Prioritising Hyperlinks for Topic-Focused Web Crawling using Lexical and Terminological Profiling

By

Mark A. Greenwood

A thesis submitted for the degree of

Master of Philosophy

in the

Faculty of Engineering and Physical Sciences

The University of Manchester

School of Computer Science

September 2009
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>2</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>DECLARATION</td>
<td>6</td>
</tr>
<tr>
<td>COPYRIGHT STATEMENT</td>
<td>7</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>8</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>9</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 1 – INTRODUCTION</td>
<td>16</td>
</tr>
<tr>
<td>1.1 Issues with the World Wide Web</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Research Questions, Aims and Objectives</td>
<td>21</td>
</tr>
<tr>
<td>1.3 Contributions and Results</td>
<td>22</td>
</tr>
<tr>
<td>1.4 Thesis Structure</td>
<td>23</td>
</tr>
<tr>
<td>CHAPTER 2 – BACKGROUND</td>
<td>25</td>
</tr>
<tr>
<td>2.1 Information Retrieval</td>
<td>25</td>
</tr>
<tr>
<td>2.2 Representations of the Web for IR</td>
<td>28</td>
</tr>
<tr>
<td>2.2.1 Document Content Representation</td>
<td>28</td>
</tr>
<tr>
<td>2.2.2 Document Link Representation</td>
<td>31</td>
</tr>
<tr>
<td>2.3 Manually Cataloguing the Web</td>
<td>33</td>
</tr>
<tr>
<td>2.4 Automatically Cataloguing the Web</td>
<td>36</td>
</tr>
<tr>
<td>2.4.1 Structure and Implementation</td>
<td>36</td>
</tr>
</tbody>
</table>
2.4.2 Problems Overcome by Web Crawlers ................................................................. 39
2.4.3 Problems Posed by Web Crawlers ................................................................. 40

2.5 Searching the Index .......................................................................................... 41

2.6 Ranking Retrieved Documents ........................................................................ 46
  2.6.1 Content-based Methods ............................................................................. 50
  2.6.2 Graph-based Methods .............................................................................. 50
  2.6.2.1 HITS ................................................................................................. 51
  2.6.2.2 PageRank ......................................................................................... 53
  2.6.2.3 Topic-Sensitive PageRank ................................................................. 54
  2.6.3 Relevance Feedback ................................................................................. 55

2.7 Topic-focused Web Crawlers ........................................................................... 56
  2.7.1 General Structure of a Topic-focused Web Crawler .................................... 58
  2.7.2 Achieving a Topic-focused Web Crawl .................................................... 58

2.8 Summary .......................................................................................................... 68

CHAPTER 3 – TOPIC PROFILING AND LINK PRIORITISATION ......................... 72

3.1 Document and Model Representation ............................................................. 73
  3.1.1 Document Features .................................................................................. 73
  3.1.2 Modelling the Information Need ............................................................. 74

3.2 Link Prioritisation ............................................................................................ 75
  3.2.1 Content ..................................................................................................... 77
  3.2.2 Context ..................................................................................................... 78
  3.2.3 Source Page ............................................................................................ 79
  3.2.4 Weighting Terms ..................................................................................... 80
  3.2.5 Combining the Scores ............................................................................. 81

3.3 Summary .......................................................................................................... 83

CHAPTER 4 – DESIGN AND IMPLEMENTATION ............................................ 85

4.1 Topic Model Engineering .................................................................................. 86

4.2 Document Retrieval and Parsing .................................................................... 87

4.3 Document and Link Classification .................................................................... 88
4.4 Abstract Data Types ................................................................. 91

4.5 Database Schema ................................................................. 91

4.6 Implementation Notes ........................................................... 92
  4.6.1 Development and Execution Environments ................................ 92
  4.6.2 External Packages ............................................................. 92
  4.6.3 Performance Discussion ...................................................... 93

4.7 Summary ............................................................................... 94

CHAPTER 5 – RESULTS AND DISCUSSION .................................... 95

5.1 Experimental Setup ............................................................... 95

5.2 Un-weighted Term Comparison Results ...................................... 100
  5.2.1 Page Classification Precision .............................................. 100
  5.2.2 Link Prioritisation: context features ..................................... 104
  5.2.3 Link Prioritisation: combining different features .................. 112
  5.2.4 Link Prioritisation Results in Larger Crawls ......................... 118
  5.2.5 Link Prioritisation with Linear Combination of Features ....... 122
  5.2.6 Summary ........................................................................ 127

5.3 Incorporation of Term Relevance ............................................. 128
  5.3.1 Page Classification Results .............................................. 128
  5.3.2 Link Prioritisation: Context Features .................................. 129
  5.3.3 Link Prioritisation: Combining Different Features .............. 133
  5.3.4 Weighted Link Prioritisation in Larger Crawls ..................... 138
  5.3.5 Summary ........................................................................ 142

5.5 Summary ............................................................................... 142

CHAPTER 6 – CONCLUSIONS AND FUTURE WORK ...................... 144

APPENDIX .............................................................................. 151

REFERENCES ........................................................................ 165

Word Count: 41,921
Abstract

The World Wide Web allows vast amounts of information to be available to users – but only if they know how to get it. While containing a wide range of information pertaining to all manner of human knowledge, the Web’s organisation and the fact that it is an ever growing, ever changing resource can make finding what one wants a difficult task. Topic Focused Web Crawlers attempt to create indexes of Web pages pertaining to some user-defined topic. Exploiting the Web’s hyperlinked structure, they index only the pages that are judged relevant to the topic. The ultimate aim is an index that can be used for search (typically as part of a larger Information Retrieval system) with a higher precision within that particular domain than a more general index.

One major problem that Topic-Focused Web Crawling faces is efficiency – visiting the relevant pages while visiting as few irrelevant pages as possible. This issue is addressed in this thesis. We use example documents arranged taxonomically from within existing Web directories to create term based, fine-grained topic models from relatively small document sets. Lexical similarity is then used to judge the closeness of link features within visited pages to the terms within the topic model to visit the links expected to yield pages closest to the topic. Maintaining the taxonomic structure of the model allows for the incorporation of ‘background knowledge’ into link prioritisation meaning that in the absence of ‘good’ links, the closest ones can be visited. This keeps the crawl at least close to the topic, visiting pages more likely to yield ‘good’ links in the future.

The lexical similarity based link classification is compared to a naive, breadth-first crawler to evaluate the efficiency of this prioritisation strategy. The lexical similarity based method is shown to be 2-6 times more efficient than this baseline. The results of the evaluation crawls are analysed in detail to highlight areas where improvements can be made in order to increase the precision with which the features are classified, and the overall efficiency may be increased. A term cloud representing the summary of concepts presented and used in this thesis is shown below.

![Figure i: Thesis Term Cloud](http://www.nactem.ac.uk/software/termine/)

---

1 Generated using TerMine (http://www.nactem.ac.uk/software/termine/) for term recognition, and IBM WordCloud (http://www.alphaworks.ibm.com/tech/wordcloud) for cloud generation
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institute of learning.

Mark Greenwood
Copyright Statement

The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “Copyright”) and he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

Copies of this thesis, either in full or in extracts may be made only in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

The ownership of any patents, designs, trade marks and any and all other intellectual property rights except for the Copyright (“the Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Further information on the conditions under which disclosure, publication and exploitation of this thesis, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of the School of Computer Science and the Dean of the Faculty of Engineering and Physical Sciences.
Acknowledgements

Firstly, I would like to offer my deepest thanks to Dr. Goran Nenadic – my supervisor for this project. Firstly for giving me the opportunity to undertake this research and also for his constant help and support throughout it. Thanks to his experience and advice, this piece of work is something I am very proud of.

I would also like to thank the other members of our research group, past and present who have given me invaluable support, through input on my ideas/work, general advice for the course and for the good times we’ve shared. Hammad Afzal, Farzaneh Sarafran, Hui Yang, James Eales, Martin Gerner and Ann Gledson.

I would also like to take this opportunity to thank my examiners, Dr. Udo Kruschwitz and Prof. John Keane for their valuable input in to this work.

My family too deserve more gratitude than I can offer here. My Mum and Dad, Elizabeth and Eric, thank you for your unending emotional and financial support and for your understanding as well as your shoves in the right direction when I needed them. Thank you for always helping me as much as possible to succeed in my goals. Thanks too to my Sister Beverley and Brother-in-Law Sam for their support and for some welcome holidays.

Thanks also to my friends for helping me blow off some steam, and listening to me talk about Web crawling. Jo, Chris, Vicky, Matt, Steve and the rest (you all know who you are) - Thanks guys, you’re all awesome.
List of Tables

Table 3.1: Examples of lexical profiling of terms ................................................... 74
Table 3.2: Examples of lexical similarity measures ................................................. 77
Table 5.1: Topics and Start Pages ............................................................................ 97
Table 5.2: Science/Biology/Bioinformatics/Companies Training Information .......... 98
Table 5.3: Society/Issues/Science and Technology/ Biotechnology/ Genetics/GeneticallyModified Food Training Information ......................................... 99
Table 5.4: Health/Conditions and Diseases/ Nutrition and Metabolism Disorders/Obesity Training Information .......................................................... 99
Table 5.5: Weights assigned to link features ............................................................. 122
**List of Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Total Sites across all Domains 08/95 – 07/09 (Adapted from <a href="http://www.netcraft.com">www.netcraft.com</a>)</td>
<td>18</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Functional Overview of Information Retrieval Systems (Salton &amp; McGill, 1983)</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Linking in the Web</td>
<td>31</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Hyperlinks as edges</td>
<td>31</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>The Web as a graph</td>
<td>32</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Example Topic Taxonomy (Extracted from the ODP - <a href="http://www.dmoz.org">http://www.dmoz.org</a>)</td>
<td>34</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Example Topic List (Extracted from the ODP - <a href="http://www.dmoz.org/Sports/Cycling">http://www.dmoz.org/Sports/Cycling</a>)</td>
<td>34</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Structure of a Web Crawler (Baeza-Yates, Ribeiro-Neto, 1999)</td>
<td>37</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>Web Crawler Processes</td>
<td>38</td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>Example Boolean query syntax tree</td>
<td>42</td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>Construction of a term-by-document matrix (Berry and Browne, 2005)</td>
<td>44</td>
</tr>
<tr>
<td>Figure 2.11</td>
<td>Densely linked set of hubs and authorities (Kleinberg, 1999)</td>
<td>52</td>
</tr>
<tr>
<td>Figure 2.12</td>
<td>A bad example for HITS (Borodin et al., 2001)</td>
<td>53</td>
</tr>
<tr>
<td>Figure 2.13</td>
<td>Breadth First Vs. Optimised Crawl Paths</td>
<td>57</td>
</tr>
<tr>
<td>Figure 2.14</td>
<td>The Topic-focused Web Crawler Processes</td>
<td>58</td>
</tr>
<tr>
<td>Figure 2.15</td>
<td>Machine learning processes</td>
<td>61</td>
</tr>
<tr>
<td>Figure 2.16</td>
<td>Decision boundary margin (Bishop, 2006)</td>
<td>62</td>
</tr>
<tr>
<td>Figure 2.17</td>
<td>Increasing the margin (Bishop, 2006)</td>
<td>62</td>
</tr>
<tr>
<td>Figure 2.18</td>
<td>Example Web Graph</td>
<td>64</td>
</tr>
<tr>
<td>Figure 2.19</td>
<td>Example links and context (<a href="http://www.bikelane.com">http://www.bikelane.com</a>)</td>
<td>66</td>
</tr>
<tr>
<td>Figure 2.20</td>
<td>Example of a link in a bad context (<a href="http://www.bikelane.com">http://www.bikelane.com</a>)</td>
<td>66</td>
</tr>
<tr>
<td>Figure 2.21</td>
<td>Problems with context analysis (<a href="http://www.bikelane.com">http://www.bikelane.com</a>)</td>
<td>67</td>
</tr>
<tr>
<td>Figure 2.22</td>
<td>Creation of the context graph (Diligenti et al., 2000)</td>
<td>68</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Summary of Overall Approach</td>
<td>82</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Automatic Weight Assignment</td>
<td>82</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>General Topic Focused Web Crawler Processes</td>
<td>85</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Diagram for Term-Level Feature Model Creation Process</td>
<td>87</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Document Download and Parsing Process Diagram</td>
<td>88</td>
</tr>
</tbody>
</table>
Figure 4.5: Link Score Assignment Process Diagram – Term-Based Method ........... 90
Figure 4.6: Database Schema ................................................................................. 91
Figure 5.1: Page Classification Precision – Un-weighted Term Comparison ........... 101
Figure 5.2: Manual Annotation and Inter-Annotator Agreement Results ............... 102
Figure 5.3: Harvest Rate Results – Context Length Variation (Bioinformatics Companies – nott.ac.uk) ................................................................................................................. 105
Figure 5.5: Harvest Rate Results – Context Length Variation (Bioinformatics Companies – http://www.solpugid.com/links) ................................................................................................................. 105
Figure 5.4: Harvest Rate Results – Context Length Variation (Bioinformatics Companies – http://www.bioinformatics.org) ................................................................................................................. 105
Figure 5.6: Harvest Rate Results – Context Length Variation (Bioinformatics Companies – http://www.bioinformatics.ca) ................................................................................................................. 105
Figure 5.7: Harvest Rate Results – Context Length Variation (Bioinformatics Companies – EBI) ................................................................................................................. 105
Figure 5.8: Early Link Placement in Seed Page (Bioinformatics Companies) .......... 107
Figure 5.9: Context Length - Overall Result Comparison (Bioinformatics Companies) nott.ac.uk .................................................................................................................. 107
Figure 5.10: Portion of Visited Pages (visits 23-29) Context Length 10 Crawl (Bioinformatics Companies) ................................................................................................................. 108
Figure 5.11: Example Links from Context Length 10 Crawl (Bioinformatics Companies) .................................................................................................................. 108
Figure 5.12: Link analysis using Context Length 6 (Bioinformatics Companies) ....... 108
Figure 5.13: Harvest Rate Results – Context Length Variation (GM Foods – http://www.actionbioscience.org) ................................................................................................................. 109
Figure 5.14: Harvest Rate Results – Context Length Variation (GM Foods – intute hot-links) ................................................................................................................. 109
Figure 5.15: Harvest Rate Results – Context Length Variation (GM Foods – http://www.genome.gov) ................................................................................................................. 109
Figure 5.16: Harvest Rate Results – Context Length Variation (GM Foods – New Scientist) ................................................................................................................. 109
Figure 5.17: Harvest Rate Results – Context Length Variation (GM Foods – ORNL) ... 110
Figure 5.18: WormBase.org shared header (http://www.wormbase.org) ................. 110
Figure 5.19: Harvest Rate Results – Context Length Variation (Obesity – http://www.diseasesconditions.com) ................................................................................................................. 111
Figure 5.20: Harvest Rate Results – Context Length Variation (Obesity – MedicineNet) .................................................................................................................. 111
Figure 5.21: Harvest Rate Results – Context Length Variation (Obesity – http://www.dh.gov) .................................................................................................................. 111
Figure 5.49: Larger crawl results (GM Foods – New Scientist) .................................... 120
Figure 5.50: Larger crawl results (GM Foods – ORNL) ........................................... 120
Figure 5.51: Larger Crawl Results (Obesity – diseasesandconditions.com) .............. 121
Figure 5.52: Larger Crawl Results (Obesity – MedicineNet) ...................................... 121
Figure 5.53: Larger Crawl Results (Obesity – dh.gov) ................................................ 121
Figure 5.54: Larger Crawl Results (Obesity – NHS.uk) ............................................. 121
Figure 5.55: Larger Crawl Results (Obesity – NIH.gov) .......................................... 122
Figure 5.56: Link Feature Weight Results (Bioinformatics Companies – nott.ac.uk) ..... 123
Figure 5.57: Link Feature Weight Results (Bioinformatics Companies – Bioinformatics.org) .............................................................................................................. 123
Figure 5.58: Link Feature Weight Results (Bioinformatics Companies – solpugid.com/Links) .............................................................................................................. 124
Figure 5.59: Link Feature Weight Results (Bioinformatics Companies – Bioinformatics.ca) .............................................................................................................. 124
Figure 5.60: Link Feature Weight Results (Bioinformatics Companies – EBI) ............. 124
Figure 5.61: Link Feature Weight Results (GM Foods – actionbioscience.com) ............ 125
Figure 5.62: Link Feature Weight Results (GM Foods – genome.gov) ....................... 125
Figure 5.63: Link Feature Weight Results (GM Foods – intutehotlinks) ..................... 125
Figure 5.64: Link Feature Weight Results (GM Foods – New Scientist) ....................... 125
Figure 5.65: Link Feature Weight Results (GM Foods – ORNL) ................................. 125
Figure 5.66: Link Feature Weight Results (Obesity – diseasesandconditions.com) ..... 126
Figure 5.67: Link Feature Weight Results (Obesity – dh.gov) ..................................... 126
Figure 5.68: Link Feature Weight Results (Obesity – MedicineNet) ............................ 126
Figure 5.69: Link Feature Weight Results (Obesity – NHS.uk) .................................. 126
Figure 5.70: Link Feature Weight Results (Obesity – NIH.gov) ................................... 127
Figure 5.71: C-Value Weighted Classification Precision Results ................................. 128
Figure 5.72: Harvest Rate – C-Value Context Length Experiment (Bioinformatics Companies – nott.ac.uk) .......................................................... 129
Figure 5.73: Harvest Rate – C-Value Context Length Experiment (Bioinformatics Companies – Bioinformatics.org) .......................................................... 129
Figure 5.74: Harvest Rate – C-Value Context Length Experiment (Bioinformatics Companies – solpugid.com/Links) ......................................................... 130
Figure 5.75: Harvest Rate – C-Value Context Length Experiment (Bioinformatics Companies – Bioinformatics.ca) .............................................................. 130
Figure 5.76: Harvest Rate – C-Value Context Length Experiment (Bioinformatics Companies – EBI) .......................................................... 130
Figure 5.77: Harvest Rate – C-Value Context Length Experiment (GM Foods – actionbioscience.com) .......................................................... 131
Figure 5.78: Harvest Rate – C-Value Context Length Experiment (GM Foods – intute hotlinks) .............................................................................. 131
Figure 5.79: Harvest Rate – C-Value Context Length Experiment (GM Foods – genome.gov) .............................................................................. 131
Figure 5.80: Harvest Rate – C-Value Context Length Experiment (GM Foods – New Scientist).............................................................................. 131
Figure 5.81: Harvest Rate – C-Value Context Length Experiment (GM Foods – ORNL) .......................................................... 131
Figure 5.82: Harvest Rate – C-Value Context Length Experiment (Obesity – diseasesconditions.com) .......................................................... 132
Figure 5.83: Harvest Rate – C-Value Context Length Experiment (Obesity – MedicineNet) .............................................................................. 132
Figure 5.84: Harvest Rate – C-Value Context Length Experiment (Obesity – dh.gov) ..................................................................................... 132
Figure 5.85: Harvest Rate – C-Value Context Length Experiment (Obesity – NHS.uk) ..................................................................................... 132
Figure 5.86: Harvest Rate – C-Value Context Length Experiment (Obesity – NIH) ..................................................................................... 133
Figure 5.87: C-Value Link Score Variation Results (Bioinformatics Companies – nott.ac.uk) .............................................................................. 134
Figure 5.88: C-Value Link Score Variation Results (Bioinformatics Companies – solpugid.com/Links) .......................................................... 134
Figure 5.89: C-Value Link Score Variation Results (Bioinformatics Companies – Bioinformatics.org) .......................................................... 134
Figure 5.90: C-Value Link Score Variation Results (Bioinformatics Companies – Bioinformatics.ca) .......................................................... 134
Figure 5.91: C-Value Link Score Variation Results (Bioinformatics Companies – EBI) .......................................................... 134
Figure 5.92: C-Value Link Score Variation Results (GM Foods – actionbioscience.com) .......................................................... 135
Figure 5.93: C-Value Link Score Variation Results (GM Foods – intute hotlinks) ..................................................................................... 135
Figure 5.94: C-Value Link Score Variation Results (GM Foods – Genome.gov) ..................................................................................... 135
Figure 5.95: C-Value Link Score Variation Results (GM Foods – New Scientist) ..................................................................................... 135
Figure 5.96: C-Value Link Score Variation Results (GM Foods – ORNL) ..................................................................................... 136
Figure 5.97: C-Value Link Score Variation Results (Obesity – diseasesconditions.com) ..................................................................................... 137
Figure 5.98: C-Value Link Score Variation Results (Obesity – MedicineNet) ..................................................................................... 137
Figure 5.99: C-Value Link Score Variation Results (Obesity – dh.gov) ..................................................................................... 137
Figure 5.100: C-Value Link Score Variation Results (Obesity – NHS.uk) ..................................................................................... 137
Figure 5.101: C-Value Link Score Variation Results (Obesity – NIH.gov) ..................................................................................... 137
Figure 5.102: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – nott.ac.uk) .................................................................................................................. 138

Figure 5.103: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – bioinformatics.org) .................................................................................................................. 138

Figure 5.104: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – Solpugid.com/Links.htm) .................................................................................................................. 138

Figure 5.105: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – bioinformatics.ca) .................................................................................................................. 138

Figure 5.106: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – EBI) .................................................................................................................. 139

Figure 5.107: Harvest Rate – C-Value Large Crawl (GM Foods – actionbioscience.com) .................................................................................................................. 140

Figure 5.108: Harvest Rate – C-Value Large Crawl (GM Foods – genome.gov) ........ 140

Figure 5.109: Harvest Rate – C-Value Large Crawl (GM Foods – intute hotlinks) ...... 140

Figure 5.110: Harvest Rate – C-Value Large Crawl (GM Foods – New Scientist) ...... 140

Figure 5.111: Harvest Rate – C-Value Large Crawl (GM Foods – ORNL) ............... 140

Figure 5.112: Harvest Rate – C-Value Large Crawl (Obesity – diseasesconditions.com) .................................................................................................................. 141

Figure 5.113: Harvest Rate – C-Value Large Crawl (Obesity - MedicineNet) ........... 141

Figure 5.114: Harvest Rate – C-Value Large Crawl (Obesity – dh.gov) ................. 141

Figure 5.115: Harvest Rate – C-Value Large Crawl (Obesity – NHS.uk) ............... 141

Figure 5.116: Harvest Rate – C-Value Large Crawl (Obesity – NIH.gov) ............... 141
Chapter 1 – Introduction

The Internet is a vast global network allowing people all over the world to communicate and share information. On the back of this underlying structure, many protocols and methods of allowing this communication have arisen and been implemented on a large scale. One of the most common of these methods is the World Wide Web (WWW or Web). This is a system of interlinked hypertext documents stored on servers all over the world, accessible through a number of protocols built on top of the internet architecture. Users can access hypertext documents (as well as documents of other formats) using Uniform Resource Locators (URLs) which are unique addresses leading to the resources. These URLs convey not only where the resources are, but how to interact with them (Gourley and Totty, 2002). Together, these pages contain information on almost all aspects of human knowledge, making harnessing this information an important challenge for computer science and other disciplines.

The Web is already an extremely powerful resource. It contains a vast amount of related and unrelated information (De Bra and Post, 1994) and has a multitude of methods for people to share experiences, information and opinions much easier, faster and cheaper than previously possible (Baeza-Yates and Ribeiro-Neto, 1999). Unfortunately, the organisation of the Web makes the utilization of information not an easy task (see section 1.1). Finding information on the WWW using only intrinsic navigational methods is nearly impossible (De Bra and Post, 1994; Pinkerton, 1994), meaning that a lot of work has been done finding ways of building on top of the underlying architecture to aid information-seeking.

Web Crawling (Pinkerton, 1994) has been used to automatically visit websites and create indexes to enable searching for information (generally) using keywords or phrases. Several search engines (i.e. Google\(^2\), Yahoo\(^3\) and Ask\(^4\)) have been implemented on top of Web Crawlers. The aim of Web Crawlers is generally to catalogue all the pages available on the Web (McBryan, 1994). Topic-Focused Web Crawling (De Bra and Post, 1994; Chakrabarti et al., 1999a, 1999b), however, aims to facilitate the discovery and cataloguing of topic-specific resources within the WWW. By traversing the hyperlinked structure of

\(^2\) http://www.google.com
\(^3\) http://www.yahoo.com
\(^4\) http://www.ask.com
the Web and cataloguing the pages that pertain to some user-specified topic, a topic-specific index can be created (rather than the more general one achieved by Web Crawlers) for use in searching within a certain general topic domain or as part of some larger information retrieval system.

This chapter aims to set out the motivation behind, and aims of this thesis, starting with a discussion of the problems posed by the organisation of the World Wide Web and the need for suitable information retrieval technologies.

### 1.1 Issues with the World Wide Web

The Web is an immense collection of inter-linked resources stored on multiple servers all over the world (Pinkerton, 1994). It is an important and powerful resource which, according to recent surveys, is estimated to be used by roughly 21% of the World’s population.

As well as large collections of informative documents, it also contains mediums for communication and the exchange of ideas/opinions. For example, there are news websites, with information on world events as they happen, and services, which allow people to access information from large databases. Without this sort of resource, the exchange of information would be time consuming and costly (Baeza-Yates and Ribeiro-Neto 1999). Therefore improving the technology it uses to date to make its use faster and its resources more accessible is an extremely worthwhile pursuit.

As revolutionary and powerful as the WWW is, there are features inherent in it that can make its usage challenging:

- **Size:** According to a recent Web server survey carried out by Netcraft, there are almost 240 million web domains responding on the Internet (fig. 1.1). Figure 1.1 shows the statistics from the Netcraft surveys carried out since 1995. Netcraft tries to contact different web addresses to count how many Web domains are currently registered. This is the only way to perform such a survey because there is no global index, so there is no direct way of finding the total number of Web domains directly.

---

6 [http://www.netcraft.com](http://www.netcraft.com) visited August 2009
Given that this number is merely the number of locations and not the number of documents across the locations (estimated to be over 11.5 Billion (Gulli and Signorini, 2005)) it is easy to see that finding information on any given subject is going to be hard even without the other difficulties faced when using the Internet.

- **No Inherent Index Structure**: Resources on the WWW are navigated to via unique addresses (URLs). These addresses are made up of a domain name, and a directory location which let internet browsers or other navigation software locate resources on the WWW. The domain name points to the host, and the directory location points to where the document is stored on the host.

  URLs and hyperlinks found in hypertext documents are the only inherent navigation techniques in the WWW due to the design of hypertext media (Pinkerton, 1994; De Bra and Post, 1994). This means that if a user wants to find a page on some certain topic, the only inherent way to find it is to know the address. There is no global index of the Web, which exacerbates the problems raised by the size mentioned earlier. With nearly 240 million separate collections of documents and no index to explain which sites are about what, finding information through simple navigation would be basically impossible (Pinkerton, 1994).
• **Dynamic Structure and Content:** The Web is constantly changing in both 
structure and content. Firstly, it is growing at an exponential rate (see Fig. 1.1). 
The Web has grown at an increasing rate over the past decade and seems to be 
continuing the trend (for example growing by around 7 million sites in one month 
(August 07 – November 07)). Ntoulas *et al.* (2004) estimate that new pages are 
created on the Web at a rate of 8% each week.

