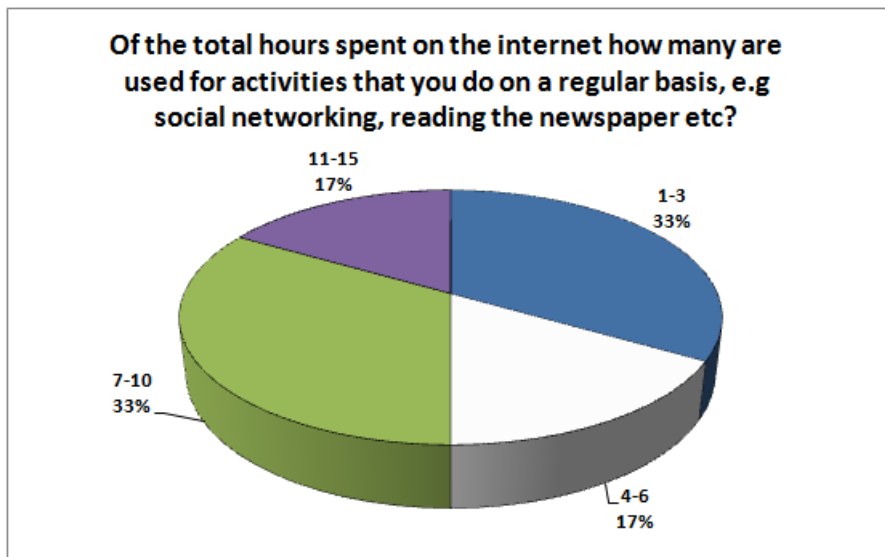


Figure 5.6: Weekly time spent on repetitive tasks, social networking etc

From Figure 5.6, the majority of users either fell into 1-3 hours or the 7-10 hours bracket and this tended to account for less than 50% of their total browsing

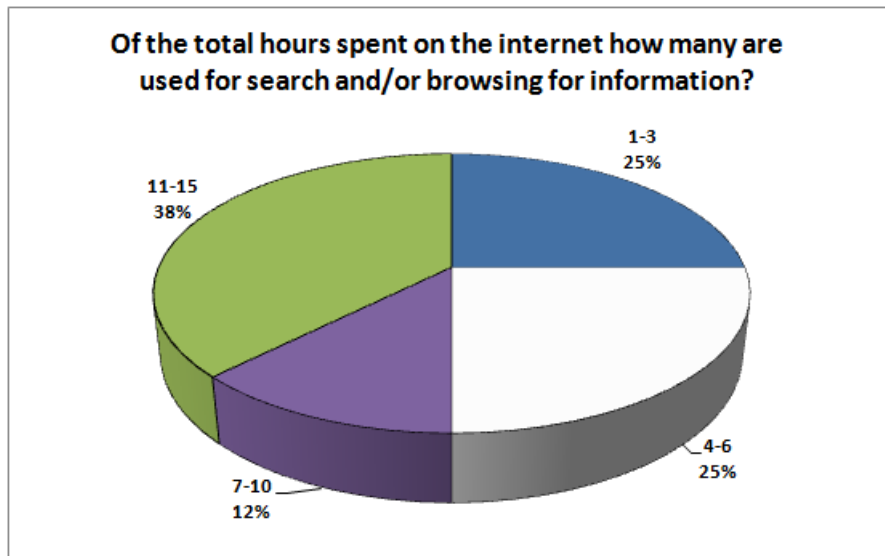


Figure 5.7: Weekly time spent browsing/searching for information

From Figure 5.7 we can see that the majority of users spend between 11 and 15 hours a week searching or browsing for information which accounts for more than half of their total browsing time.

The remaining three questions in the pre-evaluation questionnaire are centred on the user's experiences while browsing the internet. Figure 5.8 shows that the majority of users agreed that pop-ups annoyed them while browsing and this is consistent with the findings in the State of the Art when the Lumiere project [14] was discussed.

Interestingly as shown in Figure 5.9, none of our users like to fill out profile building questionnaires while browsing the internet which gives merit to the non invasive approach of AMS.

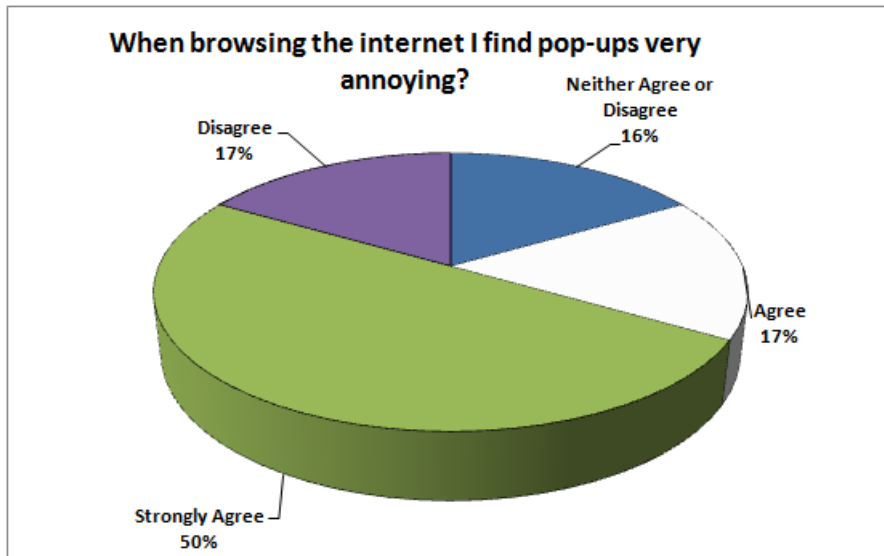


Figure 5.8: Reaction to popups

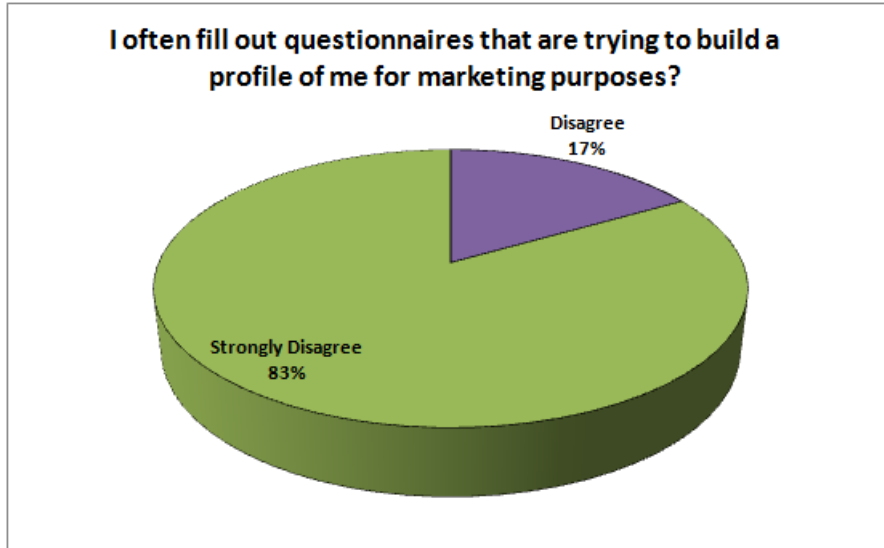


Figure 5.9: Reaction to profiling for marketing purposes

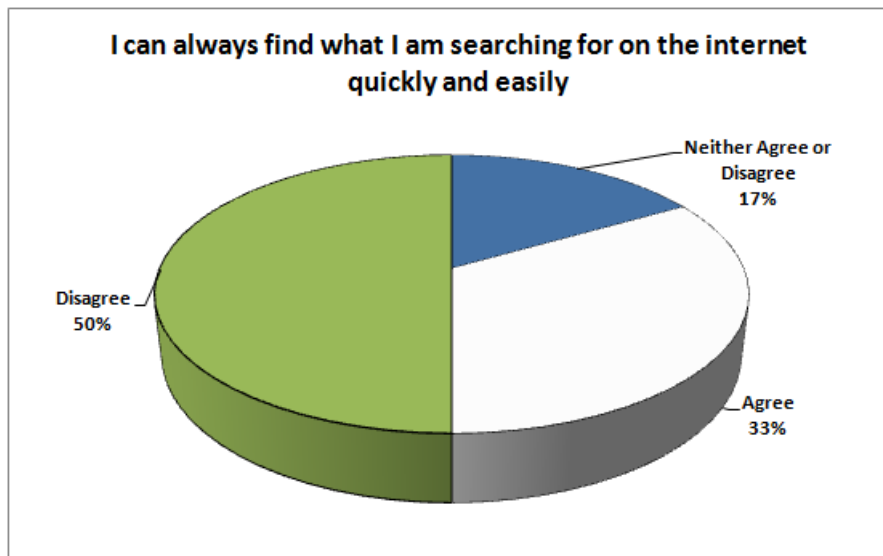


Figure 5.10: Experiences of searching for information

From Figure 5.10 we can see that half of our users have difficulty in finding what they are looking for when searching the internet. What is worth mentioning here is that none of our users chose the strongly agree or strongly disagree options that were available. This would lead us to believe that even if people are happy with their ability to perform successful searches on the internet there is still a missing link where they are not completely satisfied. Also in direct contrast to this if users are unhappy with their search ability this survey would indicate that they are not completely unhappy.