Of course, as well as new content being created, the old pages also 
disappear from the Web. Ntoulas *et al.* (2004) estimated that only 20% of the 
pages available on the Web today will be available after one year. This can be for 
many reasons, e.g. the author may just have given up writing, or a business may 
have ceased trading. For instance, in June 2007 Yahoo closed down its auction 
services due to market dominance from eBay\(^7\) in that field. This means that portion 
of the Web will now be completely different.

Probably the most dynamic part of the Web is the content itself. Pages are 
constantly changing, as new events happen, new information comes to light or even 
the domains change owner, so keeping track of what is at the end of an address can 
be taxing. Pages in the Ntoulas *et al.* (2004) test set showed low degrees of change 
(5% difference or less for over 50% of the changed documents after a year) or none 
at all – but also that there was no correlation between previous rates or frequency 
of page edits and future ones, making changes difficult to track or predict.

All these features mean that any information collected on the WWW and 
the documents contained within it needs to be constantly updated to provide a good 
enough description. This proves difficult if not impossible given the size of the 
WWW.

• **Quality of Information:** Documents on the WWW are of varying quality. There 
are many authoritative websites which present information from good and accurate 
sources. However, given that the Web is a public domain, and can be contributed 
to by anyone, some documents may not be as accurate or detailed as others. This 
could mean that the content is not clear, is not complete or is misinformation.

\(^7\) http://www.ebay.com
Information may also be out of date. If the author has stopped updating the website, and has not taken it off the Web server, then the information contained on it may be out of date, yet still be available to the public.

Where the user may find information on one site, there may be other sites which would offer better information. There is no global “scoring” mechanism on the Web, no established review system to highlight or remove bad quality information and promote the good quality information. Finding methods to identify ‘good’ sources of information is one active area of research into internet technologies (Kleinberg 1999; Brin and Page 1998; Haveliwala, 2002; Chitrapura et al., 2004; Liu et al., 2004; Richardson et al., 2006).

- **Distributed Content Storage:** The WWW is made up of documents stored on thousands of different servers all over the world. This means that finding information often means contacting many different servers. This also means that creating any kind of index or categorising these documents is not as simple as cataloguing information in a single geographical location as with historical Information Retrieval systems (i.e. library catalogues – see Section 2.1). A further limitation caused is that any changes which could be made in the structure of the Web become very difficult to implement as it would mean changing all of these servers.

- **Varying Languages and Formats:** As the WWW is an international resource, contributed to and used by people all over the world, it is natural that it is represented in many different languages. A recent survey for example, showed that the top ten languages used on the internet made up about 85% of its total users with content in English making up 30.4% alone. This means that information may be globally accessible, but is not globally understandable as the information a user might need may not be represented in their language. The Web also contains information in various formats, which offers a challenge to any Information Retrieval system trying to interpret or categories its items. While both these features are challenging to information retrieval on the Web, they will not be dealt with within this thesis.

---

8 www.internetworldstats.com, visited June 2008
To summarise, the main advantage the WWW has is the sheer quantity and availability of information and how quickly and cheaply it can be accessed (Baeza-Yates and Ribeiro-Neto, 1999). However, inherent in its organisation are problems that stand in the way of easy access to this information and may often mean that important information or good resources often go unnoticed and unrecognised. This thesis focuses on systems built on top of the underlying Web architecture that aim to give users and other systems access to the information held within it. This offers a large challenge to the Information Retrieval community – one that has given rise to a wide range of solutions attempting to harness the full capability of this very important resource.

1.2 Research Questions, Aims and Objectives

In order to make information more accessible, three main methods of cataloguing the contents of pages in the Web have been developed in order to facilitate searching for pages relevant to the user’s information need – manual cataloguing (e.g. Yahoo!, Open Directory Project), Web Crawling (McBryan, 1994; Pinkerton, 1994; Brin and Page 1998) and Topic-Focused Web Crawling (De Bra and Post, 1994; Chakrabarti et al., 1999b). These methods, their individual motivations and challenges will be discussed in more detail throughout Chapter 2. Generally speaking, their main concern is finding some way of representing (and generating the representation of) the content of the pages on the Web so that it can be navigated or searched through for the user’s desired information.

Topic Focused Web Crawling is concerned not with generating a representation of the entire Web, but just the relatively small portion pertaining to some particular user specified topic. This means that subsequent users would be able to search through an index containing just pages related to the general topic they are concerned with – giving more precise results, as problems with terms shared with other disciplines or used in various ways should have been addressed by the crawler.

The aim of this thesis concerns Topic-Focused Web Crawling and the creation of topic-specific document collections from the Web. More specifically, we aim to use existing, manually constructed taxonomical document collections of a generally limited size to model not only the target topic, but “surrounding” ones using terms extracted from example documents. It is hypothesised that this will allow the use and modelling of fine-
grained topics, whereas previously in Topic-Focused Web Crawling, more general topics have been the aim (see Section 2.6). It is also hypothesised that the use of the surrounding topics will facilitate a staged link prioritisation, allowing the crawler to identify “good” pages before downloading them. These pages will be relevant to the topic, or closely related to it based on features in previously downloaded documents which link to them.

The main research question we aim to answer is whether we can improve Topic-Focused Web Crawling through the prioritisation of link visitations according to the linguistic and terminological features of the links themselves and the documents in which they are found based on models of fine-grained, taxonomically described topics, incorporating the background knowledge presented.

Therefore, the objectives of this work can be summarised as follows:

- To design and implement methods of building models of fine-grained target topics and their closely related areas from example documents found in existing collections.
- To design and implement methods for prioritising crawler visitations according to the way they are referenced in other pages using the model developed.
- To test and evaluate the gain to efficiency (ratio of relevant page downloads to total page downloads) through using link prioritisation based on this model by comparing it to a naïve breadth-first crawler.

### 1.3 Contributions and Results

The work presented in this thesis resulted in the design and implementation of methods to

- Model target and closely related topics using existing topic taxonomies. The model is based on terminological features that lexically profile the topics.
- Prioritise the links according to the terminological features of their representations and the documents in which they are found. A weighting formula has been designed to reflect three aspects of every link: its content (the words/terms representing the link in a document), its context (the words/terms near the link
within the document) and the relevance of the document the link was found in (source page).

- Use ‘background knowledge’ from within the taxonomy to allow for staged prioritisation according to how closely related the features are judged to be to the target topic.

These methods were evaluated using a classification framework built on the same taxonomy based model, by comparing the efficiency of the prioritised crawl against a naive, un-prioritised crawl. The method was tested on three separate domains, covering very different kinds of information needs, showing performance gains of varying magnitudes.

Portions of the work completed in this thesis have been published in this publication:


1.4 Thesis Structure

This thesis has six chapters and is organised as follows

**Background:** This chapter covers a survey of the various indexing platforms which have been used/are being used on the WWW culminating with a detailed look at Topic-Focused Web Crawling, the main focus of this thesis. This includes an analysis of the main problems affecting this area and previous solutions implemented.

**Link Weights and Prioritisation:** Building on the problems set out in Chapter 2, Chapter 3 deals with the theoretical methods used in the work undertaken. It deals with the analysis of the pages found during the crawl and the intended prioritisation of visitations towards pages likely to yield ‘on-topic’ results.

**Design and Implementation:** Chapter 4 looks at the design of a Topic-Focused Web Crawler built around the methods proposed in the third chapter. Issues surrounding these
particular methods and Web Crawling at large are discussed and solutions proposed. It also discusses general implementation discussions such as the development and execution environments

**Results & Discussion:** This chapter details the experiments undertaken using the prioritisation methods outlined in Chapters 3 and 4 along with the results and their discussion. Performing a number of test crawls using various topics allows certain conclusions to be drawn about the method proposed here and Topic-Focused Web Crawling as a whole.

**Conclusion:** The concluding chapter draws on the information found during the experiments and discusses what can be learned from it, in terms of this crawler, Topic-Focused Web Crawling as a whole and the general Information Retrieval problems posed by the WWW. These conclusions are used as the basis of setting out future work which needs to be carried out in these areas.
Chapter 2 – Background

Information Retrieval (IR) is concerned with the representation, storage and access to information within a document collection (Baeza-yates and Ribeiro-Neto, 1999). The information dealt with is usually in the form of natural language document collections (Salton & McGill, 1983), and typically the World Wide Web. This collection offers its own unique challenges to the IR research community which were introduced in Chapter 1. This chapter will discuss these in more detail along with some of the methods which attempt to overcome them.

2.1 Information Retrieval

IR deals with the management of collections of information – involving the representation, storage and access to individual information items. While there are no restrictions on the kinds of information they involve, Salton and McGill (1983) note that they usually deal with various kinds of natural language documents (books, letters, newspaper articles, journal articles etc.). As noted above, the work in this thesis involves documents found within the World Wide Web which offers its own challenges to the discipline. This section aims to set out the principles behind IR and describe the challenges faced when applying them to the World Wide Web.

IR systems generally provide three aspects to facilitate the information seeking task (Salton & McGill, 1983). Firstly, for each “incoming information item” (i.e. some document to be catalogued) an appropriate representation or classification of its content is chosen through some established procedure. This process covers the creation of some index (whether it be manually created like a library catalogue system, or automatically created as with modern search engines) which will later be used by users seeking information. Secondly, the system must provide procedures for formulating requests for information required by the user. This forms the basis of finding items which satisfy the user’s needs. Thirdly, the system must have some methods for matching the user’s request to the documents in the collection. Figure 2.1 shows an overview of an information retrieval system.
Figure 2.1: Functional Overview of Information Retrieval Systems
(Salton & McGill, 1983)

Here, the three processes (indexing, search formulation and determining the similarity) and the indexing language are the components defined by the system. The documents and the requests from the user are the inputs to the system and a set of suitable documents would be the output.

The indexing language is concerned with the representation of the documents’ content. This can range from the categories assigned to books in a classic library card system (where a card with a manually created description of the book and where it can be found is grouped with other cards describing books belonging to the same category) to keywords extracted from the documents automatically (as with typical search engines).

The ‘set of requests’ are the specific requirements of the user which need to be described to the system. For instance, Baeza-Yates and Ribeiro-Neto (1999) give the following example of a possible set of requests:

“Find all the pages (documents) containing information on college tennis teams which: (1) are maintained by an university in the USA and (2)
participate in the NCAA tennis tournament. To be relevant, the page must include information on the national ranking of the team in the last three years and the email or phone number of the team coach.”

At this stage, the request does not need to be in some system-suitable encoding, as this example illustrates.

The search formulation process handles generating some representation of that information to supply to the system in order to find related materials. For example, it can include generalising to some overall ‘category’ within which the search can be narrowed in the library card catalogue system, or some keywords in the automated system. The query, in this form, can then be compared to the representations of the indexed items (categories, coupled with natural language descriptions, or some keyword search) to find the relevant documents through the similarity determining process. This basic underlying model has since been revised and added to include many facets of the searching process from both the system and the user’s perspective. In the scope of this research, most notably the inclusion of the iterative nature of the information seeking task. Saracevic (1997) and Belkin (1993) add the user’s interaction with the text as part of the information-seeking task. This means that both the user and the system adapt to the responses given until the best result is achieved. For instance, the user will adapt the way they are representing their information need (through changing the query) and the system, can adapt the responses it gives based on the user’s feedback (i.e. which documents they are looking at, how they are changing the query etc.).

Whether manual or automatic systems, the general processes described are still usually present – the actual implementation of them is dependent on the environment in which they are to function. Beyond this, two features are further required for an effective system – currency and completeness (Salton and McGill, 1983). Completeness states that the collection must contain a large proportion of items of potential interest, allowing greater chance of satisfying the users’ queries. Currency implies that the index is up to date with new material. In classic library systems this meant merely when new items are available, they are added to the index, but as described in Chapter 1 when dealing with Web based documents, this aspect becomes harder to accomplish. While in a library books once classified are typically unchanging (meaning only new items need to be
indexed) - the Web is a very dynamic environment. Not only would new items have to be found and indexed, but resources previously visited might have changed or disappeared meaning there is also a constant refresh effort required to satisfy this constraint.

The features of the Web described in Chapter 1 pose many challenges to information retrieval systems - mainly the size and dynamic nature of the collection, not only in terms of its growth, but also other changes in content already catalogued. The rest of Chapter 2 is dedicated to the description of various forms of solutions implementing the model described in Figure 2.1 which attempt to bring “order” to the Web and make information more accessible to those who need it.

2.2 Representations of the Web for IR

The aim of this section is to give an overview of the different ways the different features of the Web are represented for Information Retrieval. These features fill the ‘indexing language’ portion of the general system as shown in Figure 2.1. The features in question are the content of the documents and the links associated with them (to or from the document in question). Both these parts of the Web and their representations are used in different ways to achieve different ends. This sections aim is to give an introduction to these representations and to aid the discussion of their usage throughout the rest of this Chapter.

2.2.1 Document Content Representation

A document’s content can be described in many ways – some of which have already been mentioned in section 2.1. The most fundamental representation of the documents content is the full text itself (Salton and McGill, 1983), but this is only suitable when dealing with very small document collections due to the cost of searching through them all (either manually or automatically). Categorisation is an example of a classic indexing system (e.g. library card catalogues but also documents in the biomedical database PubMed\(^9\)), where documents are assigned headings, allowing the information seeker to look for documents under the heading that would best describe the general area they were interested in. Items would also be assigned natural language descriptions or keywords which would describe the topics covered so that the user can make a decision as

\(^9\) http://www.ncbi.nlm.nih.gov/pubmed/
to whether it would be beneficial to them. Historically this process was performed manually by “experts” (Salton and McGill, 1983). This manual system has been replicated on the Web in projects where teams of volunteers manually assign topics and descriptions to Web pages which are arranged taxonomically (see Section 2.3).

The categories and descriptions assigned to documents in this manual way can be subjective (Luhn, 1953) – different experts may provide different topic assignments to the same document dependant upon their specific interests or expertise. The same topic may also be described in different ways making the information difficult to find for the user. Automatic indexing techniques aim towards objective descriptions of the content of documents by using the words/terms present within it. This means finding the right features from within the text to suitably represent the main topics it discusses. Here we will discuss two such features and possible methods for their extraction from documents - words and terms.

Words can be described as being the “basic level of semantic richness” (Feldman and Sanger, 2007) – they are the basic linguistic tokens which make up the content of the document. It is possible for a word-level feature representation to contain a feature for each word in the document, meaning it is represented by the full text. Most of the time, some optimization is carried out, attempting to store only the words which are likely to distinguish the content of this document from others. This can be achieved using statistical analysis as set out by Luhn (1958) to compute a “significance factor” of words based upon their frequency within the document. Luhn (1958) hypothesised that words which appear too frequently, or too sparsely were not useful in determining the content of one document from another – leaving the mid-level words as the most significant and therefore valuable from an IR perspective.

Terms can be single or multi-word phrases which are meant to be “generally representative” of the content of the document (Feldman and Sanger, 2007). They are the “linguistic representation of concepts” (Frantzi et al., 2000) and are therefore the most “semantically rich” features, making them more suitable for IR than word level features (Feldman and Sanger, 2007). Historically, terms were chosen to represent a document by human indexers from a controlled vocabulary or dictionary list (Salton and McGill, 1983). These were used mainly to aid the indexer in choosing the correct term to describe some
abstract concept in a document. Problems with this method arise when new terms are adopted by a domain to describe new concepts (Nenadić et al., 2002).

Automatic term recognition aims to address the problems of dynamic trends in domain specific terminology by providing efficient means of discovering terms within a document. This can be done through statistical analysis, linguistic analysis or a hybrid approach which utilizes both. One such hybrid approach is the C-value/NC-value method (Frantzi et al., 2000). The linguistic analysis is performed first, with the following steps:

1. Part-of-speech tagging
2. Linguistic filtering
3. Stop-List filtering

The overall outcome of the linguistic analysis is a list of term candidates based on linguistic patterns formed within the text. In this case, noun phrases are extracted as they typically form concepts within text. The stop list is a list of words which are not expected to occur in terms. This means that candidate terms which contain these words are unlikely to be terms and can be removed from the candidate list.

The statistical analysis assigns a termhood score to each term candidate, meaning that the list can be ranked and low scoring terms can be disregarded. The statistical characteristics of the candidate terms combines (Frantzi et al. 2000):

a) The total frequency of the candidate term in the document;

b) Frequency of the term as part of other longer candidate terms, and the number of such longer candidate terms;

c) The length of the candidate string (number of words).

Candidate terms which occur as part of other longer candidate terms are known as nested terms. The C-Value is calculated differently for nested terms than for non-nested terms. The C-Value score for a candidate term \(a\) is calculated as follows

\[
C-value(a) = \begin{cases} 
\log_2 |a| \cdot f(a) & a \text{ is not nested}, \\
\log_2 |a| \cdot f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) & \text{otherwise}
\end{cases}
\]

Eq 2.1
Here \( f(a) \) is the frequency of the occurrence of \( a \) in the document, \( T_a \) is the set of candidate terms which contain \( a \) and \( P(T_a) \) is the number of these terms. Although the method is based on the frequency of a particular term candidate in the document, penalties are applied if these occurrences are part of other longer terms. This means that for each document, a ranked, scored list of candidate terms can be found which can be assumed as being ‘generally representative’ of its content.

Applications of the three document content representations discussed here will be covered throughout Chapter 2 with details of the retrieval methods employed with each. In Section 2.3 the online Web directories will be discussed which employ the ‘heading’ style descriptions of content. Section 2.4 deals with automatically creating indexes using Web crawlers and Section 2.5 covers searching through a word or term based index using keyword queries.

### 2.2.2 Document Link Representation

Links can give useful information on individual Web pages and the “communities” in which they lie (see Section 2.6.2). Documents in the Web can have *inlinks* (references in other documents to it) and *outlinks* (references to other documents) (see Fig. 2.2).

**Figure 2.2:** Linking in the Web

Links within the Web can be viewed as directional graphs, where the nodes are pages/resources and the directed edges are the hyperlinks between them (Chakrabarti et al., 1998; Kleinberg, 1999; Henzinger, 2000). For instance, if there are two pages \( P_1 \) and \( P_2 \), where \( P_1 \) contains a link to \( P_2 \), they could be represented like Figure 2.3.

**Figure 2.3:** Hyperlinks as edges
Using directional edges helps represent the navigational properties, as by using hyperlinks one could navigate from one page to another, but not back again. For instance, a user could travel from $P_1$ to $P_2$ as there is a hyperlink, however without another hyperlink from $P_2$ to $P_1$ there is no inherent method in the Web’s structure to go from $P_2$ to $P_1$. For anyone with experience using the Web through an internet browser, the ‘back’ button will be a familiar feature, but this is a feature of the browser itself, not one inherent in the Web’s structure.

It is also possible to represent the web as a matrix. This can be useful when making calculations as information about which page links to which is directly available with no need for further calculation.

For example the graph in Figure 2.4 can be represented as a matrix, as shown here.

\[
G = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[(G_i, G_j) = n \text{ iff there are } n \text{ links from } G_i \text{ to } G_j\]
Here, the column and row vectors represent pages, with the row values representing the number of outlinks the page has, and the column vector subsequently representing the number of inlinks it has. Given the size of the WWW and the fact that each page will only link to a small subset of the overall collection, matrix representations tend to be extremely sparse as even the example above demonstrates.

2.3 Manually Cataloguing the Web

As briefly discussed in Chapter 1, the Web is a dynamic and distributed set of semi-structured documents (Pinkerton, 1994; Baeza-Yates and Ribeiro-Neto 1999). This means that finding information within it can be hard (McBryan, 1994; De Bra and Post, 1994). What many people have set out to do is to gather information about many or all of the sites on the web in one place, so that if a user wants to find some information, they can search this collection of information and be pointed to a possible good source of documents.

This process can be done manually. People would visit uncategorised websites, and decide on the topics they best fit into or the way to describe them properly within a topic taxonomy (Berry and Browne, 2005). For example, the Open Directory Project (see Fig. 2.5), the WWW Virtual Library and Yahoo Directory are collections of documents which are manually indexed. These sites allow its ‘editors’ to group websites together into a topic taxonomy. This helped form what are known as Web Directories, a catalogue of websites organised by topics through a taxonomic structure. This means that if a user wants to find Web pages on a certain topic, they can follow this topic tree down, selecting the appropriate, more specific topic at each level. For example, someone looking for a site on ‘cycling’ could follow the tree in Figure 2.5 down through the various levels and find sites relevant to this topic at the end. Under the general topic ‘sports/cycling’ there are a lot of other sub-categories (Fig 2.5 – 2.6), which means that the user can find the specific sort of information they want to find. Once the topic becomes refined enough, there will be a list of links to web pages which are about that specific topic.

10 http://www.dmoz.org
11 http://vlib.org/
12 http://dir.yahoo.com
These Web Directories attempt to overcome the ‘unstructured’ problem of the Web, but not by creating new inherent structure in it. Instead, the projects build indexes which allow users to find what they want without knowing the address. It is also possible for the editors to pick out important terms from the document to enable users to perform
keyword searches rather than navigating the taxonomy, although this would require more effort and therefore be more time consuming. With these sorts of references, the WWW becomes easier to navigate and information more widely and easily obtainable.

The Open Directory Project also tries to overcome the quality problem. Each site recommended has a description of the services or information the site has to offer. This means the user has a more informed idea of what lies at the other end of the link before they follow it.

The main advantage this method has is that all the decisions about whether to include a site, what category to include it in and what description to assign to a site are made by qualified humans. This means that the decisions would typically be highly accurate as humans will hopefully understand the content. Computer-based decision applications (e.g. document classification (Li and Jain, 1998)) applied for the same task would potentially make more errors than human based ones because they do not “understand” the content, they merely make and apply generalizations.

Berry and Browne (2005), however, note that manual indexing efforts like these are extremely subjective and that decisions made can vary dramatically depending on the background and personality of the individual editor. The process of manual categorisation is expensive and time consuming which has become less viable with the growth of the World Wide Web. As one can see from statistics on the Open Directory Project website (as one example), they currently have around 82,392 editors and have 4,570,989 Web pages catalogued (as of March 2009). This figure against the nearly 240,000,000 sites that are registered on the WWW and the estimated 11.5 billion pages accessible on them (Gulli and Signorini, 2006), despite being large in itself, is a very small fraction of the entire web.

Although this method overcomes some of the problems with finding information on the WWW, it is not scalable enough to take advantage of the whole of this resource. Even voluntary schemes like the Open Directory cannot get the workforce needed to catalogue a significant portion of the Web, especially not with certain highly specialised parts of it. Crowdsourcing\textsuperscript{13} – where work is shared amongst a large number of online participants, each performing a very small part of the overall task for a small reward -

\textsuperscript{13} http://www.wired.com/wired/archive/14.06/crowds.html - Accessed 19/02/2010
offers a possible solution to this problem, as demonstrated by Alonso et al. (2008, 2009) who used the crowdsourcing model to retrieve manual classifications for 2500 query-result pairs. Social bookmarking (where users in a community share bookmarked pages which are manually described with user set keywords known as tags) data has also been suggested as another source of large-scale manual annotation of Web pages, and has been used to automatically classify pages within the general ODP topics (Zubiaga et al., 2009). This helps not only with the content of the pages, but also the emotive language to describe the quality of the resources within them (Yanbe et al., 2007).

2.4 Automatically Cataloguing the Web

Web directories (like the topic hierarchies manually created discussed above) have evolved into search engines, which most internet users today are likely to have used at some point to find information from the Web\textsuperscript{14}. Search engines take a generally automated approach to cataloguing the Web. Rather than the user selecting the topics they wish to find information on, with a search engine the user will generally provide a set of keywords that describes the topic and the search engine will return a list of appropriate websites (Berry and Browne, 2005). To achieve this, automatic Web crawlers can be employed to index sites on the Web to allow searching (McBryan 1994; Pinkerton, 1994; Brin and Page 1998). The organisation of these search engines and the Web crawlers that drive them is discussed in this section.

2.4.1 Structure and Implementation

As discussed above, Web crawlers are generally implemented as part of a search engine system. Search engines, like Google, Yahoo and Ask accept queries from the user and return a list of websites which are considered relevant. The general structure of search engines will be briefly discussed to place the web crawler into context within the searching system as a whole (Fig. 2.7).

\textsuperscript{14} During December 2008 Search Engine Watch reported over 8.5 Billion searches on the top 10 search engines alone - http://searchenginewatch.com/3632382
The system can be split up in to two separate parts: the search engine and the Web crawler (as described by McBryan (1994) and Pinkerton (1994)). The search engine’s job is to interface with the user, get the query and return the results by referring to a pre-computed index. The web crawler gathers websites from the Web and stores information about them in the index and keeps this information up to date by periodically revisiting indexed sites.

Web Crawlers take advantage of the links in the web graph (see Section 2.2.2). Traversing this link structure means that Web crawlers can visit a large number of sites and pages gathering information about them (McBryan, 1994; Pinkerton 1994, Chakrabarti, 2003). The ability to catalogue the entire web graph depends on how well connected individual nodes are. If some pages are not linked then there would be no way for the web crawler to reach them (McBryan, 1994).

Figure 2.8 represents the main processes carried out by most web crawlers. It will traverse the Web by exploiting the link structure to find pages and extracting features (i.e. keywords/terms or images etc.) depending on the kind of index desired (Heydon and Najork 1999).
Within any crawler implementation, two lists of URLs will be maintained, the ‘to-do’ queue and the ‘done’ list. As URLs are unique, any given URL will at most appear in one of these lists. The done list is created so that links that are discovered more than once during the crawl are not repeated in the to-do queue and are therefore not crawled more than once. It is important to note that the order in which URLs are visited in standard crawlers is not generally important as the aim of the crawl is to generate a snapshot of the entire Web graph. This means it does not matter which pages are crawled first because, in the end, the same snapshot should be created. Still, most Web Crawlers implement a breadth-first exploration of the Web graph.

The indexes created are used for searching based on the features created. For example, many search engines use keywords, where the index is based on keywords found within indexed portions of the visited documents (for example full text as with Google (Brin and Page, 1998) or page titles as with WWWWorm (McBryan, 1994)). This index
would then be searched through by supplying keywords or phrases a user would expect to find in the documents pertaining to their desired topic (see Section 2.5).

2.4.2 Problems Overcome by Web Crawlers

The Web crawler can be thought of as an automatic version of the manual solution discussed in section 2.3. So how does automating this process improve it?

- **Human Effort** – To catalogue the ever growing, ever changing data source that is the WWW requires a constant effort. By using an automated system rather than a human-based solution, the main on-going human effort is reduced to a maintenance one rather than the main driving force. This maintenance will probably only be an intermittent occurrence.