### 5.3.2 Post Evaluation Questionnaire Results

On completion of the thirty minutes of continuous browsing, users were shown the results of AMS and were asked to study them for a few minutes. They were then asked to fill out a post evaluation questionnaire (Appendix B).

The first three questions prompted a unanimous response, whereby all users found AMS easy to use and that it didn't have a negative impact on their browsing ability. Significant to point out at this time is that no users found AMS intrusive in any way as they browsed the web. This is directly related to one of our overall objectives, to build a non-invasive user-model, and we can conclude from the response to this question that we have adequately met that requirement.

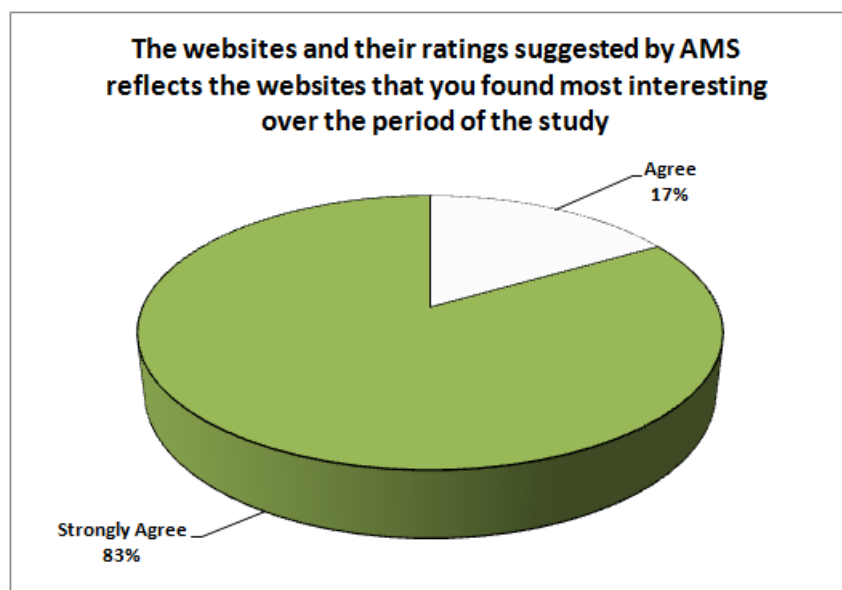


Figure 5.11: Reaction to AMS website suggestions

All users agreed, and most strongly agreed, Figure 5.11, that the websites and their ratings as shown by AMS were symptomatic of the websites that they found most interesting during the period of the study. They were asked to particularly note the ratings. Two users verbally clarified before answering the question as to whether this meant the websites that they had spent most time on and they were told that no, the question is concerned only with the websites they had most

interest in.

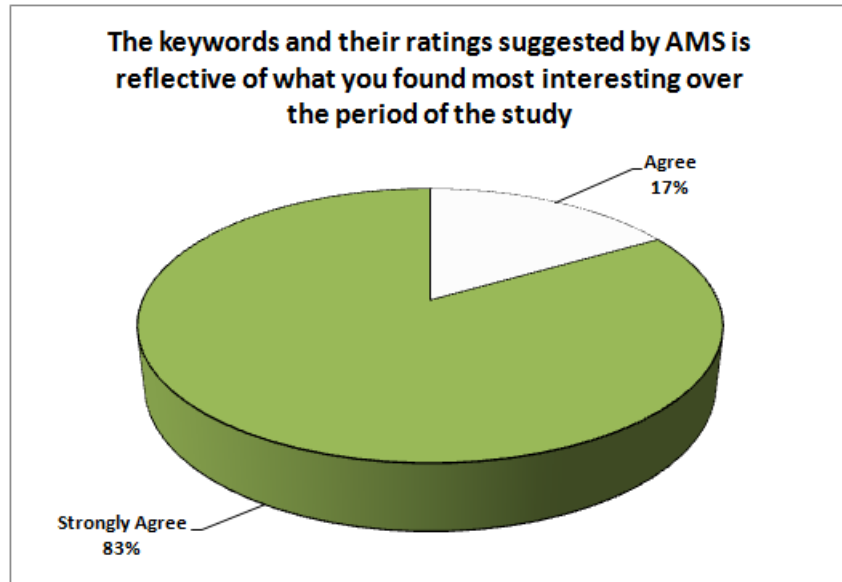


Figure 5.12: Reaction to AMS keyword suggestions

Interestingly the question posed in Figure 5.12 prompted an identical answer from all users as the previous question, which would lead us to believe that both the websites and the keywords suggested by AMS were indicative of the user's interests. In both cases all users bar one strongly agreed that these were the websites and keywords that they were most interested in. The answers to these two questions are a direct indication of the success of the project since the core objective is to construct a user model that can infer the user's interests as they browse the web.