- **Processing Time** – When one of the editors discussed in section 2.3 visits a site to classify it, they will have to gain an understanding of the content. Information retrieval methods applied in the web crawler allow it to approximate this information without gaining this understanding or ‘learning’ anything from it which means that the operation is less time consuming. The crawlers can also be parallelised to run across multiple machines (as with Google (Brin and Page, 1998)) or multiple threads (as in Mercator (Heydon and Najork, 1999)). As discussed in section 2.3, for certain sites containing complex expert information it will require someone well versed in the area concerned to decide on a specific topic to assign the page, or may take someone without this expert knowledge longer to process, whereas an automated system may be able to extract reasonable information without the background knowledge and therefore understanding a human would need. The exclusion of understanding of the content may give better processing time, but often at the expense of the quality of the features and classification produced.

- **Cost** – Consequently having reduced the time that cataloguing will take and the amount of human effort each crawl will take, the costs incurred will be reduced as there is a much smaller workforce and a lot less resources required for it. This means that commercial ventures such as the search engines seen today are more viable.
2.4.3 Problems Posed by Web Crawlers

Although automatic Web crawlers offer a lot of solutions to the problems posed by the manual method, they still pose some new challenges. With the growth of the Web over the past decade, the rate of which is still increasing, it is still incredibly difficult and time consuming to catalogue the reachable Web even using the automated method over the manual one.

With some websites changing constantly, information that the index holds can become out of date very quickly. Brewington and Cybenko (2000) estimate that for a Web index to remain 95% accurate over a week (i.e. 95% of the index pages are indexed as they were at some point in the last week) a web crawler would have to download 45 million pages per day. This rises to 94 million a day working with a daily rather than weekly scale. Bare in mind that these estimates were computed when the size of the Web was estimated at only 800 million pages (Lawrence and Giles, 1999) rather than the more recent estimate of 11.5 billion (Gulli and Signorini, 2005) and these figures are probably optimistic on the modern Web.

Another problem with this sort of crawling strategy is finding suitable methods for searching (including representing information need, finding the documents and presenting them to the user). The most common method in search engines is keyword-based queries which are processed in various ways (see Section 2.5). Studies carried out recently (Beitzel et al., 2004; Kruschwitz, 2005; Efthimiadis, 2008) found through analysing search logs for popular search engines that the average query length was around 2 words (2.2, 1.72 and 1.8 respectively). Using this sort of query alone does not seem to provide enough information to find the best resources from billions of potential candidates, or even derive a suitable information intention from the user in a lot of cases. Using a query of such a limited size against a large index covering so many forms of information means that many results returned may not be truly relevant to a user’s information need. Compared to this approach, one benefit of the directory structure is that as it is being navigated through by the user, they will be able to make links between concepts and find information using abstract routes which would be difficult to represent in a simple keyword query. Kruschwitz (2003) worked towards merging these ideas by allowing users to refine their keyword search according to automatically extracted topic themes from within the search
results. This was presented in a taxonomy-like way with general topics leading to more refined ones through a tree structure. This would allow the user to find the specific information they were looking for without searching manually through the pages of results returned from their initial query. Ranking strategies have also been used to overcome this problem, where the list of results is sorted in the order of their probability of satisfying the user’s query. This means that a low precision becomes less of a problem, as the user will be presented with the expected best results first – reducing the need for searching through results. These ranking schemes can be based on various similarity or popularity methods (covered in section 2.6). In the following section we discuss search strategies in more detail.

2.5 Searching the Index

Section 2.4.1 set out how Web crawlers attempt to tackle the representation of the Web, by visiting each page and storing some information on each page visited (usually keywords from the text, titles and headings etc. as used in Google, MSN, Yahoo! and Ask). Keyword queries are generally built around the assumption that they specify words a user would expect to find (or not to find) in documents pertaining to their interest. Keyword queries can be one or two words, or even full phrases that encode the user’s information need. This section covers some ways in which these queries are compared to the indexes to retrieve the documents best matching the users query.

Single Word Queries

The simplest form of keyword query is just a single word expected to be in the user’s desired document (Beaza-Yates and Ribeiro-Neto, 1999). With this simplicity comes vagueness. While a single word might retrieve many documents, probably including the desired ones if present in the collection (high recall) it is also likely to retrieve many irrelevant documents (low precision) making the desired information difficult to find. It is likely that the user would have to search through the resultant documents or refine the query used to find the correct information.

Boolean Queries
The Boolean query is an extension on the single word query where more specific information can be passed to the search engine using logical operators (i.e. AND, OR, BUT etc.) on a set of single keywords. The aim being that the user can specify more constraints on the kind of document that can be returned, increasing the precision of the resultant collection. Figure 2.9 gives an example of a Boolean query syntax tree (Baeza-Yates & Rebeiro-Neto, 1999).

![Boolean Query Syntax Tree](image)

**Figure 2.9:** Example Boolean query syntax tree

This query (Fig. 2.9) will return documents including the word ‘translation’ and either ‘syntax’ or ‘syntactic’. Baeza-Yates and Rebeiro-Neto (1999) and Berry and Browne (2005) point out however, that users not trained in Mathematics or related disciplines find Boolean queries of this kind difficult to formulate. Successful Boolean queries may also be formed over a series of iterations, adding or relaxing constraints based on the results of previous searches. This elongation of the searching process can be frustrating to users.

**Phrase Queries**

According to Berry and Browne (2005) searching through terms is the most prevalent method used on the Web. This is where the user supplies a query in the form of ‘a few’ words or phrases related to their desired information. This allows the proximity of the keywords within a document to be taken into account when judging relevance. The example Berry and Browne (2005) give in particular is the phrase ‘Atlanta Falcons’ – the name of an American Football Team. This form of query can increase precision at the expense of recall. For instance, with the example query if an article read

“In Atlanta, the Falcons host their arch rivals…”
it may not be judged as relevant as other documents because the keywords are not together. One main advantage of this sort of query is that it requires no search-specific expertise, coming up with phrases to achieve the best results may take a few iterations.

**Matching Single Word/Phrase Queries**

Vector space models attempt to represent terms and documents in a text collection in a way that conveys not only a terms/keyword/phrases occurrence in a document, but the *semantics* of that occurrence (Berry and Browne, 2005). This is down to the weighting schemes generally employed by them, which give some sort of measure of a term’s significance within the document. This allows the efficient search of a large corpus, the incorporation of weights in to the query matching process (useful for ranking the query results) and a more fuzzy search. The following example is used by Berry and Browne (2005) to explain how this is achieved.

**Documents**

| D1: Infant & Toddler First Aid | T1: Bab(y,ies,y’s) |
| D2: Babies & Children’s Room (For your Home) | T2: Child(ren’s) |
| D3: Child Safety at Home | T3: Guide |
| D4: Your Baby’s Health and Safety: From Infant to Toddler | T4: Health |
| D5: Baby Proofing Basics | T5: Home |
| D6: Your Guide to Easy Rust Proofing | T6: Infant |
| D7: Beanie Babies Collector’s Guide | T7: Proofing |
| T8: Safety |
| T9: Toddler |

Given the document titles (left), a subset of keywords (right) are chosen to describe them. Words are generally chosen with respect to some ‘stoplist’ (a list of words which should be removed from the resulting index as they have limited impact on discriminating concepts). Here, for instance, words like ‘first’, ‘your’ and ‘basics’ have been excluded.
from the terms list. Another feature of the normalization of this index is stemming, which is the reducing of a word to its root form (as seen with “babies”, “baby” and “baby’s” in the example) (Porter, 1980). This allows occurrences of variations of words be considered together.

These document titles and the keywords contained in them can then be represented in the form of an \( m \times n \) matrix where the \( m \) rows represent the keywords and the \( n \) columns represent the documents. In this case the 9x7 matrix is as follows:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

**Figure 2.10:** Construction of a term-by-document matrix (Berry and Browne, 2005)

In this matrix, \( a_{ij} \) is the frequency of term \( i \) in document \( j \). In this case, all the terms have an occurrence of 1 in their prospective documents, but when using full text documents it is likely that some occurrences would be more than 1, while some would be 0.

This allows the searching of the index using queries. For instance, to search this index for ‘Child Proofing’, this query can be encoded as a term vector:

\[
q = (0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0)^T
\]  
(Eq. 2.2)

representing the frequencies of the terms ‘child’ (term 2) and ‘proofing’ (term 7) in the query. We can then compute the similarity of each document (\( a_i \)) to the query (\( q \)) to allow the relevant documents to be returned using, for example, the cosine similarity.
It would also be possible to compare the keywords to the query using other measures such as the Dice or Jaccard Coefficient. Some threshold would usually be applied to only return significant matches. Browne and Berry (2005) point out that a threshold of 0.5 using this index and query would return the fifth and sixth documents. While the fifth is relevant, D6 regarding rust proofing would also be included. Also, documents 2 – 4 would also be excluded which are probably related to the users query. This is where term weighting is used in the matrix representation, where each element in the index is defined by (Berry & Browne, 2005):

\[ a_{ij} = l_{ij} g_i d_j \]  

(Eq. 2.4)

where \( l_{ij} \) is the local weight for term \( i \) in document \( j \), \( g_i \) is the global weight for term \( i \) in the collection and \( d_j \) is a document normalisation factor. This allows the importance of terms, both local to the document under consideration and in the collection as a whole, to play a part in relevance decisions and hopefully achieve a better precision and recall than the example using the un-weighted index given above. An example of this form of weighting is the term frequency/inverse document frequency (\( tf^*idf \)) weighting scheme (Salton, 1983). Here, the local weight is the local term frequency (the frequency of term \( i \) in document \( j \) ) and the global weight is the inverse document frequency (the inverse of the frequency of term \( i \) in all documents across the collection). This allows the importance of the term within the document, and its significance within the collection as a whole to determine the weight of the term.

Keyword queries can fall victim to problems caused by polysemy and linguistic variation amongst disciplines when using very general collections. For instance, the word ‘mole’ can be taken in many different contexts depending on the discipline in which it is used (polysemy). For instance, it can mean “a small, congenital spot or blemish on the
human skin” or “any of various small insectivorous mammals”\textsuperscript{15} depending on the context and discipline in which it is used. When matching keywords this can be a problem as which “mole” the user means cannot always be conveyed without additional information (Kruschwitz, 2003). This means that when ambiguous terms are used within a keyword query, the user may be faced with a large number of irrelevant results (lower precision), meaning further searching through the results or a reformulation of the query is going to be needed.

Variation in the way concepts are referred to within a discipline (synonymy) can also cause problems for keyword based queries. For example within the biology domain ‘carcinoma’ and ‘cancer’ are interchangeable, but this acceptable variation cannot be expressed easily in a keyword query. Documents returned for a keyword query will not return documents containing synonyms for the keywords provided. If the keyword given for a query was ‘cancer’ documents containing the word ‘carcinoma’ may not be returned even though the two words are equivalent. This means not all the best documents will be returned (lower recall).

**Probabilistic Information Retrieval**

Figure 2.1 shows the overall structure of a general IR system, one that, from the user’s perspective, starts with some underlying information need that is translated into some query which is some representation of it. Methods discussed previously (i.e. Boolean and Vector-based methods) have focused mainly on returning documents which are close to the query posed to the system. Probabilistic methods however, focus more on the information need that the query represents and aim towards returning those documents most likely to satisfy it. This section covers the general area of Probabilistic Information Retrieval, some more specific implementations of it and how they can relate to Web search.

As already mentioned, Probabilistic Information Retrieval methods are focused on attempting to estimate the likelihood that, given the query, the user would find individual documents within the collection relevant. More formally, for a document $d$ and a query $q$ let $R_{d,q}$ be some indicator variable which is 1 if $d$ is relevant to $q$ and 0 otherwise.

\textsuperscript{15} http://dictionary.reference.com/browse/mole - accessed 18/06/2009
Probabilistic IR methods try to find the probability \( P(R=1|d,q) \). To achieve this, they generally make varying assumptions about the text with which they work and it is here where the differentiating factors between the probabilistic models lie.

**Binary Independence Retrieval Model**

The Binary Independence Retrieval (BIR) model (Robertson & Sparck Jones, 1976) uses binary vectors to represent both documents and terms. So, a document \( d \) can be represented as a vector \( x = (x_1, ..., x_m) \) where \( x_t = 1 \) if the term \( t \) is present in document \( d \) or \( x_t = 0 \) otherwise. Likewise, we can represent the query \( q \) as the vector \( \tilde{q} \). This representation also includes the assumption that there is no relationship between the terms occurring in a document that each term occurs in a document independently of any other term. The binary representation and this independence assumption are what gives the BIR model its name. Manning et al. (2008) note that while the independence assumption is naive for a model of natural language, models which use it still manage to perform well for classification tasks.

Using these notations, the problem of classification of document \( d \) becomes \( P(R=1|\tilde{x},\tilde{q}) \), i.e. the probability that \( d \) is relevant given its representation \( \tilde{x} \) and the query representation \( \tilde{q} \). Using Bayes rule, we get

\[
P(R = 1 \mid \tilde{x}, \tilde{q}) = \frac{P(x \mid R = 1, q)P(R = 1 \mid q)}{P(\tilde{x} \mid \tilde{q})} \quad \text{(Eq. 2.5)}
\]

\[
P(R = 0 \mid \tilde{x}, \tilde{q}) = \frac{P(x \mid R = 0, q)P(R = 0 \mid q)}{P(\tilde{x} \mid \tilde{q})} \quad \text{(Eq. 2.6)}
\]

Without information about the relevant documents in the collection, the prior probabilities on the right hand side of these equations are difficult to estimate, so the problem is made into one of ranking using statistical information which is easier to find. The following equation (Manning, 2008) is the Retrieval Status Value (RSV) and is a form of the original ranking equation from Robertson and Sparck Jones (1976) (for full derivation the reader is advised to refer to these texts).
\[
RSV_d = \sum_{i, r_i = q_i} \log \frac{p_i}{(1 - p_i)} + \log \frac{1 - u_i}{u_i}
\]  
(Eq. 2.7)

where \( p_i = P(x_i = 1 \mid R = 1 \mid \tilde{q}) \), the probability of a term appearing in a document relevant to the query and \( u_i = P(x_i = 1 \mid R = 0, \tilde{q}) \), the probability of a term appearing in a document which is not relevant to the query. This reduces the ranking problem to the log-odds ratio of the presence of the query terms in relevant and non-relevant documents. The ordering model then falls to estimating \( p_i \) and \( u_i \). These probabilities can be estimated in many ways, or often guessed at in the first instance, but are tuned afterwards using search iterations and user feedback as to the relevance of the documents returned. In the Web search world, this can be supplied through click-through data. For instance, the results that the user ‘clicks’ on from all the options returned could be assumed to be the most relevant ones. This allows the probabilities for this query term-document relationship to be adjusted to give it more weight in future.

**Okapi BM25**

The BM25 weighting scheme (Spark Jones et al., 2000), which is also known as the Okapi weighting scheme after the system in which it was first implemented, extends the ideas present in the BIR model to include information about term frequency within the documents and the documents length. Initially, a term’s weight can be computed as:

\[
CW = \frac{TF_i(k_1 + 1)}{k_1 \times \left(1 + b + b \frac{L_d}{L_{Avg}}\right) + TF_i} \log \frac{N}{n_i}
\]  
(Eq. 2.8)

Where \( TF_i \) is the frequency of term \( i \) in the document, \( L_d \) is the length of document \( d \) and \( L_{Avg} \) is the average length of the documents in the collection. \( k_1 \) and \( b \) are tuning constants. \( k_1 \) affects how the weight reacts to increasing \( TF \). During their experiments with a TREC corpus, Sparck Jones et al. (2000) note that a value of \( k_1 \) between 1.2 and 2 to be effective. The second tuning constant, \( b \) is set to between 0 and 1 and dictates the effect the normalisation over the document length has on the overall weight. Again, the experiments with the TREC corpus showed values of \( b \) around 0.75 to be most effective.
On the right hand side of the $CW$ formula, the $\log \frac{N}{n_i}$ portion (where $N$ is the number of documents in the collection and $n_i$ is the number of documents in the collection including the term $i$) includes some measure of significance of the terms contribution to the weight. For instance, a term which occurs in the document $d$ but also in the majority of other documents should carry less weight than a term which occurs in $d$ but not many other documents in the collection. This assumption is born from the assumption that the number of documents the user will find relevant within the collection will be relatively small.

The formula $CW$ is the more naive form of the BM25 formula, where no retrospective relevance information is yet available. The $CIW$ formula (below) takes into account this information through user relevance feedback. Once results have been returned for queries involving the terms using $CW$, the results the user marks as ‘good’ can help improve the way the term-document relationships are interpreted in future queries.

$$CW = \frac{TF_i(k_i + 1)}{k_i \times \left(1 + \frac{b \cdot \frac{L_d}{L_{avg}}}{(1 - b) + b \cdot \frac{L_d}{L_{avg}}}\right) + TF_i} \log \frac{(r + 0.5)(N - n - R + r + 0.5)}{(R - r + 0.5)(n - r + 0.5)} \quad (\text{Eq. 2.9})$$

The left hand part of $CIW$ is the same as that of $CW$, but the relative term-document relationship (the right hand side) makes use of relevance information. Here, $R$ is the total number of relevant documents, and $r$ is the number of relevant documents that contain the term $i$. The addition of the 0.5 allows for where these values are 0. As stated before, this relevance information can be retrieved through searches done in the past using similar queries. On the Web, this sort of information can be retrieved by the documents related to the terms present, which have been clicked on by users previously searching using similar terms.

Both $CW$ and $CIW$ are used to find weights for the terms used to describe the query for each document (i.e. over the query terms that appear in the document), and can be combined together by summing them up to give an overall ranking score for the each document.
2.6 Ranking Retrieved Documents

When searching for information on the Web, it is imperative to try to pick out the best information from the probable vast amount available on any specific topic. To do this, a number of ranking strategies have been developed, based on various kinds of information available about individual resources on the web. There are two general approaches which are based on the content of a webpage and the way in which the page is linked to. In this section, some of these methods will be introduced and analysed.

2.6.1 Content-based Methods

When ranking information with regard to how important it is expected to be to the user’s information needs, it makes sense to look at how closely the page fits the user’s query. Similarity measures like the cosine similarity model (discussed in section 2.5) allow a computation of ‘closeness’ to the user’s query. Similarly the probabilistic models provide a probability that the page will be relevant to the user’s query. This allows ranking of the results based on the content of the document and the expected worth to the user (either closeness to the query, or a probability of relevance).

Ranking in this way could be seen as “short sighted” in terms of measuring the importance of documents in a collection towards some user specified query. A document which contains certain words more times than another document is not necessarily more important. Kleinberg (1999) sets this out as the ‘abundance problem’ – the fact that the number of documents that could potentially be returned for some query is too large for a human user to digest. He suggested that some analysis of each page’s ‘authority’ as a source of information must be assessed in some way as well as expected topical nature.

2.6.2 Graph-based Methods

Graph-based methods attempt to address this measure of ‘authority’ by looking at a page’s popularity within the Web. Firstly, it is necessary to explain specifically what the notion of authority refers to and how it can be measured. Kleinberg best explained the notion of authority within the web: ‘the creator of page p, by including a link to page q, has in some measure conferred authority on q’ (Kleinberg, 1999).
The more links to a page from other pages, the more authoritative that page is, as many authors thought it important enough to cite. We can assume that authors of webpages on some topic will know about that topic. Therefore, if many pages on a certain topic link to one page in particular, we can assume that in this field, this resource is an important one. This basic measure of ‘in-links’ can offer some idea of which pages are more important than others. However, it poses many problems. For instance, Kleinberg (1999) notes that in his experimental querying setup, when he searched for ‘java’, the pages returned with the largest in-link counts were ‘www.gamelan.com’ and ‘java.sun.com’, closely followed by the Amazon\textsuperscript{16} homepage and a website advertising Caribbean holidays. This means that some further calculation needs to be done to separate the important pages from the popular ones.

This is the main motivation behind the methods described below. They attempt to counter this phenomenon by adding a little more depth to the calculation than merely how many times they are referenced. The main idea is to find how many times pages are referenced by other good pages.

2.6.2.1 HITS

The motivation behind HITS is Kleinberg’s observation that pages which are all relevant to some query should not only have a high in-link count, but there should be a large overlap in the sets of pages which point to them, as they are all part of some sort of Web community (Kleinberg 1999). This leads to his definition of hubs and authorities. Authorities are the pages which are often linked to, and are therefore the pages within which we expect to find the best information. Hubs are pages which link to a lot of other pages. This mutually reinforcing relationship is demonstrated in Fig. 2.11.

\textsuperscript{16}http://www.amazon.com
The relationship between hubs and authorities can be formalised with the scoring scheme set out by Kleinberg. Each page $i$ has a Hub score ($h_i$) and an Authority score ($a_i$) which tells us the degree in which the page is considered a hub and an authority. The scores are computed as follows:

$$a_i = \sum_{j \in B_i} h_j \quad \text{(Eq. 2.10)}$$

$$h_i = \sum_{j \in F_i} a_j \quad \text{(Eq. 2.11)}$$

where $B_i$ is the set of ‘in-links’ (pages which link to a page) for page $i$, and $F_i$ is the set of out-links (pages which the page links to) for page $i$ (for in-links and out-links, see Section 2.1). A good Hub is a page which links to many good Authorities, and a good Authority is a page which is linked to by many good Hubs. This relationship allows for the ranking of pages as important in communities, and also the ranking of pages as directories within web communities.

It is important to note that while the basic in-links measurement of authority is a local measure (it deals only with the direct context of pages in the web graph), HITS is a global measure computed across and influenced by the entire web graph. This means to find the Hub and Authority scores of pages within the web, a snapshot of the entire web is required.

Borodin et al. (2001) analysed the HITS algorithm, and noticed a situation where the algorithm could provide non-intuitive results (see Fig. 2.12).
Figure 2.12: A bad example for HITS (Borodin et al., 2001)

Figure 2.12 contains two example Web graphs highlighting situations which would result in disproportionate authority being inferred on certain nodes. For instance, if there were more white authorities than there were black hubs, then all the weight would be given to the white authorities, while giving no weight to the black hub. This is because the white hub is deemed better, giving all the weight to the white authorities. But given the black authority is linked to so many times in the Web graph, one would assume it would be the more important resource than the larger group of unpopular pages. This sort of problem is blamed on two properties of the HITS algorithm. Firstly it is symmetric – both hub and authority scores are treated in the same way. Secondly it is egalitarian – when computing the hub score (or authority score) of a page p, all the pages that are pointed to from p (or point to p) are treated equally.

2.6.2.2 PageRank

PageRank (Page et al., 1998) is a system implemented in the Google search engine to rank search results returned. The basis of this algorithm is the ranking of pages in the entire Web graph rather than just locally. This model is based on a ‘random surfer’ (a user visiting web pages) who will traverse the web in the following way:

- Start at a page chosen by some distribution \( D \) and then
- With probability \( \alpha \), the surfer will follow a link picked uniformly at random;
- With probability \( 1-\alpha \), the surfer will jump to a random page chosen according to some distribution \( D \).
where $\alpha$ is given as a parameter. The authority of a page $p$ is the amount of time the random surfer will spend at that particular page. The PageRank score of a page $j$ can be calculated as

$$P(j) = \frac{(1-\alpha)}{N} + \alpha \sum_{i \in B_j} \frac{P(i)}{|F_j|}$$

(Eq. 2.12)

where $B_j$ is the set of pages which point to $j$ and $F_j$ is the set of pages which $j$ points to and $N$ is the number of documents. This method is more inclusive than the HITS method, as it accounts for the possibility of nodes without in-links. PageRank can also be computed as the dominant eigenvector of the probability transition matrix of the random walk the ‘surfer’ follows. A page which has no links to it will still have the possibility of being visited thanks to the random jump parameter $(1-\alpha)$. This parameter adds the use of addresses as a navigation technique as well as traversing the Web’s link structure. In practical terms, this may be a site which was provided to the system (i.e. does not have to be discovered by a crawler) but is not linked to in the web graph. A page’s PageRank score dictates where it should appear in a list of results returned to a user when it matches their query. High PageRank means that the page is viewed as ‘authoritative’ and therefore, should be considered a priority for consideration by the user (should it match their query).

### 2.6.2.3 Topic-Sensitive PageRank

One of the main variations on the PageRank theme has been Topic-Sensitive PageRank\(^{17}\) (Haveliwala, 2002). This method attempts to allow the query the user supplied to have an impact on the order the results are displayed. When using PageRank as an ordering metric in returning search results, the query generally only has an impact with regard to which pages are returned. The PageRank scores of the individual pages decide in what order they are displayed. This results in them being ordered in terms of their global importance, not necessarily their importance within whichever field the query was concerned.

Topic-Sensitive PageRank attempts to add bias to the importance scores by preempting the field with which the users query is concerned. Haveliwala proposes a method of computing many different PageRank scores, each with a bias towards a certain topic.

---

\(^{17}\) This method is separate from topic-focused Web crawling (covered in Section 2.7) – Topic-Focused PageRank is used as an ordering metric for search results.
These topics are obtained using the Open Directory Project’s topic taxonomy, computing what Haveliwala calls ‘personalization vectors’ offline for each top level topic of the ODP. When a query is submitted, the best representation of the topic is found in the page index (separate from the ODP), and the results are ordered according to biases towards certain kinds of information and the resultant pages comparison to the personalisation vector. An example given in the text is the query “Cycling”. The results list when biased towards the “Health” topic provides links about the fitness aspects of cycling, like personal trainer sites etc. When biased towards “Arts” pages about photo contests were retrieved where the topic was “Bicycling”.

2.6.3 Relevance Feedback

Relevance feedback (Maron & Kuhns, 1960; Rocchio, 1971) is the process of refining the query vector from historical search data, attempting to group it with documents already identified by past users as relevant, in the hope that it will also group it with documents not yet returned (Kemp & Ramamohanarao, 2002). This kind of metric has been used to rank search engine result sets, based on the relevance judgements of past users with the same or similar queries (Joachims, 2002; Kemp & Ramamohanarao, 2002; Joachims et al. 2007; Shokouhi et al., 2008). Traditionally, explicit user feedback would be used, where through user surveys or explicit user feedback data is gathered from the users of an IR system that conveys how relevant the resources were. As clickthrough data is more implicit, it requires more interpretation - there are many reasons a user might click on a resultant link that are not its expected relevance. Therefore, clickthrough data is used as a relative measure of importance (i.e. ‘as the 2nd result was not clicked on but the 3rd one was, the result ranked 3rd was more important than the result ranked 2nd for this query’), then the global implications can be applied, taking these relative considerations into account. Joachims et al. (2007) used eye-tracking surveys and query reformulations as well as implicit clickthrough data to show that this relative interpretation of the data is useful within single queries and within query chains (or sets of queries) for ordering results. The main focus is on finding suitable relative measures and ways of applying them on the global document collections.
2.7 Topic-focused Web Crawlers

Instead of attempting to index the entire Web graph, the aim of a topic-focused Web crawler is to index the subset of pages in the Web graph which are relevant to a user specified topic. Topic-focused Web crawlers traverse the Web by exploiting its link structure like generic web crawlers, discussed previously. The first topic-focused Web crawler found to appear in the literature is FishSearch (De Bra & Post, 1994) and was proposed more as an automatic browsing tool, rather than the current embodiment of Web crawlers found today as index generators.

De Bra and Post set out the ‘school of fish’ metaphor which describes the motivation behind Web crawlers, and the general rules they would follow. They liken their browsing algorithm to a school of fish breeding and searching for food.

‘A school of fish generally moves in the direction of food. How long the fish live and how many children they produce depends on the amount of food they find. When fish deviate from the school they may survive if they find food. Their offspring will form a new school. However, if they do not find food they die. Fish who enter polluted water also produce few and weak children and die quickly.’

In this metaphor, the fish correspond to the URLs in the Web graph that the crawler follows. Food would be a measure of relevance to the topic. If the URLs lead to relevant documents, then the particular crawl will thrive and more URLs will be followed along this path. If the information this path leads to is not relevant, then the path will be shortened or stopped completely. The metaphor also adds connectivity: ‘polluted waters’ corresponds to sites which do not respond to requests fast enough. These sites should also be cast from the path.

This metaphor sets out the motivation of topic-focused Web crawling quite well. Overall, what it tries to do is follow the right path through the entire Web graph to be able to collect the ‘healthiest’ information, the most relevant to the topic, while travelling through as little irrelevant information as possible.

A focused Web crawler will traverse the link structure of the Web the same way a Web crawler would. However, at each page it makes a relevance decision based on the
content of the page, and based on this decision the page is placed in or excluded from the index. Depending on the crawling strategy implemented, links can be selected from a document (relevant or irrelevant) or neglected depending on the expected outcome. This way, a focused Web crawler will be trying to optimise the results, while spending resources on promising links.

Figure 2.13 gives a useful example to illustrate the difference between focused crawls and standard unfocused crawls (Diligenti et al., 2000). Fig. 2.13(a) is an example of a standard web crawler. Generally it will just follow a breadth first exploration of the Web graph. As pages are discovered, the links within them are added to the end of the queue to be crawled. This means that if the crawl starts from some page which is $i$ pages away from a relevant one, all the pages up to $i-1$ steps from the starting point will be collected first. Fig. 2.13(b) shows the ideal topic-focused Web crawl, where the order the pages are visited in is optimised to collect the relevant pages as fast as possible, and to ignore irrelevant paths. If the crawl is started at a page which is $i$ steps from a relevant page, then only a subset of documents up to $i-1$ will be downloaded. If the crawl strategy is optimal, only $i$ steps will be taken. Methods to achieve this will be covered in the rest of this Chapter.

**Figure 2.13:** Breadth First Vs. Optimised Crawl Paths
2.7.1 General Structure of a Topic-focused Web Crawler

The organisation of a topic-focused Web crawler is much the same as a standard Web crawler, with the addition of the page classifier, and some method of ranking the ‘to-do’ queue, often referred to as the frontier. These additions to the processes are shown in Figure 2.14.

![Figure 2.14: The Topic-focused Web Crawler Processes](image)

The dashed arrow linking ‘classification’ to ‘Get Links’ is included as the ‘Get Links’ process can be included for irrelevant pages, depending on the crawl strategy. Irrelevant pages are kept out of the resultant Index however, as including them would defeat the objective of the system as a whole.

Figure 2.14 shows that the focused crawler employs only two processes further to those of a standard crawler (see Fig. 2.8). The classification process allows the system to collect information on the relevant pages only, and the frontier ranking process means that the crawl is optimised towards finding the best resources quicker. The methods for achieving these processes will be discussed later in this Section.

2.7.2 Achieving a Topic-focused Web Crawl

Having analysed the workings of a Web crawler in Section 2.2, in this section we discuss the two extra processes and methods specific to topic-focused Web crawlers.
2.7.2.1 Topic Representation

One major factor that defines how to perform the page classification and ranking tasks set before any topic-focused Web crawler is how the user represents their information requirements to the system. The representation used has a big impact on the way the system classifies the information it encounters.

Methods described in Section 2.5 (i.e. keyword/phrase based methods) are applicable again in this capacity (De Bra and Post, 1994). However, they bring with them similar weaknesses (see examples in Section 2.5). As topic specific index creation is creating a more persistent set of results than a Web search session through a search engine, topic descriptions that tend to provide more precision, but are more computationally expensive and time-consuming are employed. This is generally achieved through providing a set of example documents to the system, from which it generates an in-depth model of the desired topics (see Section 2.7.2.2 for example methods). This method is the most popular one amongst the literature on topic-focused Web crawling (Chakrabarti et al 1999, 2002, Barbosa et al. 2005, 2007, Tang et al. 2005, Dilligenti et al. 2000). Being able to analyse the entire content of a set of documents, and find re-occurring words or phrases from within a domain means the system can make more informed decisions about whether a document should be included in the results. This is easier for the user than supplying a long list of keywords to describe the area in as much depth as possible. The use of these sort of pattern matching systems also means how the words are used in the document can play a part in the classification decisions – information difficult to supply through keywords queries.

2.7.2.2 Page Classification

The first of the two processes added to the general Web crawler to be discussed will be the page classification process. This is the part of the system that ‘decides’ whether, given the user’s topic description, the page under consideration should be included in the collection or rejected. Given a document and a query (i.e. topic), the result of the page classification will generally be some ‘relevance score’ which will measure the degree with which the system estimates that the page will be relevant to the user. This can often take the form of a term frequency, or a computed probability. Some methods to achieve this are discussed here.
As mentioned in Section 2.5, the topic representation can take many forms. The first method used in this area was simple string matching (De Bra and Post, 1994) where if the current page included a certain keyword supplied by the user, it was considered relevant and included in the results returned to the user. De Bra and Post allowed the user to supply more than one keyword, but assumed no semantic relationship between them. This allowed the relevance score to be the relative frequency of the terms within the document. While not assuming any semantic relationship between multiple keywords, bonuses were given to pages where they did occur together, as this meant that if there was a semantic relationship, this page would have been definitely important. Shark-search (Hersovici et al., 1998) extends this idea, incorporating a Vector Space method for computing a page's closeness to a topic through keywords. This allows for more generalisation and a more fuzzy notion of relevance, rather than Fish-search’s present/not present or /relevant/not relevant – providing a measure of relevance between 0 and 1.

This method’s main advantage is that it is easy for the user to supply the topic representation, and it is easy to test for relevance. The main problem with it, as previously discussed (see Section 2.4.3) is that the limited number of keywords provided by the user is generally not adequate to describe the specific information needs. Also, keywords leave the retrieval process open to the problems of polysemous words (see Section 2.5), meaning that they may introduce noise into any document collection created.

Machine learning methods have also been applied to ‘learn’ what the user is looking for (Chakrabarti et al. 1999a; Chakrabarti et al. 1999b; Pant and Srinivasan, 2005). This generally involves the ‘example pages’. The user will supply example pages relevant to the topic within which their information needs lie (and also some examples of pages not relevant to the topic - though these can be automatically retrieved in many cases (Chakrabarti et al. 1999a)), and the system trains some machine learning algorithm to classify other, previously unseen documents. The fact that in this domain we are looking for relevance or irrelevance, classifiers generally only deal with the binary case (+/-). There are many machine learning algorithms which could be implemented to achieve this goal. Some of the more popular ones (Naive Bayesian, Support Vector Machines and Neural Networks) will be discussed here, and the merits of each in this application analysed.
When supplying the data to a system in order to apply some machine learning algorithm, we need to represent it as some feature vector \( x \). This will generally be some representation of the features of the document, chosen when designing the implementation. For instance, the document vectors in the term-document matrix of Section 2.5 could be used as input to a system like the ones discussed here. Features used vary, depending on the application and the desired output. For example, the words from a document may be used as features, or the terms – they may be stemmed to allow for generalization etc.

There are two phases to any machine learning application: the training phase, where the model with which we can determine one class from the other is created from some training data, and the application phase, where the model is used to classify previously unseen data (see Fig. 2.15).

Figure 2.15 shows the general pattern followed by machine learning implementations. The training phase uses a set of examples (of all classes), of which the labels are already known and a model is produced to make distinctions between the classes from the data. This is known as supervised learning (where pre-labelled data is used for training).

The first classification method discussed here is the Naïve Bayesian classification (Maron, 1961) which is based on Bayes formula. It assumes that given a class, the data is produced from a probability distribution (Pant and Srinivasan, 2005). The posterior probability of \( \Pr(+|x) \) can be represented as:

\[
\Pr(+|x) = \frac{p(x|+) \cdot \Pr(+)}{p(x)}
\]  
(Eq. 2.13)

where \( \Pr(+) \) is the prior probability, generally based on the frequency of the + class in the training data, and \( p(x) \) is a probability density function where
\[ p(x) = p(x | +) + p(x | -) \]  

(Eq. 2.14)

\( p(x|+) \) is also a probability density function, and (as \( p(x) \) is mainly a normalization factor) is where the focus of training a Bayesian classifier lies: estimating \( p(x_j|+) \) and \( p(x_j|-) \) for all \( j \) (Pant & Srinivasan, 2005). The resultant expression allows the system to estimate the probability that a given object (suitably represented) belongs to one class or the other, choosing the highest probability as the likely class label.

Support Vector Machines (SVM) (Cortes and Vapnik, 1995) depend upon the maximum margin solution. The aim of training within SVM is to find the position of a hyperplane within the feature space which separates the classes, giving the maximum margin between them. SVM classifications take the form

\[ y(x) = w^T \Phi(x) + b \]  

(Eq. 2.15)

where \( \Phi(x) \) is a feature space transformation, \( b \) is a bias parameter and \( w \) is a weight vector (Bishop, 2006). Using this function, an input vector \( x \) can be classified as + or – according to the sign of \( y(x) \). The training phase consists of finding values for \( w \) and \( b \) for which the plane \( y=0 \) separates the training data (see Fig. 2.16).

Figure 2.16: Decision boundary margin (Bishop, 2006)

Figure 2.17: Increasing the margin (Bishop, 2006)
A margin (in this sort of model) is defined as the ‘perpendicular distance between the decision boundary and the closest of the data points’ (Bishop, 2006). The decision boundary chosen ($y=0$) should be the one that maximises this margin, as this allows the model to make generalizations with the smallest error. In Figure 2.16, the boundary here can be moved towards the $y=-1$ line, so that the margin is increased, and the generalization error decreased (see Fig 2.17). Data values called the support vectors are the main factors in deciding where the decision boundary should go; examples have been circled in Figure 2.17.

Neural Networks (NN) are another classification method employed in this field. Neural networks were inspired by biological neural networks in the brain (McCulloch & Pitts, 1943). They can be employed for many different kinds of pattern recognition implementations. Neural networks are made up of layers of artificial neurons, which accept a weighted input which it sums together ($x$), and the output generated using a sigmoid function of the form. The output of the nodes in one layer form the inputs to nodes in the next layer, or the output of the entire network. Learning within a neural network again uses a set of training instances, and their target labels. The weights between the nodes are adapted to give the best error reduction, generally using a method called back propagation (Rumelhart et al., 1986) which updates the weights between nodes in the network, based on the error of the output against the target. In the case of text classification as relevant or irrelevant, the input would be a vector representation of the page (features dependent on design) and the output would be the probabilities of the membership of specified classes.

Pant and Srinivasan (2005) performed a comparison of classifiers for Web pages within a topic focused Web crawler (prioritising visitations), implemented using the Naïve Bayesian, Support Vector Machine and Neural Network approaches. They found that the SVM and NN methods provided similar results in terms of classification precision and overall crawl efficiency (see section 2.7.2.3 for frontier ranking). The Naïve Bayesian classifier performed significantly worse than the other two methods however. Their investigation found that this classifier tended to give relevance probabilities very close to either 0 or 1, not utilising the space between meaning that the potential for generalisation was not being utilised. This was attributed to the high dimensionality of the Web
classification problem (high number of terms and documents) which tends to skew results within this particular classification method (Chakrabarti et al., 2002).

2.7.2.3 Frontier Ranking

The second process added to the topic-focused Web crawler is the frontier ranking (see Fig. 2.14). This module serves to optimise the crawl towards pages expected to be relevant or important to the user. As discussed in Chapter 1, the WWWs structure is such that without knowing a page’s URL, or a page that links to it there is no way of accessing it, and that the size of the Web makes the chances of finding a page relevant to a user’s needs through following links almost impossible. This process, however, is exactly what topic-focused Web crawling tries to achieve, only automatically and the difficulties posed become evident if we look at an example Web graph (Fig. 2.18).

![Figure 2.18: Example Web Graph](image)

The problem posed is if, for instance P$_2$ is a page relevant to the topic being crawled for, then the only way to get to it from P$_1$ is to go through the other, possibly irrelevant pages. The frontier ranking portion of the crawler aims towards prioritising visitations to cut down the number of these irrelevant pages which are visited and merely discarded. Figure 2.13 shows the difference between the breadth-first and ideal topic-focused Web crawl. The breadth-first crawl would visit each page in this entire portion of the Web to find just two relevant pages, showing that it is not appropriate for the topic-focused task. The ideal crawl would download only enough sites to lead to the relevant ones, meaning it is perfectly optimised towards locating the relevant resources. As shown in Figures 2.13 and 2.18, the relevant sources are not always the next page along - sometimes a crawler has to visit some irrelevant web pages to find a relevant one. This is
one of the problems facing the design of the crawling strategy which will be discussed in this section.

One of the first clues we have towards whether a page is on-topic or not before downloading it is whether the page(s) that link to it is (are) relevant or not. This method was used by the original topic-focused Web crawler, FishSearch (De Bra and Post 1994). Basically, if a source page contained the keywords driving the crawl, then the links within the page were put to the front of the queue. This means that the pages linked to by pages relevant to the topic were assumed more likely to be on topic themselves. This assumption makes sense in that pages on certain topics are likely to reference material related to the topic either loosely or directly. However, assuming all the citations in the page are relevant can introduce some wasted downloads. For instance, if one of the links was a ‘contact us’ link, the page linked to is likely to contain the contact information of the author, and probably not contain information on the topic itself. Therefore the resultant page should be judged irrelevant and discarded. This is why many approaches look at links and their contexts (Pant & Menczer, 2003; ) in order to rank the pages linked to individually based on their own merits, not as a group based on the host page’s.

The graph-based ranking methods (often used to rank results from search engines – see Section 2.6.2) have been implemented to drive topic-focused Web crawlers (Cho et al. 1998, Chakrabarti et al. 1998). This time however, the ranking is based on computing the importance scores of pages within a local sub-graph of the Web, rather than across the (assumed) entire Web. We can tell how popular a website is within a local Web graph using the link structure seen so far, and therefore visit the pages which are perceived to be more popular within the current Web community.

The popular sites within this partial snapshot of the Web graph are not necessarily the ‘best’ ones to download. Authority within a Web graph does not always mean ‘closest to topic’ or ‘best resource on a topic’. Amento et al. (2000), for instance, surveyed three popularity measures and found that they performed just as well as a simple content-based approach. Chakrabarti et al (1998) combined the HITS graph-based authority measure (see Section 2.6.2.1) with the relevance of the links in the source pages using a weighting scheme, in order to add topic information to the authority-based ranking scheme. They added an analysis of the text within and around the link on the page, finding the frequency
of the driving keyword: if the link contained the keyword or was embedded in text containing the keyword (and therefore assumed to be about the topic) the link would carry more weight within the ranking scheme, and therefore the page should be considered authoritative not only within the sub-graph, but within the topic.

Figure 2.19 shows some links taken from a site on cycling. If, for example, the crawler detailed above which weights links according to their context came across these and the keyword supplied to represent the target topic was ‘cycling’, then one can see how these links would carry more weight. But links which do not occur in the right context could be singled out as low-importance (see Fig. 2.20)

Figure 2.19: Example links and context (http://www.bikelane.com)

As always, please continue to help me out by submitting new links. I'm looking to keep all the sections fresh with new additions. Click here to submit a new link.

Figure 2.20: Example of a link in a bad context (http://www.bikelane.com)

With the text and link in Figure 2.20, there is no mention of the keyword, and therefore it would be assigned a low priority. This would appear to be a good decision as the page one would expect to be at the end of this link would be a ‘contact us’ form, not information about cycling.

There are however, problems with this approach. Taken from the same site, Figure 2.21 shows a problem with using keyword driven context analysis. This section of links is about bicycle component manufacturers, so at the end of these links one would expect to find a lot of information on that topic. However, the system detailed above may look over these links as unimportant as the keyword ‘cycling’ does not appear within the links or their contexts. The word ‘bike’ does appear however, but without the crawler being
supplied with background knowledge of this field, the link between the words ‘bike’ and ‘cycling’ would not be achieved.

Figure 2.21: Problems with context analysis (http://www.bikelane.com)

Machine learning techniques have also been used to estimate the importance of unseen resources (Diligenti et al. 2000, Chakrabarti et al. 2002, Barbosa and Friere 2005 & 2007). The features used differ from implementation to implementation and have ranged from those detailed here (the surrounding Web graph, and text seen around the link) to things like the number of sounds or images in a document. Machine learning classifiers allow the designer to tailor their implementation towards features often seen in one particular field, without the need to create their own rules regarding them. The user can supply the system with their example (seed) links, and the classifier can learn automatically to recognise ‘good’ links from ‘bad’ links, within certain margins of error. Classifiers have also been trained to recognise ‘good’ links during the crawl, not only from the seed documents provided. Chakrabarti et al. (2002) use a semi-supervised approach where the page classifier provides automatically labelled training data to the ‘apprentice’ link classifier. Once it was trained to a certain degree, this classifier was used to classify links, and predict page relevance before downloading it. The features it trained on was the text in and around the link, with importance assigned to tokens based on their proximity to the link.

Context graphs (Diligenti et al., 2000) are another proposed solution to this problem. While using positive examples to generate a model for the desired topic, ‘backwards crawling’ techniques are used to find multiple layers of pages in the surrounding Web graph, building a model of pages which link to the other levels. These links are found by using a third party search engine, with its pre-computed index. Search engines such as Yahoo and Google supply a facility where a user can query for pages that
link to a target page. Using many of these queries, a context graph model can be created (Fig. 2.22).

![Context Graph Model](image)

**Figure 2.22:** Creation of the context graph (Diligenti et al., 2000)

An SVM model is then generated for each level, in an attempt to estimate the link distance of pages found during the crawl from a target page, concerning the target topic. One major drawback with this sort of system is the scale of the training operation. The time taken for this sort of backwards crawling operation increases exponentially with the number of starting pages and the desired number of levels. While this sort of categorisation of surrounding pages is helpful in finding relevant pages in the Web, finding a cheaper and more efficient method of model generation would be advantageous. In addition machine learning algorithms typically rely on large sets of training data, which may not be available for each level or for each topic.

### 2.8 Summary

The Web poses an information management challenge and topic-focused Web crawling offers a solution to some of the problems faced. The initial and most obvious application is the creation of high-precision indexes of topic-specific online documents for searching within certain topic domains. The main motivation behind creating focused indexes rather than general ones is the ease with which they can be kept updated. The number of
documents that have to be refreshed periodically is small. Also, the precision of the results would be expected to increase over the general indexes as the scope of queries it will need to satisfy will be smaller. The popularity of modern search engines however, suggests that this possible implementation would not be a widespread application.

More realistic is the hybridisation of Web crawlers and topic focused Web crawlers as suggested by Chakrabarti (2003). Here it is suggested that topic focused Web crawlers can be used to update the areas of general indexes concerned with information of importance at that time. For instance, while a general crawl will try to update the snapshot of the entire Web (which will take a lot of time), a focused crawl can be running in parallel keeping certain areas updated, so that the best most up to date information on likely, topical subjects is available straight away to users.

Another application is the creation of indexes as part of larger information systems. Pant and Menczer (2003) used topic focused Web crawlers to create indexes of ‘business entities’ and using it for companies to locate businesses related to their areas to find the competition and possible partners or acquisitions. Pant et al. (2004) also used topic focused crawling to extend existing digital libraries, based on the documents already included in them and to discover Web communities (groups of heavily inter-linked websites) that may be of interest to users of the library. Similar applications could be thought of in most topic areas where precision of the index is important.

There are two major challenges facing topic-focused Web crawling: the prioritisation of links to follow and the classification of visited pages. Approaches to both of these challenges have been covered in this chapter. Methods have been refined since the earliest approaches using simple keyword queries to define the topic (De Bra and Post, 1994; McBryan, 1994) to working with more comprehensive descriptions through machine learning to address both of the challenges – page classification (Chakrabarti et al., 1999a, 1999b, 2002; Diligenti et al., 2000, Liu et al., 2004; Pant and Srinivasan, 2005) and link prioritisation (Diligenti et al. 2000). The main benefit of ML methods in text classification is the ease and speed with which the models are generated compared to other more manually engineered solutions (Feldman and Sanger, 2007) despite time often needed to prepare the training data.
Machine Learning methods typically require a large amount of example documents to form a suitable model for classification. As discussed earlier, Web Directories (see Section 2.3) can be one source of training data (Chakrabarti et al. 1999a; Chakrabarti et al. 1999b; Pant and Srinivasan, 2005). The documents found in these directories are organised in a topic taxonomy and are given manually assigned topic labels. This allows them to be used as labelled examples in the training of machine learning implementations. Due to the large number of documents required, more general topics towards the top of the taxonomy are usually used, concatenating the topics below to form one large training set. This means that using the information in the taxonomy in this way is only applicable when creating very general document collections. When more precise topics (further down the taxonomy) are required then there may not be the amount of training data available to create a precise enough model as there are fewer (or no) levels below the target one to concatenate.

For reasons pointed out in Section 2.7, prioritising the crawl towards pages more likely to be classified as on-topic is imperative to making topic-focused Web crawling efficient. However, obtaining training data to train a classifier can be a costly task in itself. One example used in Chapter 2 (Diligenti et al., 2000) used backwards crawling to find the ‘types’ of pages which usually link to target pages for several layers. The training step for this backwards crawling approach is computationally expensive, as it sends many queries to an external system and downloads may documents for each level. The expense of this task grows exponentially with the number of layers the user wishes to consult.

Using the same training data (i.e. publicly available Web directories) we aim to address the problem of describing fine-grained, detailed topics for topic-focused Web crawling. The target topic heading will “describe” the topic requested by the user, but the levels in the taxonomy that lie above it will also be utilised to provide ‘background knowledge’ for the crawler, in an attempt to improve the link prioritisation. As the topic is fine-grained, the proportion of pages found which will pertain to it will be much smaller, meaning that the prioritisation of the links is even more imperative. Using the ‘background knowledge’ supplied by the taxonomy we will perform a ‘staged’ prioritisation of links found according to how close they are judged to be to the target topic within the taxonomy (i.e. many parent headings they are above the target in the tree chosen
to describe the topic). This means that when no target links are available, the crawler should be able to stay at least *close* to the topic, where chances of finding better links should be higher.
Chapter 3 – Topic Profiling and Link Prioritisation

This chapter sets out the methods (summarised in Fig. 3.1) with which we will address the task set out in Section 2.8, namely using fine-grained topics represented within a taxonomy (from publicly available Web directories) for topic-focused Web crawling.

Figure 3.1: Summary of Overall Approach
Fig. 3.1 gives a summary of the approach laid out in this chapter. It shows the inputs to the system – the topic, described using branches from a topic taxonomy (i.e. ‘Sports/Cycling/BMX’ as taken from the example taxonomy in Fig. 2.5) where the target (i.e. most specific) level (BMX) is level 1, the level above it (Cycling) is level 2 and so on. Lexical profiles of documents associated with each taxonomy level are created and compiled to form a model for each level in the topic’s taxonomical representation, which is in turn used to classify pages and prioritise links found during the crawl. This leads to the output of the system, the Index of labelled pages. Lexical profiles are generated by Automatic Term Recognition (ATR) applied both to example documents and documents considered during the crawl.

Section 3.1 explains the translation of the target topic in the taxonomy (i.e. the representation of the user’s information need) into a model used in the prioritisation of links which is covered in sections 3.2. We first explain the document and model representation used in our method.

3.1 Document and Model Representation

There are many possible features that can be used to describe content found on the World Wide Web (see Section 2.2). Pages on the Web are rich in many different forms of data and information incorporating text, images, sounds, video clips as well as other formats. Given this wealth of possible features to describe pages on the Web, it is important to set out which ones are used to support topic-focused Web crawling. This section justifies the choices made. It also aims to set out how models will be constructed to classify pages found using those features.

3.1.1 Document Features

The focus of this research is on the textual features found within pages on the World Wide Web. There are a number of ways that text can be represented (as discussed in Section 2.2.1). Firstly there is the ‘bag of words’ approach where the text is represented as an unordered set of words. This simplifying assumption reduces the dependency of the individual tokens on each other by ignoring word order and grammar, making it a suitable representation for statistical analysis (Lewis, 1998).
Terms are another possible representation of a document. According to Feldman and Sanger (2007) term-level features are made up of one or more words found within the document and are meant to be ‘generally representative’ of its content. Term features allow a smaller but more semantically richer representation than the word-level feature set.

The methods presented here are based on a term representation of documents. This is because the number of documents used for training will be small, so we hypothesise this semantically rich representation will be best suited to describe the topic. Terms in a document can be recognised using an automatic term recognition procedure (see Section 2.2.1). The terms will be expanded through lexical profiling (see Section 3.1.2) to allow generalisation over them and to expand this model a little further.

### 3.1.2 Modelling the Information Need

To generate the model of the user’s information requirements, we intend to utilize the wealth of information stored in existing Web directories, much like many of the previous Machine Learning approaches have attempted to do. However, as already discussed when covering the motivation for this work, we intend to use less general topics, which lie further down the taxonomy. Therefore, the query representation from the user will be the taxonomical representation of their desired topic, for instance:

‘Arts/Movies/Awards/Academy-Awards’

This representation of the target topic ‘Academy Awards’ within Web directories (such as the ODP) will lead to pages which relate to that topic. This will allow the retrieval of these example documents, and their incorporation in a model of the kind of pages the user might be looking for. This will be accomplished by forming term lists from the documents extracted from the taxonomy and concatenating them to form a term list which should describe the kinds of content expected in the target documents.

For each term found within the example documents, a lexical profile will be generated. A lexical profile is defined as all the possible linear combinations of word-level substrings present within the term (Nenadic and Ananiadou, 2006), including all (single) words the term consists of, and all other word-level substrings. It is these lexical profiles which will form the basis of the topic model. Examples of lexical profiles can be found in Table 3.1.
Using lexical profiles will allow for partial multi-word term matches and therefore allows the model to generalise over the vocabulary encountered during the crawl. As the training example (seed) that will be available from the ODP will be of a limited size, the maximum amount of information should be extracted from the available terms.