In Figure 5.13, when asked if they would like to see more information when AMS is clicked on, most of the users agreed that they would. The following

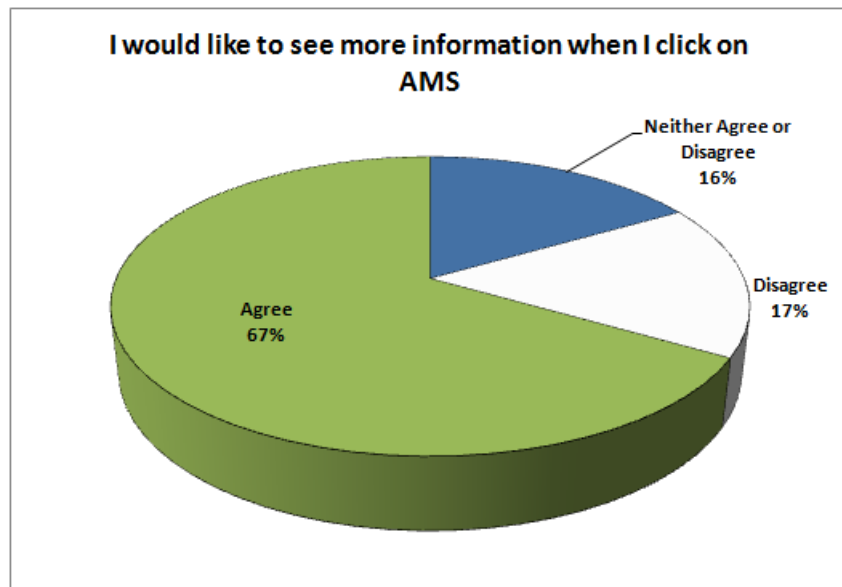


Figure 5.13: Reaction to AMS content

comments were made:

- Maybe the information could be segregated under certain topics
- A link from AMS to the linked data would be a good idea
- Some suggestions of websites that I might be interested in would be nice.

Figure 5.14 relates to the information that was shown to the user from the linked data environment, specifically from the Precision Search and Find tool<sup>2</sup> offered by Dbpedia

It is at this point that the difference is clearly shown between the users that conducted their evaluation using a chosen domain and those who used the open domain. The users in their chosen domain all strongly agreed that the information

<sup>2</sup><http://lod.openlinksw.com>

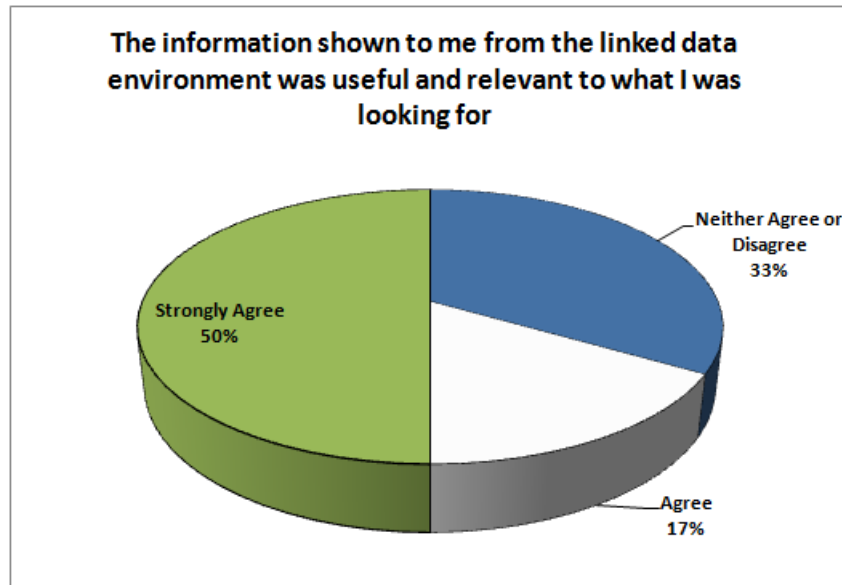


Figure 5.14: Results from Linked Data

shown to them from the linked data environment was useful and relevant to what they were looking for. We can go as far as to say that this part of the evaluation generated some excitement from the users as they had never seen or heard of linked data before and they were pleasantly surprised by the way that all the relevant information was there without everything else that they are used to seeing from regular searching. Some comments which were made about the results from linked data:

- I liked the way it brought up summary screens of something that was relevant and that I was interested in straightaway without too much information.
- It was clear to see straightaway if I wanted to see more
- I was amazed at how it brought up so much relevant information
- I didn't like the way the linked data was presented but I can really see the



potential of what is there in terms of all relevant information in one place to be viewed effortlessly compared to trying to sift through all the information presented by regular search engines.

- Excellent as it shortens the time needed on the web

In the penultimate question users were asked what they think are the possibilities that an accurate user model can provide to a user, and the following comments were made:

- It could make information easier to find for research
- It could make web browsing more user friendly
- Could promote more time spent on the computer
- Good way to keep track of what you are interested in
- Makes history of web browsing more useful
- Makes searching the internet a more personal experience.
- Shortens the time needed to look for information.
- Gets rid of all unnecessary excess of information.
- It could offer the user a comprehensive view of a topic without spending excess time on the subject matter.
- Possibilities are endless and creating such a user friendly efficient environment in which to browse will enhance the experience and enjoyment.

- It can make web browsing much more enjoyable and less frustrating and the information the user is interested in will be closer to being at the click of a button.
- It will shorten the time considerably for the user and bring up relevant information.

It is clear to see from these comments what users deem most important about web browsing and the possibilities that an accurate user model can provide, most notably in terms of time saving and having access to information which is relevant to the user.

The final question in the study asks users for any additional comments about AMS and the following is a selection of the responses:

- Overall I found it a very interesting experience; it was amazing how it was working away gathering information without having to be asked any questions.
- Impressed with the accuracy of it.
- It worked in showing me information that I am interested in.
- An interesting experiment.
- Great potential
- I would be interested in seeing further work on this subject

- This idea of personalisation to provide the information one is looking for in a time efficient manner is obviously the way forward and will make the users experience much more pleasurable.
- AMS comes back with useful information and tries to get around one of the major problem that I have with search engines and that is the complete overload of information.

## 5.4 Main Findings

From analysing the information collected in the databases of all six users we see that the number of key terms collected range from 79 to 116 over the span of the thirty minute browsing period, See Figure 5.15. The top keywords gained ratings which ranged from 54 to 76 with a similar range for the top rated websites.

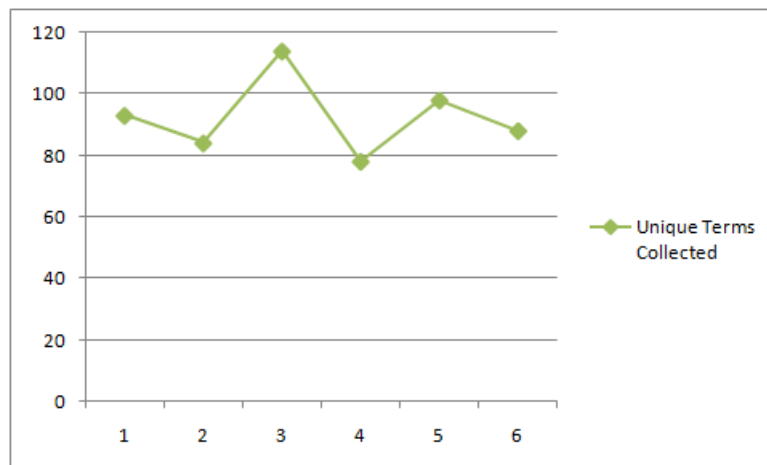


Figure 5.15: Number of unique terms collected for 6 users over a 30 minute period

The evaluation of AMS was successful insofar as all users agreed that the

websites and keywords shown by AMS were indicative of their interests while browsing the web. Bear in mind however that the evaluation was only carried out with six people and there would need to be much more extensive evaluations carried out to be able to say with complete certainty that AMS will be accurate all of the time. However AMS did model the user's interests effectively within the bounds of our evaluation. Users were in favour of the idea of a user model working incognito, building a profile automatically as opposed to having to answer questions which all users found annoying. Most users felt that an accurate user model could ultimately save time and return more relevant information.

The exercise of incorporating the linked data into the end of the evaluation was very useful for a couple of reasons

1. It showed how the system works more effectively within an individual domain. The users who were using the open domain did not find as much interesting information from the linked data as the users working within a chosen domain.
2. It showed users the potential of the system and how it is possible to be shown purely relevant information without all the excess unrelated material.

# Chapter 6

## Conclusions

### 6.1 Overview

This chapter re-introduces the research statement from Chapter 1 and summarises how this work has achieved the objectives and goals of the project. Specifically the results achieved from the evaluations of the system and their potential are discussed. There is an outline of what contribution is made to the State of the Art in user modelling systems and finally future work in this area is discussed.