The term extraction and lexical profiling steps will be repeated in the levels above the target to include some element of ‘background knowledge’ of the target topic. The overall model of the topic, therefore, involves separate term lists for the target topic (level 1) and its parent nodes within the taxonomy, along with each individual term’s lexical profile. This will allow for a link prioritisation strategy that will attempt to keep the topic within the general domain (see Section 3.2).

### 3.2 Link Prioritisation

There are a number of ways that a link might yield positive results for the crawl. Firstly, a link might point directly to a page that is classified as ‘on-topic’. However, a link might also point to a page that, although is not about the topic, contains references to pages which are (as demonstrated in Fig. 2.13). Given the size and growth of the World Wide Web, this reasoning can be extended ad infinitum. If a link exists to a page which will be classified as on-topic, then it is right that this link be visited before all others. The aim of
the strategy proposed here is to also try and estimate the long term gains associated with links that are not expected to yield immediate results, like that of Dilligenti et al. (2000), but without using backwards crawling, and therefore an external, pre-compiled index of the training example’s Web graph.

The topic taxonomy used to describe the target topic also contains information about closely related topics. In this approach, the parent nodes in the path to the target topic in the taxonomy are considered ‘related’ as the range of topics covered by sibling nodes can be wide (future studies may investigate how useful they are in this sort of approach). As the taxonomy grows more precise from the levels at the top, it is possible that the Web graph might follow the same pattern. Pages that are concerned with the topics related to the target topic, might stand more of a chance of being within the Web graph of a page concerned with closer, or target topics. Therefore, links that are expected to yield positive results will be visited first, but if the crawl is in the situation where none of those links are available, then links which are expected to yield related pages are prioritised, attempting to keep the crawl close to the target topic and populate the list of links to be followed with more positive page yielding links. This involves using information about the target pages to assess their relatedness to the topic. Features of the page the link is found in and the way the link is represented in it will be used to estimate the relatedness to the topic. More precisely, we will use the topic of the source document, the words used to represent the link (the ‘link content’) and the text surrounding the link (the ‘link context’) to estimate the link’s relatedness. These features and the way they are used are discussed in more detail in the following sections.

The classification methods detailed in this section are based on the similarity of the lexical profiles discussed in Section 3.1. The lexical similarity function (Eq 3.1) uses a weighted Dice-like coefficient to compare two terms, with more weight given to terms with longer nested constituents and additional weight if the terms share a terminological head\footnote{A terminological head of a term is its constituent (typically noun) that associates the main conceptualisation to the term. Other constituents are used for specialisation of the meaning. For example ‘database’ is the terminological head of ‘microarray database’, while ‘microarray’ specialises its meaning.}. The keywords used in this function will be sets of words found in noun-phrases, therefore adding this weight to the Dice coefficient helps give more prominence to phrases concerning the same noun or thing.
where \( h_1 \) and \( h_2 \) are terminological heads of terms \( t_1 \) and \( t_2 \) respectively, and \( P(s) \) refers to a lexical profile of \( s \). Some examples of calculating the LS are given in Table 3.2

<table>
<thead>
<tr>
<th>term(_1)</th>
<th>term(_2)</th>
<th>LS(term(_1),term(_2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>microarray core facility</td>
<td>DNA microarray facility</td>
<td>0.67</td>
</tr>
<tr>
<td>microarray core facility</td>
<td>microarray database</td>
<td>0.11</td>
</tr>
<tr>
<td>DNA microarray facility</td>
<td>machine learning</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Table 3.2:** Examples of lexical similarity measures

Allowing for weighted partial matches like this allows flexibility in the model needed as the number of example terms will be low. The rest of this section will detail how this term similarity measure will be used to prioritise the crawl towards pages more likely to yield pages that are on-topic.

### 3.2.1 Content

The ‘content’ of the link is its anchor text. That is the text that appears within the ‘a’ tags in the HTML code, and the word or phrase that appears (typically) underlined and highlighted when the page is displayed in a browser that, when clicked, will navigate the user to the intended page. Other forms of media can be the content of the link - images for instance; strategies for dealing with these cases are also discussed in this section.

The fact that the portion of text in question has been used to represent the page that it links to implies that it somehow describes the content of that outgoing page. Therefore, utilising this information is useful for predicting how important the page is to the crawl. To assign a score to the content of the link, the anchor text is treated as a single term. The grouping of the text as representative of the link is assumed to imply that, whether there is one word, or multiple words it is one unit of meaningful text. The content of the link is compared to the term lists that make up the model of the topic taxonomy, and the taxonomy level that best fits is assigned to it.
The average Lexical Similarity is used in this instance (Eq. 3.2).

\[
\text{LinkContentScore}(\text{text}, \text{Level}_i) = \frac{\sum_{t \in \text{Level}_i} \text{LS(\text{text}, t)}}{|\text{Level}_i|} \tag{Eq. 3.2}
\]

Here, \(\text{LS(\text{text}, t)}\) is the lexical similarity between the anchor text and term \(t\) that appears in the topic profile of Level\(_i\). These values are then averaged for level Level\(_i\) (\(|\text{Level}_i|\) is the number of terms that have been used to represent |\text{Level}_i|).

The LinkContentLevel is set to \(i\) where \(i\) is the level for which the LinkContentScore was the highest. This therefore gives us the expected taxonomy level of the target page, based on the content of the link.

### 3.2.2 Context

The text surrounding the link, or its context, may also offer some idea as to what content to expect in the target page. The subject being discussed within the page’s content where the reference occurred should also offer some idea of what the target page is about. Extracting terms from this (typically) small section of text may be hard to do automatically as the information is sparse for statistical processing. The approach used here uses the lexical profiling described in Section 3.1 to find all the linear combinations of the words surrounding the text, offering a list of all the possible linear combinations and therefore all the possible “terms” from the context. The size of the context used will be passed to the crawl as some parameter. Experiments to find optimum context lengths will be discussed in Chapter 5. Similarly to the content scorings, these combinations are then compared to each level in the taxonomy model, and the best fit is found as follows:

\[
\text{LinkContextScore}(\text{P(context)}, \text{Level}_i) = \max_{p \in \text{P(context)}} \{ \max_{t \in \text{Level}_i} \{ \text{LS}(p, t) \} \} \tag{Eq. 3.3}
\]

where \(\text{P(context)}\) is the lexical profile of the context of the link, \(p \in \text{P(context)}\) and \(\text{Level}_i\) is the set of terms which describes the \(i\)-th level of the taxonomy. Finding the maximum fit, for each of the candidate terms from the context, within each term list from the model means that should any term (terms within the model, terms outside the model are irrelevant) occur in the context, then its score will be included. The maximum lexical similarity of the terms which do occur is then the context’s LinkContextScore, and the
context is assigned the taxonomy level which has the largest score\textsuperscript{19}. This means that the optimum number of words to be considered as the context will have to be investigated. Given the context length will generally be a short piece of text; this method should not be too expensive computationally.

\textbf{3.2.3 Source Page}

The topic covered in the page the link was found in may also offer some information as to the topic covered in the target page. In Chapter 2, the assumption that pages relevant to the topic were more likely to contain links to other relevant pages was discussed (De Bra and Post, 1994). This assumption is also used in this work, but with the extension that pages which are assigned a topic closely related to the target topic might also contain links to relevant pages. The page feature of the links will be the level assigned to the score of their host page according to Equation 3.4.

For the page under consideration we have a list of terms which have had their lexical profiles derived. The topic model is made up of \( m \) lists of lexical profiles extracted from example documents within the ODP – each representing a level of the taxonomy branch which describes the desired topic. To compare these lists to the given page, the maximum lexical similarity is used (Eq. 3.4).

\[
PageScore_{\max}(\text{Page}, Level_i) = \frac{\sum_{t \in Level_i} \max_{p \in \text{Page}} \{ LS(p, t) \}}{|\text{Page}|}
\]  

(Eq. 3.4)

where \( \text{Page} \) is the list of terms describing the content of the page visited during the crawl and \( t \) is a term from \( Level_i \), belonging to the list of terms describing the \( i \)-th taxonomy level. The level for which the page receives the highest score will be the one judged most fitting, and the page will be classified as being relevant to the topic represented by that level.

The maximum lexical similarity is used as the first step towards normalising the score. If a simple average was taken over the two lists (one representing the current visited page, the other representing the taxonomy level), then levels in the taxonomy with many terms would be penalised when comparing the PageScores associated with them. Hence,

\textsuperscript{19} If two levels have the same score, the lowest level (closest to the target topic) is assigned (i.e. the method is optimistic).
the maximum lexical similarity is used to make sure that the scores are comparable. If the score is 0 for all the levels, then the page is judged not relevant at all (level $n+1$ is assigned in this case, where $n$ is the number of levels within the target taxonomy).

Some way of accounting for usual overlaps in the vocabulary of the different taxonomy levels has to be used, some boundary at which a score becomes significant may be implemented to make sure that decisions settled upon are the right ones. Therefore, the average lexical overlap of each level with all the other levels may be computed. Therefore, not only does the score have to be the largest amongst the levels, but it has to be above that particular level’s ‘boundary score’. This is a step towards accounting for a shared vocabulary within different sub-topics of the same overall topic.

### 3.2.4 Weighting Terms

As discussed in Section 2.2.1, a given term can be assigned different weights to reflect its significance within the collection of documents it was found in. This measure of significance can be used to weight the decisions placed upon the individual link features, meaning that extra weight is given to terms found within link features that match (or partially match) the more significant terms within a given level of the topic model. Equations 3.5-3.7 show the scoring functions for the linkContent, linkContext and PageScore features modified to integrate this importance measure.

$$\text{LinkContentScore}_{C-Value} (\text{text}, \text{Level}_i) = \frac{\sum_{t \in \text{Level}_i} C_t \times LS(\text{text}, t)}{|\text{Level}_i|} \quad \text{(Eq. 3.5)}$$

$$\text{LinkContextScore}_{C-Value} (P(\text{context}), \text{Level}_i) = \max_{t \in \text{Level}_i} \left\{ \max_{p \in \text{Page}} \left\{ C_t \times LS(p, t) \right\} \right\} \quad \text{(Eq. 3.6)}$$

$$\text{PageScore}_{C-Value} (\text{Page}, \text{Level}_i) = \frac{\sum_{t \in \text{Level}_i} \max_{p \in \text{Page}} \left\{ \frac{C_p + C_t}{2} \times LS(p, t) \right\}}{|\text{Page}|} \quad \text{(Eq. 3.7)}$$

In these equations, $C_t$ represents the weight assigned to term $t$ in the given taxonomy level. Similarly, $C_p$ is the weight of term $p$ reflecting its significance within the page $Page$. In our experiments, we have used C-Value to represent terms’ importance (see Section 2.2.1). The mean C-Value score is used in equation 3.7 to reflect the weight of both terms.
together that are being combined in the lexical similarity function. The use of this weighting scheme will be evaluated in Chapter 5.

3.2.5 Combining the Scores

Once the individual features of the link have been analysed and an expected level assigned to each, the decisions should be combined in order to create an overall conclusion of the link’s expected significance. This allows the ordering of the links so that the ones expected to yield the most promising results sooner (to both the resultant document collection and the pool of links) will be visited first. One combination can be a multiplication of the taxonomy levels assigned to the three link’s aspects (content, context and page):

\[
\text{LinkScore} = \text{LinkContextLevel} \times \text{LinkContentLevel} \times \text{PageLevel} \quad \text{(Eq. 3.8)}
\]

where each LinkContextLevel, LinkContentLevel and PageLevel are the levels assigned to the context, content and source page respectively. Unlike the individual level scores, the LinkScore does not yield a classification of the target page to a level within the taxonomy, but merely a unified score by which to prioritise the visitation of pages with lower feature scores. This score aims to supply a metric through which the list of links can be ordered in terms of prospective gain. This way, links where the representation features (linkContextLevel, LinkContentLevel and PageLevel) all appear to refer to the target level in the taxonomy (level 1), they should be followed first. However, where these links are not available, the links with the most features closest to the target topic (i.e. lower feature scores) should be followed to keep the crawl as close to the topic as possible.

Another combination can be a linear combination of the three levels assigned to a link (Eq. 3.9).

\[
\text{LinkScore}_{\text{Weighted}} = (a \times \text{LinkContextLevel}) + (b \times \text{LinkContentLevel}) + (c \times \text{PageLevel}) \quad \text{(Eq. 3.9)}
\]

There is obviously an issue which feature (link context, content or page relevance) offers the most information as to the importance of the destination page. A weighting scheme can be implemented to learn the relative significance of these scores on a training dataset using a genetic algorithm-like optimisation approach (summarised in Fig. 3.2)
where each combination of weights set to a pre-defined precision is used to sort a queue of pre-classified links. The resultant queue is then assessed according to some ‘fitness’ function.

\[ LinkScore = (a \times LinkContextLevel) + (b \times LinkContentLevel) + (c \times PageLevel) \]

\[ Pre-classified \ Link \ List \]

\[ OrderScore = \sum_{k=1}^{n} \frac{DocScore_k}{k} \]

**Figure 3.2: Automatic Weight Assignment**

All combinations of \( a, b \) and \( c \) can be assessed, and each instance follows these criteria:

\[ 1 \times 10^n \leq a \leq 1 \]

\[ 1 \times 10^n \leq b \leq (1-a) \]

\[ c = (1-a-b) \]

where \( n \) is the level of accuracy required. Formula 3.9 is used to sort a queue of pre-classified training links according to the features that were observed during an actual...
crawl. The links are sorted in descending order according to expected relevance. The placement of each link in the queue is then evaluated using the equation

\[
OrderScore = \frac{\sum_{k=1}^{m} DocLevel_k}{k}
\]

(Eq. 3.10)

where \(m\) is the number of links in the queue, \(DocScore_k\) is the level the page linked to by the \(k^{th}\) link in the queue was classified as. The queue with the lowest OrderScore is judged to be the most effective, with the pages classified closest to the target topic towards the top of the queue, and the ones classified further away from the target topic at the bottom. This allows the best combination of weights given a certain precision to be found and then applied for further crawls. Figure 3.2 shows the process followed for each iteration of this algorithm to discover the optimum weights. The results of the application of both the weighted and un-weighted combination of features will be discussed in Chapter 5.

3.3 Summary

This section has set out methods for

1. The production of a term based hierarchical model that represents each level of an existing Web Directory.

2. The lexical comparison (i.e. classification) of documents and link features (link content, context and source page) to these models.

3. Weighting the lexical comparison of documents and link features according to the significance of terms used in the model representation.

4. Combining the link features into a unified score used for prioritisation of the crawl frontier.

It is hypothesised that the prioritisation of the links found during the crawl using this method will allow for a staged ‘to-do’ queue, allowing the crawler to visit pages expected to concern topics related to the target topic, so that the crawler stands more chance of finding links to relevant pages. This would give an efficient crawl, cutting the number of irrelevant pages visited to find the relevant ones. This hypothesis will be tested in Chapter 5.
In allowing this staged classification, using lower levels of the topic taxonomy, the specification of fine-grained topics will be allowed, whereas the focus of previous work has been on more general topics. This is because methods developed using ML techniques require a large number of training documents. This method will allow the use of smaller training sets for model generation – allowing the use of background knowledge in the form of the levels in the taxonomy above the target topic.
Chapter 4 – Design and Implementation

This chapter deals with the design and implementation of the topic-focused Web crawler used to implement the methods discussed in Chapter 3. It will first analyse the main processes that will be followed and then look at each one in finer detail showing how general design issues of topic-focused Web crawlers will be tackled and also how the methods set out in the previous chapter will be implemented.

To place the individual processes into context, Fig. 4.1 shows the overall process that the topic-focused Web crawler will follow. The inputs to the system are training documents (used to create the model), the seed URLs (initial pages for the crawl to visit) and a crawl limit (used to limit the number of pages visited). The output is the set of pages that have been identified as relevant to the given topic.

Figure 4.1: General Topic Focused Web Crawler Processes
Fig. 4.1 shows the inputs to the system and the processes followed by the system to carry out the topic-focused crawl, with the sections which deal with each individual process indicated in the top right hand corner. The discussion of these processes will make up most of this chapter, along with the data structures they will use and solutions to other more general Web crawling problems.

4.1 Topic Model Engineering

The model building process involves taking the example data for each level, and extracting the features needed to compare crawled documents and links to it. In this case, the example data is a list of documents pertaining to each level of the taxonomy leading to the target topic. The features extracted are dependent on the model desired. In this section, different models used in this work are discussed.

As explained in Sections 2.2.1 and 3.1.1, terms can be made up of a single, or multiple words and are meant to be ‘generally representative’ of the content of the document. This model is based on term features extracted from each level from the training documents. The creation of this model largely depends on an Automatic Term Recogniser (ATR). In the experiments presented later, we used TerMine which is based on the C-Value method (see Section 2.2.1). Figure 4.2 shows the process for building the model based on the training data provided by the user. The training information is the URLs which point to documents pertaining to a topic represented by some level in the taxonomy. Each URL is used to download the HTML document it represents, which is subsequently parsed, allowing the plain text to be extracted and written to a file. The ATR is then used to extract the most significant terms from within the text which are then added to the term list used to represent that level.
The result is a term based representation of each taxonomy level, including the target level (Level 1). Each level is represented by a list of terms extracted from documents representative of its intended content, a representation which is used with the methods introduced in Chapter 3.

4.2 Document Retrieval and Parsing

This section deals with the download and parsing processes. This includes connecting to the given URL, downloading the target document and parsing the HTML to extract the desired portions (plain text and links) to support the other tasks. The queue holds the links which are to be processed by the system – the first URL in the queue is retrieved and followed by the download portion of the application. The queues structure and functions will be discussed later in this chapter (Section 4.5).
4.3 Document and Link Classification

This group of processes deals with the classification of a page’s links and the page itself according to their individual properties as discussed in Chapter 3 using the different models created from the training data (Section 4.1).

This section describes the processes used to classify documents and link features according to the term-based model as described in Section 4.1.1.

---

Page Classification

To classify a document visited during a crawl using this method, the first task is to generate the term representation of the document and then compare it to each level of the model as discussed in Section 3.2 (see Fig. 4.5).

Figure 4.4: Document Classification Process Diagram – Term-Based Method

The term features of the page are compared to each level of the model. The level which fits best is assigned to the document.

Link Feature Classification and Score Assignment (Frontier Ranking)

This process handles comparing the features of the links to the different levels of the model, assigning each of the features an expected level, and ultimately combining the feature levels to an overall score used for link prioritisation as described in Section 3.3 (see Fig. 4.6).
The content of the link is compared to each level in the model, assigning the level for which it fits best. The context (the number of words taken as the context is set as a parameter) is split up into all the possible linear combinations to give possible term candidates. These are then compared to each level. The best fitting candidate score is used for each level, and the context is assigned the score for which this best fitting score is highest. The score combination process (4.8) combines all the levels assigned to each of the features (including the page, process 3) to create an overall score used when ranking the links in the queue.
4.4 Abstract Data Types

The Abstract Data Types (ADTs) handle the representation and a lot of the functions performed on the various data used by the system. The specific design of the ADTs used within this project can be found in Appendix 8 in various Unified Modelling Language (UML) format diagrams.

The main data types used were there term and link lists, which together form an abstract representation of the pages visited by the system. Term lists are also used to represent the model used to classify these pages. These lists handled the storage of the terms, and also the terms comparison. A sortable link queue is also needed to store the pages the crawler is to visit and to facilitate the prioritisation aspects discussed in Section 3.2.

4.5 Database Schema

Figure 4.18 shows the schema for the database responsible for storing the results of the crawls – the pages visited and their features. This database structure should allow for detailed analysis of the crawls, looking at which links were followed and why.

![Database Schema Diagram]

**Figure 4.6: Database Schema**
4.6 Implementation Notes

This section covers the realisation of the design already discussed, with some of the more practical considerations that were dealt with.

4.6.1 Development and Execution Environments

The environmental decisions within this project were largely dictated by the system the application was to be deployed on. The research group has the use of several servers based on an 8-core architecture, each utilising 16GB of main memory. One of these servers would be used to run the crawls and gather results much faster than would be possible running it from a home or office PC. These servers are also already equipped with working versions of the external ATR that the application will need to invoke.

Java was chosen as the implementation language as portability is important when development and deployment environments are different. The utilisation of external applications using Java is also well catered for with the ‘Runtime’ object used to run the ATR in the code. The servers are equipped with Java SE Runtime Environment version 1.6 allowing for easy deployment.

One of the servers runs the MySQL database engine. This, along with the easy interface between MySQL and Java, means that it is the obvious environment of choice for the database end of the project.

4.6.2 External Packages

A number of external packages were used to implement the Crawler, covering three areas, HTML document gathering and parsing, term extraction and connectivity to the database. This section covers how and why they were used, and where they are available from.

4.6.2.1 Document Gathering and Parsing

The downloading and parsing of documents was largely handled by the HTMLParser library. HTMLParser is a well established implementation from SourceForge\(^{21}\). It has been chosen because of its nested document representation and filter interfaces that are useful for creating classes to extract any type of desired information. This library does

\(^{21}\) [www.sourceforge.net](http://www.sourceforge.net)
however, leave what to do with objects once found within the HTML down to the individual programmer through methods which require overriding. It will not automatically extract objects from the document, but will send notifications to other methods once tags have been found, meaning that the parser can be tailored to the specific requirement of that particular implementation. This allows it to be versatile and enables the focus to be on the information required, ignoring the irrelevant objects.

4.6.2.2 Term Extraction

Once the page has been parsed (Section 4.7.2.1) then the terms can be extracted from the plain text. This is carried out using an internal implementation of the C-Value Automatic Term Recogniser (ATR) (Nenadic et al., 2004). As already mentioned, this has already been deployed on the server and so was the most available term extraction software available. A demonstration of the ATR is available at the National Centre for Text Mining’s website\(^\text{22}\). The ATR itself requires part-of-speech tagging, for which it uses GENIA Tagger\(^\text{23}\) version 2.0.2.

4.6.2.3 MySQL Connectivity

The package to enable the crawler to connect to the MySQL database on the server was the official JDBC driver for MySQL, Connector/J\(^\text{24}\). Implemented within the Java framework, it too is portable and easily interfaced with.

4.6.3 Performance Discussion

Each process within the topic-focused Web crawler is carried out linearly with only one document being processed at one time. This design decision was made so that the efficiency results would not be skewed by various factors like the responsiveness of the Web hosts visited and the time taken to extract information from the document (affected by document length and length of terms extracted etc.). Performing the processing linearly means that the order the links are visited has to be representative of the state of the queue.

The lengthiest portion of the document processing is the extraction of the terms, which varies greatly depending on the document, with part-of-speech tagging taking the

\(^{22}\) http://www.nactem.ac.uk/software/termine/
\(^{23}\) http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/
\(^{24}\) http://dev.mysql.com/downloads/connector/j/5.0.html
longest time within that process. This is because of the data required by it (like dictionaries etc.) that has to be loaded for each document. A politeness feature was also added which means that between each link visitation, there is a 5 second gap where the crawler idles. This is to ensure that the Web crawler does not overload Web servers with requests, but has an obvious effect on the overall performance of the crawler.

All these performance factors considered, the crawler can take between 1 and 2 hours to complete a crawl of 100 pages. This could be improved dramatically, for actual deployment, largely through multi-threading and parallel processing, but for the requirements of experiments within this project, linear processing is best.

4.7 Summary

This chapter described the design of the topic-focused Web crawler and sets out the processes for

- Retrieval and parsing of documents from the Web;
- Deriving hierarchical topic models based on example documents;
- The use of the models for link prioritisation;
- The use of the models to classify other unlabelled documents,

The main ADTs used to realise these processes have also been laid out in Section 4.4. The output of the system will be the details of the pages and the classifications made on them stored in the database as set out in Section 4.5. Section 4.6 discusses the implementation considerations and performance issues dealt with during actual deployment of the topic-focused Web crawler.
Chapter 5 – Results and Discussion

The aim of the experiments carried out in this research is to test the suitability of the taxonomical lexical model formed using existing, freely available Web directories for use in topic-focused Web crawling as designed in the previous chapters. This chapter is dedicated to the presentation and discussion of results obtained using the methods and variations described throughout this thesis.

This chapter is set out as follows: Section 5.1 will introduce the experiments and experimental setup that will be employed. The un-weighted term comparison results (both polynomial and weighted linear versions) are discussed in Section 5.2. Section 5.3 will discuss the implications of the addition of the significance of the terms being compared (C-Value) into the comparison functions. All of these sections will be summarised and discussed in Section 5.4.

5.1 Experimental Setup

The methods described previously deal with two main functions of topic-focused Web crawling – the classification of pages and prioritisation of the links that target them – using the models generated from existing Web directories. The experiments carried out attempt to ascertain the extent to which these fine-grained models are suitable for these tasks.

The basic methods and variations along the theme (for instance different parameters or the use of weighting as discussed in Chapter 3) will be tested to get a broad spectrum of results and allow the analysis of factors which may affect this particular method and topic-focused Web crawling at large.

The experimental setting will include the following methods:

- **Lexical Similarity Method**
  A lexical model of the Web directory taxonomy is used to classify both pages, and features of the links. The model uses terms identified in documents and links as the main source for lexical comparisons.
• **Term Weighting using C-Value Method**

An extension of the lexical similarity method where the C-Value scores for the terms found in the model and candidate pages are used to weight the similarity according to average importance of the terms.

The ability of these methods to classify documents found during the crawl will be analysed first, in order to provide a basis for the interpretation of further experiments. These experiments will mainly address the prioritisation of the links – how efficient the methods are at accessing the best pages as quickly as possible. For each of the example topics, the following parameters will be varied:

• **Context Length**: This involves using various window sizes for the link context.

• **LinkScore Functions**: Varying the function and parameters used to combine the features of links to generate the overall score.

• **Large Crawls**: Whereas the other experiments are performed on relatively small crawls, this experiment uses the best combination of parameters from the other experiments on a larger crawl to see how the crawler performs in the wider Web graph.

The evaluation metric used throughout the experiments will be the harvest rate achieved, i.e. the ratio of the number of relevant pages (classified as belonging to level 1 in the target topic taxonomy) against the total number of pages downloaded during a particular crawl. This can be plotted as a graph showing the decision made at each point before the visitation of a page, showing the efficiency of the crawl at each stage. The value assigned to the PageLevel feature of the link prioritisation function is used as the classification of the current page in the evaluation stage.

The methods depend largely on various parameters and data supplied to the crawl. Crawls may be completely different depending on the way the topic is represented, the pages the crawl was started from or even from one day to the next down to the size and dynamic nature of the WWW.
Each experiment will be carried out on three different topic areas extracted from the ODP, and for each topic area 5 seed pages (where the crawl starts from) will be used, in five separate crawls. Table 5.1 details how the topics appear in the Open Directory Project and their start pages.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Start Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Science/Biology/</td>
<td><a href="http://evol.nott.ac.uk/cmelun/links.html">http://evol.nott.ac.uk/cmelun/links.html</a></td>
</tr>
<tr>
<td>Bioinformatics/</td>
<td><a href="http://www.solpugid.com/Links.htm">http://www.solpugid.com/Links.htm</a></td>
</tr>
<tr>
<td>Companies</td>
<td><a href="http://bioinformatics.ca/links_directory/">http://bioinformatics.ca/links_directory/</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.bioinformatics.org/">http://www.bioinformatics.org/</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.ebi.ac.uk/">http://www.ebi.ac.uk/</a></td>
</tr>
<tr>
<td>Science and Technology/</td>
<td><a href="http://www.genome.gov/10001740">http://www.genome.gov/10001740</a></td>
</tr>
<tr>
<td>Biotechnology/Genetics/</td>
<td><a href="http://www.intute.ac.uk/cgi-bin/search.pl?term1=hot+topics;&amp;limit=0">http://www.intute.ac.uk/cgi-bin/search.pl?term1=hot+topics;&amp;limit=0</a></td>
</tr>
<tr>
<td>Genetically Modified Food</td>
<td><a href="http://www.newscientist.com/topic/genetics">http://www.newscientist.com/topic/genetics</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.ornl.gov/sci/techresources/Human_Genome/elsi/elsi.shtml">www.ornl.gov/sci/techresources/Human_Genome/elsi/elsi.shtml</a></td>
</tr>
<tr>
<td>Health/</td>
<td><a href="http://www.diseasesconditions.com/">http://www.diseasesconditions.com/</a></td>
</tr>
<tr>
<td>Conditions and Diseases/</td>
<td><a href="http://www.medicinenet.com/diseases_and_conditions/article.htm">www.medicinenet.com/diseases_and_conditions/article.htm</a></td>
</tr>
<tr>
<td>Obesity</td>
<td><a href="http://www.nhs.uk/Conditions/Obesity/Pages/Introduction.aspx">www.nhs.uk/Conditions/Obesity/Pages/Introduction.aspx</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.nih.gov/">http://www.nih.gov/</a></td>
</tr>
</tbody>
</table>

**Table 5.1: Topics and Start Pages**

The topics were chosen to reflect typically difficult situations for a topic-focused Web crawler. Websites to do with companies will be difficult to find as the likelihood of companies linking to each other will be low as they are somewhat a special case. Usually, when an on-topic page is found, it would be expected that it would contain links to other on-topic sites, as they would be related. Companies are an obvious example where this would not be the case as companies would be in competition with each other and therefore would probably not include links to other companies on their website. ‘Bioinformatics Companies’ was chosen as a topic to offer a difficult situation for the crawler and force the prioritisation of links to play an important role. At the end of a productive period in the crawl (i.e. when a company is found), the link queue might not be replenished by the pages.
visited, therefore the prioritisation may be key to finding the links already seen in the pages leading to the on-topic page that could potentially yield similar results.

Similarly, the GM Foods topic was chosen as it is found at the end of the ‘social issues’ arm of the taxonomy, and is likely to be divisive (i.e. pro-GM Foods and anti-GM Foods) and will contain many points of view (i.e. geneticists, farmers, food producers, medical implications etc.). Finally, the Obesity topic was chosen as this health issue has many facets (i.e. complicating factors, related conditions, treatments etc.) and, while quite a fine-grained topic, obesity has a wide spread of information which can be gathered. The start pages were gathered by conducting a search using the Google search engine and keywords taken from the title of the level above the target level in each taxonomy. This was to ensure that the crawler started from a related sub-web, but would still have to search out positive results.

Documents listed underneath each individual topic heading in the ODP are used as examples to create the multi-level model. The details of the training data are shown in tables 5.2-5.4.

<table>
<thead>
<tr>
<th>Taxonomy Level</th>
<th>Documents</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Companies</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Bioinformatics</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Biology</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Science</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.2: Science/Biology/Bioinformatics/Companies Training Information

Table 5.2 shows the number of documents retrieved from the taxonomy for each level for “Bioinformatics Companies” along with the number of terms found within them. The top level of the taxonomy (Science) contained no documents. This is probably due to the possibility that any document that falls within this topic, will be better suited being assigned a sub-topic (i.e. more specific areas - physics, chemistry, biology etc.). However, in the lower levels, few documents have been used to represent each level which is a major part of what will be tested throughout the experiments – the implications of the small training sets and whether these methods can be successful using it.
Table 5.3 presents the data for the ‘Genetically Modified Food’ topic. Being an issue based topic, it may be interesting to analyse the results in terms of which sides of the argument or related topics are included in the documents returned.

<table>
<thead>
<tr>
<th>Taxonomy Level</th>
<th>Documents</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GM Foods</td>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td>2 Genetics</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>3 BioTechnology</td>
<td>4</td>
<td>62</td>
</tr>
<tr>
<td>4 Science and</td>
<td>4</td>
<td>136</td>
</tr>
<tr>
<td>Technology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Issues</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6 Society</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.3: Society/Issues/Science and Technology/Biotechnology/Genetics/Genetically Modified Food Training Information

Again, we can see that the top two levels in the taxonomy are empty - possibly due to their generality, and that the levels below have been assigned a low number of documents. However, a large number of terms have been extracted – the effect this has will be investigated.

Table 5.4 presents the data for the ‘Obesity’ topic. In this taxonomy, one of the lower levels has no representative documents, again probably due to generality. This is one reason this topic was chosen – to see how this gap in the ‘background knowledge’ affects the prioritisation. Does the taxonomy have to be complete to allow for successful incremental prioritisation?

<table>
<thead>
<tr>
<th>Taxonomy Level</th>
<th>Documents</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Obesity</td>
<td>6</td>
<td>69</td>
</tr>
<tr>
<td>2 Nutrition and Metabolism Disorders</td>
<td>6</td>
<td>50</td>
</tr>
<tr>
<td>3 Conditions and Diseases</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4 Health</td>
<td>6</td>
<td>163</td>
</tr>
</tbody>
</table>

Table 5.4: Health/Conditions and Diseases/ Nutrition and Metabolism Disorders/Obesity Training Information
The rest of the configuration of the crawler depends on the individual experiment, discussed in the remainder of this chapter (Sections 5.2-5.3).

As the evaluation of the topic-focused crawler depends on the automatic classifier, there is a need for the evaluation of its accuracy. This gold standard was generated by the author and verified through blind, objective multiple annotation of part of the gold standard. Inter-annotator agreement (IAA) results will be discussed in Section 5.2.1, and the evaluations of the automatic classifier will be presented in sections 5.2.1 and 5.3.1.

5.2 Un-weighted Term Comparison Results

This section covers the presentation and analysis of the results gathered using the un-weighted classification for link features. It uses the un-weighted lexical similarities between term candidates to assign the taxonomic level to the links content, context and source page.

5.2.1 Page Classification Precision

To estimate the precision with which the methods proposed can classify the topic of pages found during the crawl, a random sample of 45 documents from a crawl of 100 pages is selected and visited manually for each of the three topics (see Appendix 9 for lists). The author then decided which level (if any) within the test taxonomy best describes it.

Figure 5.1 shows the percentage of pages where the automatic classifications matched the manual annotation for each topic. A larger experiment was also carried out involving 180 pages (45 from each of the remaining 4 larger crawls) from one topic (Bioinformatics Companies) to validate the findings of the smaller experiments.
The models for each topic can be used to classify documents into multiple classes with a precision hovering around the 50% mark. Including the irrelevant class, there are 4 classes for the Bioinformatics Companies topic and 5 classes for the GM Foods topic, which means that the classifier performs relatively well compared to the results expected if random classifications were made (i.e. 25% or 20%). As the results for the larger experiment over 180 pages were similar to the smaller ones, we maintain that the conclusions that can be drawn from the smaller evaluation sets are justifiable.

In addition, to validate the manual classifications, two objective annotators were asked to classify a small subset of the pages annotated by the author from the 180 pages samples for both the non C-Value weighted (discussed here) and C-Value weighted (see Section 5.3.1 for actual precision outcomes) page classifications. A total of 40 pages (5 from each of the 4 crawls carried out for both methods) were presented to the annotator in two separate sets of 20. The links to the pages were provided and a text box to input the level corresponding to the heading in the taxonomy (1-5) the annotator thought best described the page (which included the ‘irrelevant’ class). Along with each set of 20 documents, 5 random documents from 10 training documents extracted from the ODP
were presented to the annotator without indication of their origin. They were subjected to
the same process of manual classification used to validate the standard and were thought to
be useful in interpreting the annotation agreement (see below). Figure 5.2 summarises the
results of the IAA.

![Figure 5.2: Manual Annotation and Inter-Annotator Agreement Results](image)

Figure 5.2 shows the rates of agreement between the objective annotators’ manual
classifications and the author’s manual classifications (the ‘Non C-Value’, ‘C-Value’ and
‘Overall’ columns). Both the non C-Value weighted and C-Value weighted results are
included to show the lack of skewing. They are combined in the ‘Overall’ column
showing that the annotators agreed with the authors annotations in 55% of cases. The
‘Annotator Agreement’ column shows the proportion with which the annotators
themselves agreed on the level of each of the 50 pages. A similar ratio (45.95%) was
observed between the objective annotators and publicly curetted ODP data (“ODP Gold
Standard”). The ODP Gold Standard was proposed to set a baseline for the annotators
agreement with the author as this conveys the annotators agreement with a collaborative,
publicly available and scrutinised collection. As the agreement between the annotators and
the author is above this level (around 9% higher) the rate of agreement between annotator
and the author is justifiable and therefore largely validates the manual classifications made
in the precision testing results (Sections 5.2.1 and 5.3.1).

When looking at some of the individual classifications made, some interesting trends
emerge that may point out the strengths and potential weaknesses of this method. During
the ‘Bioinformatics Companies’ crawls, some pages were found which would have been
easy to misclassify, but for which the correct classification was made. For instance, a page
detailing the mouse delivery service offered by The Jackson Laboratory\textsuperscript{25} contains a lot of
information on ensuring genetic quality and is part of a company website. It would be
easy to misclassify this page as a bioinformatics company, given the kind of language
used. This page was classified as belonging to the ‘Biology’ section of the taxonomy, in
what we believe is the best place for it. This page could be more accurately described, but
‘Biology’ in this instance, would be a valid generalisation. This sort of distinction is
repeated with the website of a company providing molecular structure databases for use in
Chemistry research. While discussion of database systems with similar applications as
those that may be provided by Bioinformatics companies is present in the page, it was
classified as irrelevant – the best generalisation given the scope of the model. Cases such
as these were repeated throughout the crawl.

One problem that was seen using the Bioinformatics Companies topic, was the
classification of sites dealing with Bioinformatics journals. This sort of classification is
open to interpretation. One repeated case is the bioinformatics journal at Oxford
Journals\textsuperscript{26} - repeatedly classified as a bioinformatics company. While these are companies
or organisations whose primary interests lie in bioinformatics, the topic specified by this
taxonomy would seem to centre more on suppliers of bioinformatics services and software.
It is interesting to note however, that the classifier was able to pick up on the type of
website it is (not merely ‘bioinformatics’, but some sort of bioinformatics organisation)
from the terms extracted – even though it was a somewhat irregular example compared to
the training documents.

Throughout the topics, there are instances where it is evident that the classifier has
gathered some aspects of the vocabulary and topic trends necessary to classify the page
correctly, but has missed other elements. Within the ‘Bioinformatics Companies’ crawls,
these were generally pages discussing some sort of product, usually a software product.
While this was useful when identifying the companies providing software for use in a
bioinformatics capacity, it allowed the classification of pages related to non-bioinformatics
software and therefore companies as relevant pages. Similar examples are available within
the GM Foods crawls. Many pages where classified as relevant that discussed agricultural

\textsuperscript{25} http://jaxmice.jax.org/genetichealth/index.html
\textsuperscript{26} http://bioinformatics.oxfordjournals.org
issues and issues related to eco-systems, but not necessarily containing the required genetics element for them to be correct classifications.

This problem identified one assumption made that may not fit within the method. The assumption was that the vocabulary within the taxonomy would be suitably representative at each level to allow for a distinct model of each. As these sort of examples show, it does not appear to be the case. The levels model certain types of very specific information, while not representing certain essential elements for an adequate topic description. This leads to possibilities for improving this topic descriptions by performing some inference between the levels, allowing conditions where pages assigned to a certain level in the taxonomy must contain elements pertaining to the level (or levels) above it in the taxonomy. This would allow for an explicit requirement of the representation of more general topics within pages classified as belonging to each level.

Note that we will use the page classification model to evaluate if a page is on-topic in the experiments presented in the remainder of this section. We assume that the method makes consistent and evenly distributed errors so that comparisons between different link prioritisation methods can be performed.

5.2.2 Link Prioritisation: context features

The link’s context window is the number of words each side of each link taken to be the link’s context. Variation of this parameter should affect the priority each link has, therefore directly affecting the path the crawl takes and ultimately its efficiency.

For each context window (between 4 and 10 words) the harvest rate is plotted for a crawl of 100 pages starting from the same page allowing a comparison in terms of quality. The baseline for this experiment will be a breadth-first crawl which is started from the same page. It will also classify pages found using the same model, meaning that we can find its harvest rate, allowing for conclusions to be drawn as to the performance increase using prioritisation and in this case, the optimum context window for the links.

The harvest rate shows the ratio of relevant pages (those classified as level 1 in the topic taxonomy) against the total number downloaded. It tracks the efficiency of the crawl at each step it takes. Figures 5.3-5.7 show the results for the context length experiment for the Bioinformatics Companies topic, with each start page.
Figure 5.3 shows the best results for the overall crawl. Achieving around 35 relevant pages from the 100 crawled. This shows the value of good seed pages. This particular page (http://evol.nott.ac.uk/cmelo/link.html) is entitled “The greatest
Bioinformatics Links page in the world...Ever!

and would be expected to be in a lucrative part of the Web graph for this particular topic. Very productive periods are also observed in the length 6 and 4 crawls in Figure 5.6 and also in the context length 5 crawl in Figure 5.7. The context length 6 crawl in Figure 5.6 reaches pages within Wikipedia\(^{27}\) which are largely false positives. However, the context length 4 crawl also has a very productive period towards the end. Here the crawler is traversing various research groups to do with the computational biology department at the Simon Fraser University of Canada. The EBI crawl (Fig. 5.7) with context length 5 finds a sub-web within the U.S Department of Energy website which deals with research groups employing computational biology methods to investigate the environment and eco-systems on a micro-biological scale. These are both very positive results, and show the crawler’s ability to seek out promising pages.

Looking at the most productive crawl, the context length 6 crawl in Figure 5.3, in its early stages (the first 10 pages) the breadth-first outperforms this crawler, finding relevant pages earlier on. More specifically it finds the websites of the European Bioinformatics Institute\(^{28}\) (EBI), GenBank\(^{29}\) and the DNA Data Bank of Japan\(^{30}\) (DDBJ). The reason these pages were found so early on in the breadth first crawl is simply because they appear at the beginning of the seed page. More interesting is why they were not found earlier in the prioritised crawl.

Looking at their presentation in the seed page (Fig. 5.8) the reason they were “overlooked” by the prioritised crawl is that they are represented by their names, or abbreviated names, which in turn means that they are less likely to be part of wide ranging domain vocabulary that have been learned from the example pages. This makes these links difficult for any textual feature based Web crawler. While the link is descriptive of the target page’s content, it takes an understanding of the page at large and its structure to be able to deduce it.

---

\(^{27}\) http://www.wikipedia.org

\(^{28}\) http://www.ebi.ac.uk


\(^{30}\) http://www.ddbj.nig.ac.jp/
After this early run of positive results, the breadth-first crawler largely tails off, finding a range of results, with some successes. Still, many of the prioritised crawls stay constantly below the level that the Breadth-First crawler attains. These are the 7 and 8 results, showing that these windows of text either side of the links cause confusion (i.e. noise), leading the crawler to non-relevant results. One pattern repeated through the crawls and seed pages is that varying the context length can have a large effect on the crawls overall outcome. For instance, in the nott.ac.uk crawls after the context length 7 and 8 crawls which perform below the breadth-first crawl, the context lengths 9 and 10 start producing a better quality of result again, reaching more relevant pages sooner in the crawl. Figure 5.9 shows the overall number of relevant pages gathered by each prioritised crawl, with the breadth-first line added for comparison. It shows the phenomenon explained, where the crawl gets progressively worse the larger the context length becomes, but then starts rising again.

Figure 5.9: Context Length - Overall Result Comparison (Bioinformatics Companies) nott.ac.uk
This pattern might point towards a new feature not previously considered during this project. If, for instance, this was caused by the links’ contexts overlapping, meaning that if one was relevant, its neighbours would stand more chance of being judged relevant, then we might consider the proximity of a link to one that has proved/been classified as relevant as an indicator towards its expected information yield. This seems to prove the case in some examples during the crawl mentioned above. Figure 5.10 shows one such example from the context length 10 crawls.

Figure 5.10: Portion of Visited Pages (visits 23-29)  
Context Length 10 Crawl (Bioinformatics Companies)

Figure 5.10 shows a sample of a particularly productive period during the crawl. The interesting pages in this case are 24, 25 and 28. Pages 25 and 28 are linked to by 24 and looking at the reason why they were followed so closely together in this case highlights the possibility that proximity may be a feature worth considering.

Figure 5.11 shows that the contexts may overlap, with the content term of link 214171 appearing as the first word in the next link’s context. The context score of the second link is then classified as belonging to the same level as the content of the first link with that term therefore had some role in that classification. Figure 5.12 shows the results of the same link analysis carried out using a context length of 6.

<table>
<thead>
<tr>
<th>page_id</th>
<th>link_id</th>
<th>target</th>
<th>score</th>
<th>contentScore</th>
<th>contextScore</th>
<th>pageScore</th>
<th>content</th>
<th>context</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td><a href="http://www.cbs.dtu.dk/biolinks/align2.html">http://www.cbs.dtu.dk/biolinks/align2.html</a></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td><a href="http://evolution.genetics.washington.edu/phylip.html">http://evolution.genetics.washington.edu/phylip.html</a></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td><a href="http://evolution.genetics.washington.edu/phylip.html">http://evolution.genetics.washington.edu/phylip.html</a></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td><a href="http://evolution.genetics.washington.edu/phylip.html">http://evolution.genetics.washington.edu/phylip.html</a></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td><a href="http://evolution.genetics.washington.edu/phylip.html">http://evolution.genetics.washington.edu/phylip.html</a></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Example Links from Context Length 10 Crawl (Bioinformatics Companies)

<table>
<thead>
<tr>
<th>target</th>
<th>score</th>
<th>contentScore</th>
<th>contextScore</th>
<th>pageScore</th>
<th>content</th>
<th>context</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://evolution.genetics.washington.edu/phylip.html">http://evolution.genetics.washington.edu/phylip.html</a></td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>Programs</td>
<td>The package About the Executables About the &lt;code&gt; - compiling it yourself The documentation...</td>
</tr>
<tr>
<td><a href="http://evolution.genetics.washington.edu/phylip.html">http://evolution.genetics.washington.edu/phylip.html</a></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>Source code</td>
<td>A general description of PHYLIP - details in the PHYLIP package About the Executables About the...</td>
</tr>
</tbody>
</table>

Figure 5.11: Example Links from Context Length 10 Crawl (Bioinformatics Companies)

Figure 5.12: Link analysis using Context Length 6 (Bioinformatics Companies)
While the contexts still overlap, it is by a smaller margin, and the terms that caused the classification to be good in the length 10 experiment are obviously not present in this version of context, of length 6. The fact that the second page was not classified at the desired end of the taxonomy shows that this length of context is too inclusive. However, it highlights the possibility of another feature to be used as well as a reasonable length of context, therefore having less prevalence than it may have had in this particular crawl.

Figures 5.13-5.18 show the results obtained using the Genetically Modified (GM) Foods topic.
Overall, these crawls were more productive than the bioinformatics companies crawls previously discussed, with more crawls retrieving between 20 and 40 relevant pages out of the 100 visited. One striking feature of one of these crawls is the very productive time towards the end of the context length 8 crawl in Figure 5.13. Each of these pages is from the http://www.wormbase.org/db/ domain and each was classified as belonging to level 1 in the taxonomy with the same pageScore, suggesting that it was some shared features of the page that caused this classification. Indeed, Figure 5.18 shows the shared header used by each page within the WormBase domain. It contains general page sections, and also the search bar, with options as to where to search. This shared terminology (within the header and the drop down box, within the search section) caused each page visited within the wormbase domain to be classified to the same level, leading to more pages within the domain. Unfortunately the crawler got stuck in this area of the Web like this making these mis-classifications.
This problem raises the question of which content to include in a classification within the World Wide Web. While menus and headers as seen in Fig. 5.18 add a necessary degree of functionality to the page and can offer some insight into what the site’s aim is overall, the insight into the content of that particular page may not be as revealing.

The results for the Obesity topic crawls (Fig. 5.19-5.23) show various context lengths being the optimum value, between 5 and 9.

---

**Figure 5.19:** Harvest Rate Results – Context Length Variation (Obesity – http://www.diseasesconditions.com)

**Figure 5.20:** Harvest Rate Results – Context Length Variation (Obesity – MedicineNet)

**Figure 5.21:** Harvest Rate Results – Context Length Variation (Obesity – http://www.dh.gov)

**Figure 5.22:** Harvest Rate Results – Context Length Variation (Obesity – NHS.uk)
Figure 5.23: Harvest Rate Results – Context Length Variation (Obesity – NIH.gov)

Figure 5.22 and Figure 5.23, the NHS and NIH crawls both show sharp inclines within certain crawls that tend to be the best performing crawls. The context length 8 crawl for the NHS crawl shows a sharp incline. These pages were from a lab test company’s website concerning tests carried out on those deemed at high risk of stroke and coronary heart disease included due to their discussion of risk factors. In the NIH crawl, the context length 9 and 5 manage to find sub-webs within the NIH’s Center for Scientific Review website which details projects and studies carried out in the U.S on obesity and diabetes. These show positive results in the ability of this method to seek out relevant sub webs and exploit the information successfully within pages that link to them.

For each set of crawls, the optimum context length is chosen for the rest of the experiments carried out in this section. The average context length in this set of experiments is around 7 (6.467) across all domains and start pages. Almost the entire range of possibilities has shown to be the best in some cases (between 4 and 9) with no one score being particularly more prevalent than another overall.

5.2.3 Link Prioritisation: combining different features

This section details the results obtained using the variations of the LinkScore function. This allows us to see the particular part each individual feature analysed played in the path the crawler took. In each, the context length is set to the value which performed best in the experiments detailed in the previous section.

31 http://www.labtestsonline.org.uk
The results from the Bioinformatics crawls (Fig. 5.24-5.28) show a range of features performing the best. Overall, combining all three features perform best in the majority of crawls (achieving the best crawl with 3 of the 5 start pages). The other results show that a persistent feature is the Content of the link (‘Content’ performs best in the Solpugid.com crawl shown in Figure 5.26 and best when joined with the ‘Page’ feature in the bioinformatics.org crawls show in Figure 5.25).

**Figure 5.24:** Harvest Rate Results – LinkScore Function Variations (Bioinformatics Companies – nott.ac.uk)

**Figure 5.25:** Harvest Rate Results – LinkScore Function Variations (Bioinformatics Companies – bioinformatics.org)

**Figure 5.26:** Harvest Rate Results – LinkScore Function Variations (Bioinformatics Companies – solpugid.com/Links)

**Figure 5.27:** Harvest Rate Results – LinkScore Function Variations (Bioinformatics Companies – bioinformatics.ca)
Analysis of the ‘Page’ and ‘Content x Context x Page’ crawl in the ‘nott.ac.uk’ start page crawls (Fig. 5.24) shows that while the ‘Page’ crawl, and the ‘Content x Context x Page’ crawl seem to yield similar results in terms of how many relevant pages are returned, the only pages they have in common are the seed pages. This means that using differing features on the same crawl discovers entirely different sub-webs (with similar precision).

The GM Foods results (Fig. 5.29-5.33) show a similar pattern to the ones seen in the Bioinformatics Companies crawls. With three out of five of the start pages, all three features work best together, with the content and page feature being most important in the other two (‘content x page’ in Fig. 5.33 and ‘page’ in Fig. 5.30). In the actionbioscience crawl, the two best performing context length results (Fig. 5.13) are shown due to the irregular results at the end of the context length 8 crawl. This is to put the other results in context, so as to reduce the impact that this anomaly has on the interpretation.
One interesting result is the polarisation of features with the New Scientist crawls (Fig. 5.32). Here, we see the ‘context, content, page’ crawl performing best, with the ‘content, context’ crawl producing a very similar path. All the other results, however, perform very badly. These two crawls find their way to the World Health Organisations website\(^{32}\) - more specifically the ‘food safety’ sub-web\(^{33}\) – which is strongly related to the GM Foods topic. Interestingly, the ‘content’ and ‘context’ crawls using the individual features find their way to this area of the Web, but much later on in the crawl (we start to see the benefits in both just before the 100 page limit is reached), but combining these features in this case allows this sub-web to be exploited much earlier.

\(^{32}\) http://www.who.int

\(^{33}\) http://www.who.int/foodsafety/en/
The results obtained using variations of the functions on the Obesity topic are shown in Figures 5.34-5.40. The MedicineNet crawls were done using context lengths 6, 7 and 9 as they reported very similar results in the context length experiments.

**Figure 5.34:** Harvest Rate – LinkScore Variations (Obesity – diseaseconditions.com)

**Figure 5.35:** Harvest Rate – LinkScore Variations (Obesity – MedicineNet Context Length 6)

**Figure 5.36:** Harvest Rate – LinkScore Variations (Obesity – MedicineNet Context Length 7)

**Figure 5.37:** Harvest Rate – LinkScore Variations (Obesity – MedicineNet Context Length 9)

**Figure 5.38:** Harvest Rate – LinkScore Variations (Obesity – dh.gov)

**Figure 5.39:** Harvest Rate – LinkScore Variations (Obesity – NHS.uk)
Once more, three out of the five sets of crawls result in the combination of all the features together giving the best harvest rate overall. One of which is the diseaseconditions.com crawl – while the ‘page’ crawl appears to perform best thanks to a very productive period towards the end this was caused by misclassifications of a website on Obsessive Compulsive disorder. Due to the ‘page’ score being the only thing in consideration when prioritising the links found within it, this original error was propagated, and many links from within that sub-Web were explored. As the efficiency takes in to account the entire crawl, the ‘Context x Content x Page’ crawl was deemed best as even to the point where the ‘Page’ crawl deviates, the crawl using all the features is performing best. The ‘Context x Page’ crawl performed the best in the NHS.uk crawl and the ‘Content x Page’ crawl performed best in the NIH crawl.