### 6.2 Motivation and Objectives

The research question motivating this dissertation asked to what degree a user's interests can be inferred by gathering text, performing text analysis and applying a rating, on information from the pages that he/she visits while both actively searching for information and casually browsing for information. To address the question, one of the goals of the project was to build a non invasive user modelling

system that could surmise the user's browsing interests. We can say that both of these objectives were accomplished with the AMS system from the unanimous vote by evaluators as to the non-invasiveness of the system and the ability of the system to accurately infer their interests. Within the boundary of our limited number of evaluators AMS modelled the users interests effectively and was more effective while working within specific domains.

The other objective of the dissertation was to identify key areas of interest for the user as they browsed the WWW. Both the User Centred Design process and State of the Art analysis played a significant role in determining what information would be most effective to use and this proved to be the case when the system was evaluated.

### **6.3 Contribution to State of the Art**

This research has focused on collecting a variety of information from web pages that users visit, processing the information and incorporating a ratings system on the data to generate a user model of browsing behaviour. Since all users agreed that the information presented to them by AMS was relevant to them and was indicative of not only their interest but their level of interest, we can say now that collecting key information from web pages such as the text from the page title, text from headings, keywords from the meta tags, text in bold and text highlighted as the user browses, can be a useful means of summarising the content of the page and inferring what the page is about. Applying a rating to this information based on how long the user has spent viewing the page with respect to the length of

the page, is where the system becomes a user modelling system, profiling the individual user's interest. It's a simple concept but an effective one; find out what the page is about and rate it according to how long the user has spent viewing the information.

Another significant accomplishment of this work is the ability to collect and store all data from within the browser without hampering performance in any way, providing users with the possibility of a non-intrusive user modelling system without any privacy concerns for the collected data as it is collected and housed within the browser of the user's computer.

## **6.4 Future Work**

The success of the part of the evaluation which presents information from the linked data environment to the user, indicates that future work needs to incorporate the linked data results directly from the extension. Our evaluations showed that the system was much more successful when working within a single domain than it was in the open domain. The reason for this is because all data collected from a single domain can be said to have some type of relationship with each other. Since the idea behind linked data is to define relationships and separate information into domains, it is easy to see why our evaluations worked better from a single domain.

As we seen in Figure 5.16, AMS can gather a substantial amount of data in a small period of time. The challenge now is to find some way to cluster the data generated by the system and categorise it into distinct areas of interest, to get

the best possible results from the linked data environment. Ironically the human visual system is remarkably good at clustering information, and recognising trends and patterns [47]. Therefore we need to teach a computer to do what the human brain can do easily. If we can find a way to cluster data from the open domain into areas of interest to the user, our system will become very powerful.



# Bibliography

- [1] B. Kules. User modeling for adaptive and adaptable software systems. In *ACM Conference on Universal Usability*, 2000.
- [2] A.L. Montgomery and C. Faloutsos. Identifying web browsing trends and patterns. *Computer*, 34(7):94–95, 2001.
- [3] A.W. Lazonder, H.J.A. Biemans, and I.G.J.H. Wopereis. Differences between novice and experienced users in searching information on the World Wide Web. *Journal of the American Society for Information Science*, 51(6):576–581, 2000.
- [4] J. Kramer, S. Noronha, and J. Vergo. A user-centered design approach to personalization. *Communications of the ACM*, 43(8):44–48, 2000.
- [5] J.R. Anderson. Cognitive psychology and intelligent tutoring. In *Proceedings of the Cognitive Science Society Conference*, pages 37–43, 1984.
- [6] S. Madhavaram and R. Appan. The potential implications of web-based marketing communications for consumers’ implicit and explicit brand attitudes: A call for research. *Psychology and Marketing*, 27(2):186–202, 2010.

- [7] L.D. Catledge and J.E. Pitkow. Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN systems*, 27(6):1065–1073, 1995.
- [8] L. Tauscher and S. Greenberg. How people revisit web pages: empirical findings and implications for the design of history systems. *International Journal of Human Computer Studies*, 47:97–138, 1997.
- [9] A. Cockburn and B. McKenzie. What do Web users do? An empirical analysis of Web use. *International Journal of Human-Computer Studies*, 54(6):903–922, 2001.
- [10] B.A. Huberman, P.L.T. Pirolli, J.E. Pitkow, and R.M. Lukose. Strong regularities in world wide web surfing. *Science*, 280(5360):95, 1998.
- [11] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *sbv*, 14(w3c):9, 2009.
- [12] T.D. Wilson. Information behaviour: An interdisciplinary perspective\* 1. *Information Processing & Management*, 33(4):551–572, 1997.
- [13] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, page 40. ACM, 2001.
- [14] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265. Citeseer, 1998.

- [15] R. Catrambone, J. Stasko, and J. Xiao. Anthropomorphic agents as a user interface paradigm: Experimental findings and a framework for research. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, pages 166–171. Citeseer, 2002.
- [16] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, 41(4):288–297, 1990.
- [17] J. Fink, A. Kobsa, and A. Nill. Adaptable and adaptive information access for all users, including the disabled and the elderly. In *International Conference UM97. Wien New York: Springer*, volume 171, page 173. Citeseer, 1997.
- [18] J. Fink, A. Kobsa, and A. Nill. User-oriented Adaptivity and Adaptability in the AVANTI Project. In *Designing for the Web: empirical studies*. Citeseer, 1996.
- [19] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. In *ACM SIGIR Forum*, volume 37, pages 18–28. ACM, 2003.
- [20] G. Brajnik and C. Tasso. A shell for developing non-monotonic user modeling systems. *International Journal of Human-Computer Studies*, 40(1):31–62, 1994.
- [21] T.R. Gruber et al. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5):907–928, 1995.
- [22] J. Kay. Ontologies for reusable and scrutable student models, position paper.

- Proceedings of AIED99 Workshop on Ontologies for Intelligent Educational Systems*, 1999.
- [23] J. Kay, B. Kummerfeld, and P. Lauder. Personis: a server for user models. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 203–212. Springer, 2006.
- [24] L. Razmerita. Ontology-Based User Modeling. *Ontologies*, pages 635–664, 2007.
- [25] A. Kobsa. Generic user modeling systems. In *The adaptive web*, pages 136–154. Springer-Verlag, 2007.
- [26] H. Lieberman et al. Letizia: An agent that assists web browsing. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 924–929. LAWRENCE ERLBAUM ASSOCIATES LTD, 1995.
- [27] M.J. Pazzani, J. Muramatsu, D. Billsus, et al. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the national conference on artificial intelligence*, pages 54–61, 1996.
- [28] E. André, W. Graf, J. Müller, H.J. Profitlich, T. Rist, and W. Wahlster. AiA: Adaptive communication assistant for effective infobahn access. *Document, DFKI, saarbücken*, 1996.
- [29] T. Berners-Lee. Semantic web road map. *W3C Design Issues*, 1998.
- [30] T. Berners-Lee. Linked data, 2006. *W3C Design Issues*, 2006.
- [31] T. Berners-Lee. Design issues: Linked data. *Online, Retrieved May, 25, 2009*.

- [32] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.
- [33] A.H. Tan. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, pages 65–70. Citeseer, 1999.
- [34] T. Nasukawa and T. Nagano. Text analysis and knowledge mining system. *IBM Systems Journal*, 40(4):967–984, 2001.
- [35] M. Krauthammer and G. Hripcsak. A knowledge model for the interpretation and visualization of NLP-parsed discharged summaries. In *Proceedings of the AMIA Symposium*, page 339. American Medical Informatics Association, 2001.
- [36] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(Suppl 1):S74, 2001.
- [37] M. Hearst. What is text mining. *Retrieved October*, 18:2005, 2003.
- [38] M. Porter. The porter stemming algorithm, 2001.
- [39] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.
- [40] C.C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, page 909. VLDB Endowment, 2005.

- [41] R. Agrawal and R. Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.
- [42] C.C. Aggarwal and P.S. Yu. A condensation approach to privacy preserving data mining. *Advances in Database Technology-EDBT 2004*, pages 183–199, 2004.
- [43] X. Shen, B. Tan, and C.X. Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, page 831. ACM, 2005.
- [44] D. Schuler and A. Namioka. *Participatory design: Principles and practices*. CRC, 1993.
- [45] D. Petrelli, A. De Angeli, and G. Convertino. A User-Centered Approach to User Modelling. *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES*, pages 255–264, 1999.
- [46] O. Conlan. *The multi-model, metadata driven approach to personalised eLearning services*. Trinity College, 2005.
- [47] B. Wünsche. A survey, classification and analysis of perceptual concepts and their application for the effective visualisation of complex information. In *Proceedings of the 2004 Australasian symposium on Information Visualisation-Volume 35*, page 24. Australian Computer Society, Inc., 2004.

# Appendix A

## Java Service

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class TextService {
    private static TextService _instance = null;
    public TextService() {
    }
    public static TextService getInstance() {
        if (_instance == null) {
            _instance = new TextService();
        }
        return _instance;
    }
    //create instance of the porter stemmer class
    PorterStemmer ps = new PorterStemmer();
    public ArrayList<Text> analyseText(String words) {
        words = wordFilter(words.toLowerCase());
        Map map = new HashMap();
        ArrayList<Text> wordResults = new ArrayList();
        StringTokenizer st=new StringTokenizer(words);
        for (int i=0, n=st.countTokens(); i<n; i++) {
            while(st.hasMoreTokens()){
                String key = st.nextToken();
                String key = ps.stem(key2);
                //count frequency of word appearance
                Integer frequency = (Integer)map.get(key);
                if (frequency == null) {
                    frequency = 1;
                } else {
                    int value = frequency.intValue();
                    frequency = new Integer(value + 1);
                }
            }
        }
    }
}
```