Overall, the combination of all the features is most frequently the best for prioritising links (in about 60% of cases from these experiments). Very rarely is it the case that a single feature of the links is the best (13% of crawls), meaning that more information is definitely better, with the topic of the page it was found in being a persistent feature.

We have also experimented with the impact of different link features on the final score when these features are combined using a linear combination of scores suggested by individual metrics (see Eq. 3.9). These are presented in Section 5.2.5.
5.2.4 Link Prioritisation Results in Larger Crawls

Once the various optimum parameters have been discovered, it is possible to carry out larger crawls to test the crawler’s performance on a more remote Web graph, beyond the relevant neighbourhood of the seed pages. In this section, crawls of 400 pages were carried out for each topic and each start page using the best performance prioritisation function and context length as suggested in section 5.2.2 and 5.2.3. The rest of the experimental setup is the same as set out in 5.1.1, using models generated from the same example documents. A breadth-first crawl was again carried out for evaluation purposes.

Firstly, the results for the Bioinformatics Companies crawls are shown in Figures 5.41-5.45.
While in each case, the prioritised crawl out-performs the breadth-first crawl, there are some interesting trends in many of these results. The crawl starting from the nott.ac.uk page performs very well, collecting almost three times as many relevant pages than the breadth-first crawler. While still performing better than the breadth-first crawls, the other prioritised crawls tend not to fair as well as the nott.ac.uk crawl’s 34% harvest rate. The fact that both crawls suffer implies that the problem is the start pages that were chosen. Despite this, each crawler finds the links within the pages that guide them towards the relevant sub-webs (all be them small). Interestingly the solpugid.com crawl (Fig. 5.43) shows the breadth-first crawler ‘catching up’ with the prioritised one later one. The productive period between the 150th visitation and the 310th of the breadth-first crawl that sees it meet the prioritised line again is caused by the visitation of the same sub-web on a biodiversity conference and its sponsors – the prioritised crawl having found the positive pages much faster. The EBI crawl (Fig. 5.45) also has an interesting trend in that it seems to follow the same pattern as seen in the previous crawls for the context length and linkScore variation crawls, but hitting the same relevant areas much later (after around 150 pages rather than around 50). This is due to changes in the pages it was visiting between the crawls.

The results from the GM Foods crawls (Fig. 5.46-5.50) show a mixed picture. Most prioritised crawls achieve results much better than the breadth-first one. However, the genome.gov (Fig. 5.48) and ORNL (Fig. 5.50) crawls show the breadth-first crawl achieving much the same efficiency. This may be due to having a very highly linked start page, where the majority of the links lead to relevant pages. Interestingly, these four crawls don’t deviate too much from the site they are started in, i.e. most of the pages they visited come from within the domain in which the start page lies. The other crawls show up to 6 times as many relevant pages found. The actionbioscience.com crawl was carried out with both context length 4 and 8 as these context lengths performed similarly before the mis-classification within the context length 8 crawl. On the larger crawl, after the 100 page limit of the previous crawl, the context length of 4 finds many more valuable sub-Webs performing much better than the context length 8 crawl.
The ‘Obesity’ crawl results (Fig. 5.18) show similar results with just over twice as many relevant pages found by the prioritised crawl. This crawl managed to gain a lot of relevant pages after finding its way in to the ‘obesityfocused’ domain through link prioritisation, a website which has a lot of articles containing information on obesity and obesity related issues.

The Obesity results (Fig. 5.51-5.55) shows the prioritised crawls out-doing the breadth-first crawls once again, with each finding more relevant pages. The diseaseandconditions.com crawl (Fig. 5.51) performs similarly to the breadth-first crawl
until the last 100 pages or so, where it manages to break away and retrieve many more pages. Similarly, the MedicineNet results show the breadth-first crawl catching up with the prioritised crawl, before the prioritised crawl once again breaks away. The dh.gov crawl (Fig. 5.53) also shows the breadth-first crawl eventually catching up. This shows that the breadth-first crawl will eventually find the relevant pages, but the prioritised crawl has the capacity to seek them out much sooner. The NIH crawl (Fig. 5.55) shows a very productive start – looking at the pages that make up this part of the crawl, many are concerning conditions not associated with Obesity from the NIH sub-web – this is down to the misclassification. The equivalent crawl using C-Value weighted feature comparisons (see Section 5.3.4, Fig. 5.116) the precision of these classifications is increased, and this crawl is, again, very productive.

![Figure 5.51: Larger Crawl Results (Obesity – diseasesandconditions.com)](image1)

![Figure 5.52: Larger Crawl Results (Obesity – MedicineNet)](image2)

![Figure 5.53: Larger Crawl Results (Obesity – dh.gov)](image3)

![Figure 5.54: Larger Crawl Results (Obesity – NHS.uk)](image4)
Overall the prioritised crawl manages to find more relevant pages more efficiently than the breadth-first crawler in 13 out of the 15 larger crawls performed (87%) showing that the method is able to find the more relevant sub-webs with a much greater efficiency. This is after the optimum configuration has been applied.

5.2.5 Link Prioritisation with Linear Combination of Features

This section deals with the results obtained through using the feature weighting scheme detailed in Section 3.2.5. This method uses a brute force style method to find weights for the features based on links already seen for pages already classified. The data used to generate these weights were a random subset of 1200 links visited to obtain the pages in each prioritised ‘large crawl’ (i.e. 400 pages for each topic) – this number was chosen to suitably represent links found within crawls (20% of the entire collection). The values for the features have already been gathered and the classification of the page they link to already obtained.

The weights for Equation 3.9 were generated using all of the links together in one set from across the subject areas, meaning that they were not tailored to one specific area in particular. The weights settled upon using the method outlined in Section 3.2.5 are shown in Table 5.5. The weights assigned imply that a lot less information is obtained from the link’s content than from the context and page it is found in.

<table>
<thead>
<tr>
<th>Content $(a)$</th>
<th>Context $(b)$</th>
<th>PageScore $(c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.062</td>
<td>0.627</td>
<td>0.311</td>
</tr>
</tbody>
</table>

Table 5.5: Weights assigned to link features
This arrangement of weights shows that the PageScore and the ContextScore have the most weight, while the content of the links holds very little weight. While most crawls in the linkScore combinations experiments (see Section 5.2.3) had all three features as the best performing, all but two included the page feature, so it probably follows that it should receive a lot of weight in this function. However, Content was included in the optimum functions more often than the Context feature was – it is therefore surprising that Context should receive 10 times as much weight as the Content. These weights were generated on a lot more data than was used for the LinkScore combinations experiments, however, and therefore different conclusions are acceptable. The weights were used when combining the link features during crawls for each of the target subject areas. To evaluate them, the breadth-first crawls and the most successful prioritised crawls from the preceding sections will be used.

Firstly, the results for the crawls for the Bioinformatics Companies topic are presented in Figures 5.56-5.60.

![Figure 5.56: Link Feature Weight Results (Bioinformatics Companies – nott.ac.uk)](image1)

![Figure 5.57: Link Feature Weight Results (Bioinformatics Companies – Bioinformatics.org)](image2)
These results show the un-weighted polynomial function out-performing the weighted linear function in all cases. While in the bioinformatics.org crawl (Fig. 5.57) the weighted function retrieves more relevant pages overall, the un-weighted crawl retrieves more pages faster, and efficiency is the criteria we are looking at more than anything. The lead the weighted crawl achieves towards the end is only two pages and is negligible. In the bioinformatics.ca crawl (Fig. 5.59) the weighted crawl gains the advantage for about 20 pages (between visits 50 and 70), but this is ultimately not exploited and is lost before the end of the crawl. The weighted nott.ac.uk crawl (Fig. 5.56) performs only just as well as the breadth-first crawl.

The GM Foods results are shown in Figures 5.61-5.65.
Once again, the linear weighted crawls are outdone by the un-weighted polynomial crawls in almost all cases. The linear function retrieves results similar to the polynomial one in the ORNL crawl (Fig. 5.65).

The Obesity topic crawls (Fig. 5.66-5.70) repeat the trend once more. Here, the un-weighted crawl usually outperforms the weighted linear crawl. One exception to this is the dh.gov crawl (Fig. 5.67) which stays ahead of the un-weighted polynomial crawl for
around 30 page visits. In this instance, the weighted linear crawl finds its way into a network of pages in the nhs.uk domain which pertain to ‘commissioning primary care and community service’ which give frameworks for care within the community, dealing with many issues including Obesity. The un-weighted polynomial crawl has visitations within a network of pages to do with Web standards, before retrieving other results from a different sub-Web. During the unproductive period towards the end of the weighted crawl, the crawler visited many pages to do with funding within the NIH, where there is little information to be had.

**Figure 5.66:** Link Feature Weight Results (Obesity – diseaseconditions.com)

**Figure 5.67:** Link Feature Weight Results (Obesity – dh.gov)

**Figure 5.68:** Link Feature Weight Results (Obesity – MedicineNet)

**Figure 5.69:** Link Feature Weight Results (Obesity – NHS.uk)
These experiments show that the weighting procedure proposed is not suitable for weighting this function, with the un-weighted version outperforming in nearly every situation. The exceptions do show some interesting trends, finding areas of the Web that haven’t been visited in previous experiments, and therefore indicating that a better weighting of these functions could improve the crawls dramatically. One major weakness of the method proposed is that the ordering of the links is not incremental. The links are presented as a list to find the optimum weights, where we should be looking not only to find the best pages first, but also the best pages to get links from as well. It may be better to provide links iteratively, depending on the pages at the top of the list, allowing the recreation of an actual crawl rather than just simulating a single state. This would obviously be a much lengthier process, but the results gained may be a lot more promising.

5.2.6 Summary
The crawls based on lexical similarities that are combined using their multiplication have been shown to have an advantage over the un-prioritised crawls in most cases given the correct parameters with only a small number of exceptions. The function for link prioritisation including all the proposed features of the link has fared better in each topic resulting in the best crawl in 60% of cases. The Page feature was the most prominent in the exceptions, included in all the top performing functions apart from one (Bioinformatics companies - solpugid crawl, see Fig. 5.26). The best performing context length has ranged between 4 and 9 words either side of the link, with an average of around 6 words (6.2).
The weighting strategy proposed and implemented, didn’t manage to improve the efficiency of the crawls in most cases. One proposed solution was providing links to the weight assignment algorithm iteratively, rather than as one queue, for the prioritisation and fitness testing. This would allow the weights to convey the features likelihood to find good pages, and pages which stand a better chance of linking to good pages.

5.3 Incorporation of Term Relevance

The purpose of adding importance measures (as C-Value scores) into term comparisons (as introduced in Chapter 3) is to give more weight to the more prevalent terms found in the training documents and the terms in the documents and link features found during the crawl. This section sets out the results gathered using this method and the analysis of the C-Value as a suitable weighting mechanism.

5.3.1 Page Classification Results

Figure 5.71 sets out the precision results using both the C-Value method, and the non c-value method (introduced in section 5.2.1) so that a comparison can be made.
It also includes the results from each of the manual annotation of the larger document set from the Bioinformatics Companies crawls. All these results were gathered as described in Section 5.2.1.

The results in this test were gathered in the same way as in the previous experiment (see Fig. 5.1). A random selection of 45 pages classified during a crawl carried out with the C-Value methods were chosen for each topic, and manually classified. Figure 5.71 shows the classifications that matched the manual level assignments. Overall, it shows a decrease in precision. This may be because using the C-Value makes the models too specific, too focused on the documents used to make them. An in depth look at the terms extracted from pages found during the crawl shows that most are selected with a C-Value of around 2 meaning the information added to the classification is minimal anyway. It appears using the C-Value score in this manner is not suitable for the classification of pages found on the Web.

5.3.2 Link Prioritisation: Context Features

The link prioritisation functions also use a C-Value weighted approach to the lexical similarity which may change the parameters used for the context window and the combination of features used to generate the visitation order. This section deals with the context window length.

In the majority of the crawls carried out on the Bioinformatics Companies topic, the C-Value weighting method has been more successful than the un-weighted feature score assignment functions.
The optimum context lengths range between 4 and 9 once more, with an average of around 7 (6.8) words either side of the links found. In the nott.ac.uk crawl, the un-weighted feature score functions perform about twice as well as the weighted version, but this is the only exception in the 5 crawls. The EBI crawl, for instance, with a context length of 9 performs about 3 times as well as its un-weighted counter-part.

The GM Foods crawl results (Fig. 5.77-5.81) show a different picture regarding the C-Value weighting of the feature classification. Here the non weighted feature classification performs better in 4 out of the 5 crawls, collecting more than twice as many relevant documents in some cases.
The best context lengths in this case range between 6 and 9, with an average length of around 8 (7.8) words either side of the link. Most of the prioritised crawls perform much better than the breadth-first ones, with the exception of the genome.gov set of
crawls, which performed similarly to the crawls seen in the last section collecting about as many relevant pages as the breadth-first version. Even the breadth-first one in this instance performs better than the original un-weighted prioritised crawl. This is likely due to changes in the Web graph and the different page classification method used, incorporating the C-Value weights.

The results for the Obesity topic crawls show 3 out 5 cases where the un-weighted feature classification functions have found more relevant documents than the weighted versions (Fig. 5.82, 5.83 and 5.84), but not generally by very large margins. The cases where the weighted crawls have out-done the un-weighted versions (Fig. 5.85 and 5.86) have found up to twice as many as the un-weighted versions.

Figure 5.82: Harvest Rate – C-Value Context Length Experiment (Obesity – diseasesconditions.com)

Figure 5.83: Harvest Rate – C-Value Context Length Experiment (Obesity – MedicineNet)

Figure 5.84: Harvest Rate – C-Value Context Length Experiment (Obesity – dh.gov)

Figure 5.85: Harvest Rate – C-Value Context Length Experiment (Obesity – NHS.uk)
In each crawl, however, the prioritised crawls have managed to perform better than the breadth-first crawl with optimum context lengths between 6 and 9. The average optimum context length was again about 8 (7.6) words either side of the link.

Overall, the C-Value feature classification weighting scheme manages to perform better than the un-weighted classification scheme in 9 out of 15 (60%) of the cases using the optimum context length. The overall average context length is between 7 and 8 (7.4) words either side of the link.

5.3.3 Link Prioritisation: Combining Different Features

The change in the way the various link features are assessed means that the way the features are combined may differ from the ways previously explored. The linkScore combination experiments (as set out in section 5.2.3) were repeated with the C-Value weighting. This section sets out the results.

The crawls performed for the Bioinformatics Companies topic are shown in Figures 5.87-5.91.
Using all the feature available performs the best in terms of the C-Value weighting schemes in 4 out of the 5 sets of crawls. The ‘Page’ feature performing better in the nott.ac.uk crawls (Fig. 5.87). The C-Value weighted feature classification function performs better than the un-weighted version in 3 out of the 5 crawls, but only with the
optimum feature combination. This increases the overall efficiency of the crawl dramatically in most cases. In the two crawls where the un-weighted function fairs best (Fig 5.87 and 5.89) the un-weighted crawl is shown to be very similar to the weighted one. In Fig 5.89 for instance, the crawls follow a similar pattern at the beginning (the first 40 pages) but then the un-weighted crawl moves away sharply.

The GM Foods crawls (Fig. 5.92-5.96) show the un-weighted feature comparison consistently performing better than all the weighted feature comparison function variations.

Figure 5.92: C-Value Link Score Variation Results (GM Foods – actionbioscience.com)

Figure 5.93: C-Value Link Score Variation Results (GM Foods – Genome.gov)

Figure 5.94: C-Value Link Score Variation Results (GM Foods – intute hotlinks)

Figure 5.95: C-Value Link Score Variation Results (GM Foods – New Scientist)
Aside from the direct purpose of this experiment, an above average classification performance during the ‘content x context’ crawl was achieved in the actionbioscience.com crawl (Fig. 5.92) – 86% of the pages classified as ‘level 1’ were manually found to be correct. This crawl was out-performed by its non C-Value weighted counter-part, but as the analysis of these results found in section 5.2.2, this crawl’s very productive end was made up largely of false positives.

It is also interesting looking at the types of page found during this crawl, finding many different aspects of the arguments surrounding GM Foods. The ‘context x content’ crawl from Figure 5.92 discovered a sub-web from http://biotech-info.net which looks at the implications of Genetic Engineering in many different forms. The economic, environmental and health issues surrounding GM Foods are covered from both the positive and negative stand point. This coverage is despite the many ways that GM Foods are represented in the documents using various synonyms and abbreviations – including Genetically Modified Organisms (GMO), Genetically Modified Crops (GMC) and even Agricultural Biotechnology, all of which have been recognised as relevant. This result may be down to the links chosen by the crawler to follow – the ‘confusing’ pages for the classifier do not appear to have been followed meaning that the overall quality as well as the quantity of the relevant pages was increased.

The Obesity crawls show a similar pattern to the GM Foods crawls, with the un-weighted feature classification function performing better than many of the C-Value weighted ones. The MedicineNet crawl (Fig. 5.98) shows the un-weighted crawl, and the
crawl using all available link features performing similarly, with the C-Value weighted
crawl retrieving more relevant pages, before being caught up by the un-weighted crawl.

Figure 5.97: C-Value Link Score Variation Results (Obesity – diseasesconditions.com)

Figure 5.98: C-Value Link Score Variation Results (Obesity – MedicineNet)

Figure 5.99: C-Value Link Score Variation Results (Obesity – dh.gov)

Figure 5.100: C-Value Link Score Variation Results (Obesity – NHS.uk)

Figure 5.101: C-Value Link Score Variation Results (Obesity – NIH.gov)

Overall, the most prevalent weighted feature classification crawls were the ones
incorporating all the available features, performing best in 3 out of 5 sets of crawls (Fig.
5.99-5.101). The ‘Page’ and ‘Content’ features performed best in the other two crawls
(Fig. 5.97-5.98). In the diseasesconditions.com crawl (Fig. 5.97) the weighted function
and the un-weighted one follow the same path, discovering the same sub-Web, meaning
that the C-Value feature weighting played next to no part in the decisions this crawl took. Both of these crawls were based on the ‘Page’ feature, one with C-Value term weighting and one without. This means that the C-Value weighting didn’t alter the levels assigned to the pages encountered within this crawl.

5.3.4 Weighted Link Prioritisation in Larger Crawls

As with Section 5.2.4, larger crawls where carried out using the optimum parameters found during the previously described experiments, to test the crawl effectiveness on the more remote Web graphs. Crawls of 400 pages were used again, to see how pages further away from the neighbourhoods of the seed pages affected the crawl and its effectiveness at prioritising the link visitations.

In three of the five Bioinformatics Companies crawls (Fig. 102-106), the prioritised crawl where the feature classifications are weighted with the constituent terms’ C-Value score out-performs the un-weighted version. We can see the weighted version doing up to 3 times better than the un-weighted version here (see Fig. 5.103 and 5.104).

![Figure 5.102: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – nott.ac.uk)](image1)

![Figure 5.103: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – bioinformatics.org)](image2)

![Figure 5.104: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – Solpugid.com/Links.htm)](image3)

![Figure 5.105: Harvest Rate – C-Value Large Crawl (Bioinformatics Companies – bioinformatics.ca)](image4)
In the first case where the weighted crawl is out-performed (Fig. 5.102) the crawl explores the Apple computers website\(^{34}\) before moving on to many articles from the National Institutes of Health website\(^{35}\) - all of which resulted in very few relevant pages being found. In the second case (Fig. 5.105) the crawl got diverted to a largely unproductive area of the Web concerning Web standards. This meant that it only just did as well as the breadth-first crawl.

A mixed picture is given by the GM Foods crawls (Fig. 5.107-5.111) where the non weighted crawl out-performs the weighted version in the first two crawls (Fig. 5.107 and 5.108) and performs just as well in another two (Fig. 5.109 and 5.111). The weighting strategy only has a positive effect on the one crawl (Fig. 5.110) where it collects around twice as many relevant pages than the non weighted crawl overall. A small difference is made towards the end of the Genome.gov crawl (Fig. 5.110) where the weighted crawl manages to pull away slightly from the breadth-first and non-weighted crawl. This increase in gradient shows that it managed to find lucrative sub-webs faster than the other two crawls, but they are all too similar to say the method was successful in this case. The Genome.gov crawl also yielded similar results in previous sets of experiments, where the prioritised crawls perform as well as the breadth-first crawl. This is an example of the impact the page the crawl starts from can have on the overall crawl. If a seed page is well connected enough with relevant pages, there is little a prioritisation strategy can do in terms of retrieving more relevant results.

\(^{34}\) http://www.apple.com

\(^{35}\) http://www.nih.gov
The term weighting procedure has helped the crawls in the Obesity topic greatly, improving the results of 4 out of 5 of the crawls. In the un-weighted set of crawls (Fig. 5.51-5.55), the prioritised crawls only managed to collect more relevant pages than breadth-first in one of the five crawls (Fig 5.55 – the NIH.gov crawl). Usually there were only short term gains in efficiency, where the prioritised crawl would be caught up before too long by the breadth-first one. Here, we see the efficiency improve further, with the equivalent difficult crawls from the un-weighted set (Fig. 5.112-5.115) collecting 2-3
times as many relevant pages as the breadth-first crawl and similar gains over the un-weighted crawls. The NIH crawl (Fig. 5.116), where the un-weighted crawl gathers roughly 75% more relevant pages than the un-weighted one suffers due to changes in the classification of pages. The un-weighted crawl’s early productive period contains many misclassifications as discussed in Section 5.2.4. The quality of the pages gathered using C-Value weighted term comparisons is much better. While the un-weighted crawl finds many pages to do with various conditions within the NIH sub-web, this one centres mainly on the issue of childhood obesity which is very relevant to this topic.
5.3.5 Summary
The weighting of the term comparisons has shown to have a mixed effect on the outcome of the crawls. It has helped greatly in the Obesity domain crawls, but not in the others. Out of the 15 larger crawls (5 for each topic) the weighted version performed best in roughly 53% of them. As the majority of the smaller crawls showed the un-weighted crawl performing better than the weighted one, it appears that the C-Value weighting becomes more helpful in more remote areas of the Web, where relevant terms are scarcer.

5.5 Summary
This chapter has covered the various results gathered using the crawler introduced during Chapters 3-4. For each of three topic areas (Bioinformatics Companies, Genetically Modified Foods and Obesity), crawls were carried out from five separate start pages to find:

- The optimum context length (i.e. the number of words either side of a link) to be considered when judging the possible relevance of the links for link prioritisation;
- The optimum combination of link features for link prioritisation.

The results were compared to the baseline (the breadth-first crawl starting from the same point with the same topic description) using the harvest rate (ratio of the number of relevant pages against the total number of page downloads) as the evaluation metric. The methods were also tested for automatically generating weights for the combination of the link feature classifications into an overall prioritisation score (Section 5.2.5) as well as weighting the classification of the individual link features using the C-Value of terms being compared as an importance measure (Section 5.3).

The overall results show that the lexical similarity method is capable of driving a topic-focused Web crawl. In 25 out of the 30 larger crawls (i.e. after optimum variables have been found) the prioritised crawl produced better results than the breadth-first crawls, some achieving between two and six times as many relevant pages found in the 400 pages visited. The remaining 5 crawls performed as well as the breadth-first crawl and collected
similar amounts of documents to the other prioritised ones. This means that the quality of
the start page played a large part in the breadth-first crawl performing better, rather than
the prioritised crawl performing worse.

The linear combination of the link features while combining them to get the overall
prioritisation score has shown some gains in efficiency, but the overall outcome of the
experiments was that the un-weighted polynomial function performed better in the
majority of cases. As highlighted earlier, this may be due to the assumptions made during
the generation of the weights, and a refinement of this method may yield more promising
results.

The weighting of terms according to their C-Value within the individual feature
classifications seems to have a mixed effect on the crawls. The page classification
precision (see Section 5.3.1) was not significantly affected by the application of this
weighting scheme. While the weighted method performed better than breadth-first in 87%
of the larger crawls, it only performed better than the un-weighted version in around 53%.
This shows that this weighting scheme is not suitable in its current form. One possible
modification could be to add global significance weighting to the C-Value scores of the
terms, based on their frequency in the pages visited. This possible modification will be
discussed in more detail in Chapter 6.

Interesting trends and features of the crawl have opened up many areas for possible
work to increase the precision and efficiency of the crawl such as new link features and
analysis of relationships of vocabulary in the taxonomy. Throughout the Chapter,
interesting trends found within the results have been highlighted, and possible directions
for further work have been suggested. The conclusions which may be drawn from these
findings will be discussed in Chapter 6.
Chapter 6 – Conclusions and Future Work

This thesis has investigated topic-focused Web crawling, introducing methods for the specification and harvesting of fine-grained topics. This was achieved by building lexical profiles of the previously classified documents found in the levels of a topic taxonomy. The lexical similarity of link features (Section 3.2) to the various levels in the taxonomy was used to allow for a staged prioritisation of the links yet to be followed according to their expected benefit to the crawl (either in relevant pages, or pages likely to link to relevant pages etc.). This prioritisation method (along with various variations) has been evaluated in terms of its efficiency in gathering the best pages (harvest rate) against a breadth-first strategy.

The thesis introduced the following approaches to topic-focused Web crawling:

1. A term-based lexical profiling of existing Web directories, which is used to model taxonomic relationships between different topics. The model uses all terms found in documents from a given level in the taxonomy, and generates a lexical profile of the entire topic that will be used to evaluate candidate pages and links as relevant/irrelevant for that topic. This method proved to be efficient in modelling content of fine-grained topics in particular, where ML approaches would typically fail to generalise the topic due to few training documents being available.

2. A prioritisation strategy that can be used to improve frontier ranking, i.e. to allow more important links to be visited earlier. The strategy involves combining three features that characterise the link: its anchor text (content), surrounding text (context) and the source document (page). Each of these textual neighbourhoods have been compared to the topic model (as well as to the models of the taxonomically related topics) and the closest levels have been assigned to the links. This comparison has been implemented using an un-weighted strategy and using individual terms’ C-Value significance score to weight terms within the links, pages and topic model. The scores from the three textual features have been combined in two ways: by multiplication of the individual values, and by linear combination.
The use of the lexical profiling methods to create models of fine-grained topics from relatively small document sets has been shown to be feasible. Possible areas for refinement of the method have also been discussed throughout Chapter 5. This chapter will summarise these findings and set out the conclusions which can be drawn from them, suggesting directions for further work.

**Link Prioritisation using Taxonomical Models based on Lexical Profiles Improves Topic-Focused Web Crawling Harvest Rates**

The harvest rate has been used to evaluate the effectiveness of the prioritisation strategy in gathering the most relevant pages with the least amount of ‘wasted’ downloads, comparing them against the un-prioritised, breadth-first crawls. Overall, the prioritised crawls tended to fare better than the breadth-first crawls, finding up to six times as many relevant documents with the same number of downloads.

Most of the crawls showed that the more features of the links that were considered, the better. The combination of all three proposed link features (Content, Context and source Page level) performed the best in all but a few cases. The ‘Page’ feature was present in the overwhelming majority of optimum functions, showing the importance of where the link is found in terms of determining where the link points to. Clues were also offered as to other features that may be beneficial to the crawl (i.e. the proximity feature suggested in Section 5.2.2).

One of the most interesting patterns that emerged from the results analysis, is the way in which the vocabulary of the pages listed under the topic headings encodes the topic itself. It often transpired that many false positives within the results of the crawls were caused by parts of the topic being represented, but not others. A fundamental assumption that was made in this work is that vocabulary would be shared between the levels in the taxonomy. This can be demonstrated by looking at one of the example topic areas used ‘Science/Biology/Bioinformatics/Companies’. This taxonomy implies that each child branch of the topic is related to the parent in some way. For instance, it implies we are not looking for any company, but ones related to the bioinformatics field. The assumption here is that this relation implies that vocabulary extracted from pages at the ‘companies’ level would also encode some aspect of ‘bioinformatics’ – at least, enough to infer whether
Throughout the results analysis, with each topic domain, false positives were present where the classification was justifiable given the target level on its own, without regard to the others. For instance, the GM Foods topic found many sites related to food, and food related issues, but appeared to lack the ‘genetics’ angle that would make them true positives. This implies that the relationship between parent and child branches is not sufficiently encoded implicitly in the child level, and therefore future attempts to encode topics like this may benefit from adding some sort of explicit inference over topic level relationships (e.g. including representative terms from parent levels).

The obvious limitation of the proposed method is that it is not possible to generate models for topics that are not represented in the taxonomy. The main advantage of using a public taxonomy as a source of training data is that it is curated by many people allowing large scale scrutiny of the decisions made. This means that the data held within it can be assumed to be accurate and the descriptions applied to documents apt according to their content. There are some drawbacks to its use though. Firstly, topics that are not present within the taxonomy become hard to model. While the user could probably create their own set of training examples in the sort of hierarchical structure the ODP offers this collection would lose the objectivity and public scrutiny that is the main benefit of employing the ODP. Furthermore, using ODP means that the models (the terms extracted from example documents to represent the topic area) created are static in what may be a dynamic topic area. For instance, a topic may be modelled using current documents from within its domain, but should that domain change (i.e. new concepts arise, or existing concepts are revised) then the documents within the taxonomy would need to be edited (or new documents added) and the model would need to be remade. However, as the ODP only holds links to the documents, it could be assumed that as content within the Web is dynamic, documents already visited may themselves represent the dynamics of the topic to which they are related and visiting them again may suitably model the current state of the topic area without having to wait for new documents. Quantifying the effects of these issues along with the effects they have on the topic-focused Web crawling process lies outside of the scope of this work, but would be an interesting topic for future work.
The way in which content is treated should also be looked at. Throughout the work in this thesis content and links have been processed with no consideration to where they appeared within the page. Menu items, for instance, were treated as text for the purposes of classification for the page feature. This added noise into certain crawls. While these structures offer vital clues as to the purpose of the site as a whole, the weight given to these items may be best reduced when concerning individual pages and their features. One possible area of further investigation is including this sort of context into feature classification to improve the crawler’s efficiency, and which particular areas of the site benefit individual tasks (for instance link prioritisation, page classification or even at the site level rather than the page).

**Better Weighting Strategies are Needed to Improve Harvest Rates**

Combining the link features linearly to get the overall prioritisation score has shown short term gains, but the overall outcome of the crawl was closer to the breadth-first results than the prioritised crawl that used the polynomial combination. This may be due to the assumptions made during the generation of the weights. The fact that each set of weights was evaluated on a static queue, rather than a staged queue which would have been closer to the real world scenario they were to be employed in may have caused weights to be chosen that were not optimal for the task. Pages can only be visited during a crawl once a page that links to it has been visited. While the weighting strategy implemented moves towards finding more relevant pages quickly (the best weights were the ones that resulted in the highest number of relevant documents at the top of the queue), the assumption that the links are already known is evident. By looking at the organisation of the example Web graph, the idea that the crawl should also aim towards pages which stand a high chance of offering other good links should also be included when generating weights. Using example Web graphs to evaluate the weight assignments may be more beneficial in generating a less short-sighted result.

The weighting of terms according to C-Value within the individual feature classifications seem to have a mixed effect on the crawl, with average page classification precision remaining roughly the same when adding term importance weighting and the link prioritisation performing slightly less well. As there are so few training documents, assigning importance based on them may be short sighted, whereas simply giving the
crawl the terms that appear may allow the crawl to generalise more, and be more sensitive to wider areas of the topic. This may be further rectified by normalising the importance using a term frequency – inverse document frequency (TF*IDF) normalisation strategy, or applying an IDF-like normalisation factor to the C-Value score. Altering the context length and the combination of features used for link prioritisation has not had an effect on this. As there are so few training documents, a semi-supervised approach to term weighting may be applied, where extra weight is assigned to terms which have previously yielded ‘good’ results during the crawl. The notion of a ‘good’ result is difficult to determine as any particular link may prove beneficial (in that it provided more good links) many pages down the line.

Future Work

The findings of the investigations carried out have offered various opportunities for further study – this section details the possible directions for further work.

- **Taxonomic Relationships:** Investigation into the relationship between a parent and child nodes in the taxonomy could improve the precision of the classification results. The experiments carried out show that the vocabulary of the parent topics is not always suitably represented in the sub-topics. As already mentioned, one example is the pages found in the GM Foods topic, which were relevant to foods, but not necessarily to Genetically Modified Foods. The ‘food’ vocabulary was represented, but it lacked the ‘Genetics’ factor that would make it relevant. Understanding the relationship between the levels and incorporating it into the classification decisions may result in better results, and may therefore improve the efficiency of the crawl through more precise link feature level assignments. This way, to be classified as relevant, a document would have to incorporate sufficient vocabulary from the target node and the nodes leading up to it to be classed as relevant, increasing precision.

- **Additional Link Features:** The investigations into the various combinations of link features tend towards the more information used the better, with the combination of all three features available performing best in the majority of cases. Investigations into other suitable features may improve the efficiency of the crawls further, developing and exploiting a broader picture of what makes a relevant or
related link. As already proposed in Section 5.2.2, one possible feature would be proximity to other links which yielded a positive result. Also proposed is the investigation into where in the page good links are more likely to occur (i.e. within the main body of text, the menu, feature boxes etc.). This could be performed during a crawl through an incremental, semi-supervised approach where the links followed and their outcome is used as the training data as it is collected. This way features such as where the good links are appearing can be utilised. Web pages’ markup structure makes the extraction of the layout relatively simple and data through Web crawling could yield statistics that could point towards these sort of layout features.

- **Automatic Link Feature Weight Allocation:** The feature weight assignment method implemented in this thesis has been evaluated to be unsuitable. This is due to its short sightedness in that it only looks towards getting the most relevant pages – the best set of weights is the one with the highest number of relevant pages at the top of the list. In real world crawls, the topic-focused crawler has to take into account that the pages likely to yield good links also have to be visited and not overlooked. One proposed method of improving this feature weight allocation method is to use real-world incremental data – where the links available in the queue depend on the page at its top. This more dynamic approach means that the fitness of the weight combination would be evaluated in a situation closer to the actual application. This would build in the ability to strike a balance between finding pages that are relevant to the topic and finding pages that will provide other useful links while the current implementation only focuses on the former. Implementing this using Genetic Algorithms (GA) may be more efficient and produce better results than the brute-force method introduced in this work. GA means that the precision of the weights would not have to be pre-defined so that the absolute best weights could be found. GA could also potentially reduce the number of iterations needed to settle on the optimum weights.

Overall, the methods presented in this thesis has been shown to be effective for focused crawling for fine-grained topics, based on taxonomic descriptions using small numbers of example documents. The use of terminological lexical profiling of taxonomic Web directories to drive topic-focused Web crawls has been shown to improve efficiency over
the naïve crawling methods and generate efficient models from a smaller number of documents than more widely implemented machine learning based methods. More precisely, the suggested methods managed to improve the breadth-first approach by collecting up to 6 times as many relevant pages. The results of several experiments suggested that lexical profiling of term-based representations of topic models can be an effective strategy to build topic-focused Web crawlers.
Appendix

1. Example Settings File

```plaintext
# # Separator: ":" separates label from value
# PageModel: the kind of Model used for page classification (svm/term)
# LinkModel: the kind of Model used for link classification (svm/term)
# Maxwords: the number of words to use as context length either side
# of the <a>...</a> tags
# TrainingLoc: The location of the training data
# Log: The location that the Log File should be created in
# Seed: One of the seed pages to be used during the crawl (multiple elements)
# CrawlLimit: The maximum number of pages which should be collected
# QueueLimit: The maximum number of links that can be stored in the
# # Queue (Default = Unlimited)
# #(hash): at beginning of line: comment marker
# sort: Whether to sort the todo queue or not (true/false - default true)
# pause: The time the crawler should wait between page fetches (sec)
# This is to prevent abuse of network resources, server and
# (client-side)
# c: SVM Parameter 'Cost' (default 1)
# g: SVM Parameter Gamma for the kernel function
#
PageModel: term
LinkModel: term
maxwords: 5
TrainingLoc: ./training
Log: ./log.txt
CrawlLimit: 100
DatabaseLoc: localhost
seed: http://evol.nott.ac.uk/cmelun/links.html
pause: 5
C: 0.03125
g: 0.0076125
```
2. **visitStringNode** Method

```java
if(ignoreTag>0){}
else{
    String temp = stringNode.getText();
    if(temp.trim().equals("")==false)
    {
        text = text+stringNode.getText().replace("\nb"," ")+
    }
    if(linkTag)
    {
        text = text+" </link> ";
        linkTag=false;
    }
}
text = Translate.decode(text); // translates numeric character references to real characters
```

3. **TermModel.train** Method

```java
public void train()
{
    model = new TermList[maxLevel];
    defaultLevel = maxLevel+1;
    for (int a=0;a<maxLevel;a++) model[a] = new TermList();

    Iterator it = examples.iterator();

    while(it.hasNext())
    {
        Example temp = (Example)it.next();
        int level = temp.getLevel();
        model[level-1].add(temp.getTerms());
    }

    for (int a=0;a<maxLevel;a++)parent.log.write("Level "+(a+1)+" TermList:\r\n"+model[a].toString());
    setBoundaries();
}
```
4. TermLinkModel.addExample(Page p, int l) Method

{  
  if (parent.model instanceof TermModel! = true)
    {  
      Examples.add(new Example(l,p.getTerms()));  
      if (maxLevel<l) maxLevel = l;  
    }
  
}

5. TermLinkModel.train() Method

{  
  if (parent.model instanceof SVMModel)
    {  
      model = new TermList[maxLevel];
      for (int a=0;a<maxLevel;a++) model[a] = new TermList();
      
      Iterator it = Examples.iterator();
      
      while(it.hasNext())
        {  
          Example temp = (Example)it.next();
          int level = temp.getLevel();
          model[level-1].add(temp.getTerms());
        }
    }
  else maxLevel = parent.model.getMaxLevel();
  
}

6. TermModel.term_classify(String text) Method

{  
  float max = 0;  
  int level = defaultLevel;  
  if (text!=null & text.trim().equals("")==false)  
    for(int a=0;a<model.length;a++)  

float score = model[a].term_lexicalSim(text);
if(score>max && score>=minScores[a])
{
    max = score;
    level=a;
}
return level+1;
return result/terms.size();

7. TermLinkModel.profile(String[] con) Method

TermList result = new TermList();
for(int c = 0;c<con.length;c++){
    String[] w = con[c].split(" ");
    for(int a=0;a<w.length;a++)
    {
        String term = w[a];
        result.add(term,1);
        for(int b=a+1;b<w.length;b++)
        {
            term = term + " " + w[b];
            result.add(term,1);
        }
    }
}
return result;
8. Abstract Data Types

These UML diagrams show the structure of the main ADTs used by the Web Crawler. The last shows the overall structure of the program. The full source is available at http://www.gnode1.mib.man.ac.uk/WebCrawler.
9. Manual Classification Lists

These are the 45 page sets chosen for each topic and manually classified to test classification precision. The 360 page set and inter-annotator data is available at http://www.gnode1.mib.man.ac.uk/WebCrawler

Bioinformatics Companies

<table>
<thead>
<tr>
<th>Count</th>
<th>location</th>
<th>Automatic Level</th>
<th>Manual Level</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td><a href="http://www.cbs.dtu.dk/biolinks/dbase.html">http://www.cbs.dtu.dk/biolinks/dbase.html</a></td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.oxfordjournals.org/japan/for_librarians/">http://www.oxfordjournals.org/japan/for_librarians/</a></td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.mgms.org/diary.htm">http://www.mgms.org/diary.htm</a></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.cbs.dtu.dk/databases/">http://www.cbs.dtu.dk/databases/</a></td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.oxfordjournals.org/contact_us">http://www.oxfordjournals.org/contact_us</a></td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><a href="http://portal.acs.org:80/portal/PublicWebSite/membership/acs/welcoming/international/japanese/index.htm">http://portal.acs.org:80/portal/PublicWebSite/membership/acs/welcoming/international/japanese/index.htm</a></td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.ccdc.cam.ac.uk">http://www.ccdc.cam.ac.uk</a></td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.oxfordjournals.org/help/techinfo/oaipmh_implement.html">http://www.oxfordjournals.org/help/techinfo/oaipmh_implement.html</a></td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.oxfordjournals.org/faq/for_librarians/ip_registration.html">http://www.oxfordjournals.org/faq/for_librarians/ip_registration.html</a></td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td><a href="http://evol.nott.ac.uk/cmelun/links.html">http://evol.nott.ac.uk/cmelun/links.html</a></td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td><a href="http://oasys2.confex.com/acs/235nm/techprogram/">http://oasys2.confex.com/acs/235nm/techprogram/</a></td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td><a href="http://portal.acs.org:80/portal/PublicWebSite/meetings/national/spring2008/services/index.htm">http://portal.acs.org:80/portal/PublicWebSite/meetings/national/spring2008/services/index.htm</a></td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>URL</td>
<td>Page</td>
<td>Column 1</td>
<td>Column 2</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>15</td>
<td><a href="http://www.jax.org/">http://www.jax.org/</a></td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td><a href="http://www.ccdc.cam.ac.uk/news/">http://www.ccdc.cam.ac.uk/news/</a></td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>17</td>
<td><a href="http://www.oxfordjournals.org/corporate_services/">http://www.oxfordjournals.org/corporate_services/</a></td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>18</td>
<td><a href="http://www.ccdc.cam.ac.uk/news/archive_2003/">http://www.ccdc.cam.ac.uk/news/archive_2003/</a></td>
<td>5</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td><a href="http://jaxmice.jax.org/genetichealth/index.html">http://jaxmice.jax.org/genetichealth/index.html</a></td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>20</td>
<td><a href="http://www.ccdc.cam.ac.uk/products/csd_system/">http://www.ccdc.cam.ac.uk/products/csd_system/</a></td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>22</td>
<td><a href="http://brain.oxfordjournals.org/current.dtl">http://brain.oxfordjournals.org/current.dtl</a></td>
<td>5</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.oxfordjournals.org/for_librarians/promotional_support.html">http://www.oxfordjournals.org/for_librarians/promotional_support.html</a></td>
<td>1</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>23</td>
<td><a href="http://www.ccdc.cam.ac.uk/products/gold_suite/">http://www.ccdc.cam.ac.uk/products/gold_suite/</a></td>
<td>2</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>24</td>
<td><a href="http://www.oxfordjournals.org/news/marketing_contacts.html">http://www.oxfordjournals.org/news/marketing_contacts.html</a></td>
<td>1</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>26</td>
<td><a href="http://www.ccdc.cam.ac.uk/news/archive_2006/">http://www.ccdc.cam.ac.uk/news/archive_2006/</a></td>
<td>2</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>27</td>
<td><a href="http://phylogeny.arizona.edu/tree/home.pages/phylobiology.html">http://phylogeny.arizona.edu/tree/home.pages/phylobiology.html</a></td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td><a href="http://www.oxfordjournals.org/our_journals/index.html">http://www.oxfordjournals.org/our_journals/index.html</a></td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td><a href="http://nar.oxfordjournals.org/cgi/alerts/etoc">http://nar.oxfordjournals.org/cgi/alerts/etoc</a></td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td><a href="http://www.cbs.dtu.dk/dtucourse/intmasters.html">http://www.cbs.dtu.dk/dtucourse/intmasters.html</a></td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>31</td>
<td><a href="http://www.biotech.ist.unige.it/interlab/mpdb.html">http://www.biotech.ist.unige.it/interlab/mpdb.html</a></td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>32</td>
<td><a href="http://www.cbs.dtu.dk/contact/">http://www.cbs.dtu.dk/contact/</a></td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>33</td>
<td><a href="http://bioinformatics.oxfordjournals.org/cgi/alerts/etoc">http://bioinformatics.oxfordjournals.org/cgi/alerts/etoc</a></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
GM Foods

<table>
<thead>
<tr>
<th>location</th>
<th>Automatic Level</th>
<th>Manual Level</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 <a href="http://www.actionbioscience.org/genomic/bereano.html">http://www.actionbioscience.org/genomic/bereano.html</a></td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.actionbioscience.org/newfrontiers/merkle.html?print">http://www.actionbioscience.org/newfrontiers/merkle.html?print</a></td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3 <a href="http://www.actionbioscience.org/education/glenncomm.html">http://www.actionbioscience.org/education/glenncomm.html</a></td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>4 <a href="http://www.actionbioscience.org/environment/kassas.html">http://www.actionbioscience.org/environment/kassas.html</a></td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td><a href="http://www.actionbioscience.org/biodiversity/hawtin_cherfas.html">http://www.actionbioscience.org/biodiversity/hawtin_cherfas.html</a></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6 <a href="http://www.actionbioscience.org/evolution/">http://www.actionbioscience.org/evolution/</a></td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7 <a href="http://www.actionbioscience.org/biotech/">http://www.actionbioscience.org/biotech/</a></td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8 <a href="http://www.genetic-id.com/request/">http://www.genetic-id.com/request/</a></td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.actionbioscience.org/biotech/oxfam_spinney.html">http://www.actionbioscience.org/biotech/oxfam_spinney.html</a></td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.ornl.gov/hgmis/">http://www.ornl.gov/hgmis/</a></td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td><a href="http://www.actionbioscience.org/announcements/aibs_participates_in_encyclope.html">http://www.actionbioscience.org/announcements/aibs_participates_in_encyclope.html</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td><a href="http://www.ornl.gov/TechResources/Human_Genome/glossary/">http://www.ornl.gov/TechResources/Human_Genome/glossary/</a></td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td><a href="http://www.actionbioscience.org/newfrontiers/davis.html">http://www.actionbioscience.org/newfrontiers/davis.html</a></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td><a href="http://www.actionbioscience.org/">http://www.actionbioscience.org/</a></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td><a href="http://www.actionbioscience.org/biodiversity/wilson.html">http://www.actionbioscience.org/biodiversity/wilson.html</a></td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td><a href="http://www.actionbioscience.org/announcements/nabt_and_aibs_partner_to_offer.html">http://www.actionbioscience.org/announcements/nabt_and_aibs_partner_to_offer.html</a></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>17</td>
<td><a href="http://www.actionbioscience.org/genomic/barash.html">http://www.actionbioscience.org/genomic/barash.html</a></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td><a href="http://www.actionbioscience.org/biodiversity/plotkinbookexcerpt.html">http://www.actionbioscience.org/biodiversity/plotkinbookexcerpt.html</a></td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td><a href="http://www.actionbioscience.org/announcements/5-mar-2008.html">http://www.actionbioscience.org/announcements/5-mar-2008.html</a></td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>20</td>
<td><a href="http://www.actionbioscience.org/educators/educator-resources.html">http://www.actionbioscience.org/educators/educator-resources.html</a></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td><a href="http://www.genetic-id.com/services/certification/">http://www.genetic-id.com/services/certification/</a></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td><a href="http://www.actionbioscience.org/privacy.html">http://www.actionbioscience.org/privacy.html</a></td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>23</td>
<td><a href="http://www.actionbioscience.org/environment/derocher.html">http://www.actionbioscience.org/environment/derocher.html</a></td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>24</td>
<td><a href="http://www.actionbioscience.org/education/aaasproj2061.html">http://www.actionbioscience.org/education/aaasproj2061.html</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>25</td>
<td><a href="http://www.actionbioscience.org/search.html">http://www.actionbioscience.org/search.html</a></td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>26</td>
<td><a href="http://www.actionbioscience.org/newfrontiers/">http://www.actionbioscience.org/newfrontiers/</a></td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td><a href="http://www.actionbioscience.org/education/guilfoile.html">http://www.actionbioscience.org/education/guilfoile.html</a></td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Location</td>
<td>Automatic Level</td>
<td>Manual Level</td>
<td>Count</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>1 <a href="http://www.diseasesconditions.com/Eye_Diseases/">http://www.diseasesconditions.com/Eye_Diseases/</a></td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2 <a href="http://www.calregistry.com">http://www.calregistry.com</a></td>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>3 <a href="http://www.diseasesconditions.com/">http://www.diseasesconditions.com/</a></td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4 <a href="http://www.cdc.gov/osi/goals/places/placesWorkplaces.html">http://www.cdc.gov/osi/goals/places/placesWorkplaces.html</a></td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5 <a href="http://www.cdc.gov/healthmarketing/">http://www.cdc.gov/healthmarketing/</a></td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6 <a href="http://www.cdc.gov/washington/testimony.html">http://www.cdc.gov/washington/testimony.html</a></td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>7 <a href="https://secure.mktwebdesign.com/aarda/contribute.htm">https://secure.mktwebdesign.com/aarda/contribute.htm</a></td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8 <a href="http://www.cdc.gov/Environmental/">http://www.cdc.gov/Environmental/</a></td>
<td>6</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>9 <a href="http://www.cdc.gov/healthmarketing/resources.htm">http://www.cdc.gov/healthmarketing/resources.htm</a></td>
<td>1</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>10 <a href="http://www.cdc.gov">http://www.cdc.gov</a></td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>11 <a href="http://www.diseasesconditions.com/Resources/Directories/">http://www.diseasesconditions.com/Resources/Directories/</a></td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>12 <a href="http://www.cdc.gov/spanish/">http://www.cdc.gov/spanish/</a></td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>13 <a href="http://www.life-in-motion.org/">http://www.life-in-motion.org/</a></td>
<td>4</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>14 <a href="http://www.aarda.org/research_display.php?ID=37">http://www.aarda.org/research_display.php?ID=37</a></td>
<td>1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>15 <a href="http://www.aarda.org/links.php">http://www.aarda.org/links.php</a></td>
<td>6</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>16 <a href="http://www.cdc.gov/about/leadership/CDCdirector.htm">http://www.cdc.gov/about/leadership/CDCdirector.htm</a></td>
<td>4</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>17 <a href="http://www.aarda.org/survey.php">http://www.aarda.org/survey.php</a></td>
<td>1</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>18 <a href="http://www.cdc.gov/Other/egovernment.html">http://www.cdc.gov/Other/egovernment.html</a></td>
<td>6</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>20 <a href="http://www.aarda.org/feedback.php">http://www.aarda.org/feedback.php</a></td>
<td>1</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>URL</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------------------</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>21</td>
<td><a href="http://www.aarda.org/research_display.php?ID=8">http://www.aarda.org/research_display.php?ID=8</a></td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td><a href="http://www.aarda.org/">http://www.aarda.org/</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td><a href="http://www.cdc.gov/osi/goals/places/placesHCSettings.html">http://www.cdc.gov/osi/goals/places/placesHCSettings.html</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td><a href="http://www.diseasesconditions.com/Sexual_Transmitted_Diseases/">http://www.diseasesconditions.com/Sexual_Transmitted_Diseases/</a></td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td><a href="http://www.diseasesconditions.com/Ear_Nose_and_Throat/">http://www.diseasesconditions.com/Ear_Nose_and_Throat/</a></td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td><a href="http://www.diseasesconditions.com/Genetic_Disorders/">http://www.diseasesconditions.com/Genetic_Disorders/</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td><a href="http://www.cdc.gov/doc.do?id=0900f3ec80093c90">http://www.cdc.gov/doc.do?id=0900f3ec80093c90</a></td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>29</td>
<td><a href="http://service.govdelivery.com/service/subscribe.html?code=USCDC_24_1">http://service.govdelivery.com/service/subscribe.html?code=USCDC_24_1</a></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td><a href="http://www.cdc.gov/osi/index.html">http://www.cdc.gov/osi/index.html</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>31</td>
<td><a href="http://www.diseasesconditions.com/Neurological_Disorders/Movement_Disorders/Angelman_Syndrome/">http://www.diseasesconditions.com/Neurological_Disorders/Movement_Disorders/Angelman_Syndrome/</a></td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td><a href="http://www.pandemicflu.gov/">http://www.pandemicflu.gov/</a></td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>33</td>
<td><a href="http://www.diseasesconditions.com/link.html">http://www.diseasesconditions.com/link.html</a></td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>34</td>
<td><a href="http://www.cdc.gov/contact/">http://www.cdc.gov/contact/</a></td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>35</td>
<td><a href="http://www.cdc.gov/about/">http://www.cdc.gov/about/</a></td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>36</td>
<td><a href="http://www.cdc.gov/about/goals.htm">http://www.cdc.gov/about/goals.htm</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>37</td>
<td><a href="http://www.cdc.gov/healthmarketing/basics.htm">http://www.cdc.gov/healthmarketing/basics.htm</a></td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>38</td>
<td><a href="http://www.diseasesconditions.com/Musculoskeletal_Disorders/">http://www.diseasesconditions.com/Musculoskeletal_Disorders/</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td><a href="http://www.aarda.org/research_archived.php">http://www.aarda.org/research_archived.php</a></td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>URL</td>
<td>Row</td>
<td>Col</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>40</td>
<td><a href="http://www.cdc.gov/healthmarketing/training.htm">http://www.cdc.gov/healthmarketing/training.htm</a></td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>41</td>
<td><a href="http://www.diseasesconditions.com/Wounds_and_Injuries/index.html">http://www.diseasesconditions.com/Wounds_and_Injuries/index.html</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>42</td>
<td><a href="http://www.diseasesconditions.com/Communication_Disorders/">http://www.diseasesconditions.com/Communication_Disorders/</a></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>43</td>
<td><a href="http://www.cdc.gov/HealthMarketing/APHA2007/Health_Marketing_Panel.htm">http://www.cdc.gov/HealthMarketing/APHA2007/Health_Marketing_Panel.htm</a></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>44</td>
<td><a href="http://www.cdc.gov/osi/contact.html">http://www.cdc.gov/osi/contact.html</a></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>45</td>
<td><a href="http://www.cdc.gov/od/oc/media/">http://www.cdc.gov/od/oc/media/</a></td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
References


167