```

    }
    map.put(key, frequency);
    }
    }
    for(Iterator i = sortByValue(map).iterator(); i.hasNext(); ) {
    String key = (String) i.next();
    System.out.printf("key: "+key+ " value: "+map.get(key)+"\n");
    wordResults.add(new Text(key, (Integer)map.get(key)));
    }
    return wordResults;
    }
    //method to sort by value descending
    public List sortByValue(final Map m) {
    List keys = new ArrayList();
    keys.addAll(m.keySet());
    Collections.sort(keys, new Comparator() {
    public int compare(Object o1, Object o2) {
    Object v1 = m.get(o1);
    Object v2 = m.get(o2);
    if (v1 == null) {
    return (v2 == null) ? 0 : 1;
    }
    else if (v1 instanceof Comparable) {
    return ((Comparable) v2).compareTo(v1);
    }
    else {
    return 0;
    }
    }
    });
    return keys;
    }
    //method to extract all stopwords from text
    public String wordFilter(String text) {
    String regexx = "";
    String regexx2 = "";
    try{
    //stopwords are located in external file
    FileInputStream fstream = new FileInputStream("stopwords.txt");
    // Get the object of DataInputStream
    DataInputStream in = new DataInputStream(fstream);
    BufferedReader br = new BufferedReader(new InputStreamReader(in));
    String strLine;
    //Read File Line By Line
    while ((strLine = br.readLine()) != null){
    //pattern matching
    regexx2 = text.replaceAll("\\W\\s", "");
    regexx2 = regexx2.replaceAll("\\d", "");
    regexx = regexx.concat(strLine+"|");
    }
    Pattern replace = Pattern.compile("\\b(" + regexx + ")\\b");
    Matcher matcher = replace.matcher(regexx2);
    text = (matcher.replaceAll(""));
    }catch(IOException e)
    {
    System.out.println("Exception : "+ e.getMessage());
    }
    return text;
    }
    }

```



# Appendix B

## Evaluation Questionnaires

## AMS Pre Evaluation Questionnaire

---

Please answer the following questions to the best of your ability.

Gender:

Age:

Which of the following best describes the line of work you are in?

- Student
- Teacher/Lecturer/Researcher
- Business
- Homemaker
- IT
- Other, please state

1. What is the TOTAL number of hours a week that you spend on a computer?

- 0
- Less than 1
- 1-3
- 4-6
- 7-10
- 11-15
- 16-20
- 20+

2. Of these hours, how many are used for activities that you do on a regular basis, e.g social networking, reading the newspaper?

- 0
- Less than 1
- 1-3
- 4-6
- 7-10
- 11-15
- 16-20
- 20+

**3. How many are used for searching and/or browsing for information?**

- 0    Less than 1    1-3    4-6    7-10    11-15    16-20    20+

**4. How would you rate your level of computer expertise?**

- Novice    Average    Expert

**5. When browsing the internet I find pop-up's very annoying**

- Strongly agree    Agree    Neither agree or disagree    Disagree    Strongly disagree

**6. I often fill out questionnaires that are trying to build a profile of me for marketing purposes**

- Strongly agree    Agree    Neither agree or disagree    Disagree    Strongly disagree

**6. I can always find what I am searching for on the internet quickly and easily**

- Strongly agree    Agree    Neither agree or disagree    Disagree    Strongly disagree

## AMS Post Evaluation Questionnaire

---

Please answer the following questions to the best of your ability.

**1. Are you happy with your current web browsing experience and if not why?**

**2. Did you find AMS easy to use?**

**3. Did you find AMS intrusive in any way as you browsed the web?**

**4. Did AMS have a negative impact on your browsing ability?**

**5. The website and their ratings suggested by AMS reflects the websites that I found most interesting over the period of the study**

Strongly agree  Agree  Neither agree or disagree  Disagree  Strongly disagree

**Comments:**

**6. The keywords and their ratings suggested by AMS is reflective of what I found most interesting during the period of the study**

Strongly agree  Agree  Neither agree or disagree  Disagree  Strongly disagree

**Comments:**

**7. I would like to see more information when I clicked on AMS**

Strongly agree  Agree  Neither agree or disagree  Disagree  Strongly disagree

**Comments:**

**8. The information shown to me from the linked data environment was useful and relevant to what I was browsing for**

Strongly agree  Agree  Neither agree or disagree  Disagree  Strongly disagree

**Comments:**

**9. What do you think are the possibilities that an accurate user model can provide to a user**

**10. Any further comments:**