Artificial Immune Systems for Web Content Mining: Focusing on the Discovery of Interesting Information

A thesis submitted to

The University of Kent

in the subject of computer science

for the degree of

doctor of philosophy

By

Andrew Secker

June 2006

Abstract

This thesis explores the way in which biological metaphors can be applied to web content mining and, more specifically, the identification of interesting information in web documents. Web content mining is the use of content found on the web, most usually the text found on web pages, for data mining tasks such as classification. Due to the nature of the search domain, i.e. the web content is noisy and undergoing constant change, an adaptive system is required. The discovery of interesting information is an advance on basic text mining in that it aims to identify text that is novel, unexpected or surprising to a user, whilst still being relevant. This thesis investigates the use of Artificial Immune Systems (AIS) applied to discovery of interesting information as AIS are thought to confer the adaptability and learning required for this task.

Two novel Artificial Immune Systems are described and tested. AISEC (Artificial Immune System for Interesting E-mail Classification) is a novel, immune inspired system for the classification of e-mail. It is shown that AISEC performs with a predictive accuracy comparable to a naïve Bayesian algorithm when continually classifying e-mail collected from a real user. This section contributes to the understanding of how AIS react in a continuous learning scenario.

Following from the knowledge gained by testing AISEC, AISIID (Artificial Immune system for Interesting Information Discovery) is then described. A study involving the subjective evaluation of the results by users is undertaken and AISIID is seen to discover pages rated more interesting by users than a comparative system. The results of this study also reveal AISIID performs with subjective quality similar to the well known search engine, Google. This leads to a contribution regarding a better understanding of the user's perception of interestingness and possible inadequacies in the current understanding of interestingness regarding text documents.

Acknowledgements

I would like to thank my immediate family for their help and support, without which I never would have been able to begin this research. Many years ago, when I was around 8 years old my parents bought me a Sinclair Spectrum +3 computer. At that time I wanted a BBC Micro, a popular model at the time, so I feel I was a little ungracious for the gift. However, while many remember the Sinclair Spectrum range of computers, few remember the BBC Micro, so maybe my parents knew more about computers than they were letting on? This present was one of the most influential items I have ever owned. Its simplistic BASIC programming language allowed me to experience and enjoy programming for the first time. It was from there that my interest in computing grew to result, 18 years later, in this PhD thesis. Apart from their contribution in this respect I would like to thank them for their consistent support throughout my time as a student, both undergraduate then postgraduate. I don't think they even once suggested I got a proper job as I, once again, raided their fridge and cupboards for much needed food!

I would also especially like to thank my supervisors, Dr Alex Freitas and Dr Jon Timmis. Their input has been invaluable throughout my time as a postgraduate. Working with them has been both enlightening and a pleasure. Both have spent many hours helping me with my research and offering advice concerning this thesis and this considerable effort has not gone unnoticed.

Finally I would like to thank the Computing Laboratory at the University of Kent for kindly supplying me with a bursary. Without this financial help I would never have been able to even begin this research.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Tables vii

List of Figures viii

Chapter 1 Introduction 1

1.1 Motivation and Background 1

1.2 Challenges 3

1.3 Goals and Contributions 5

1.4 Thesis Outline 6

1.5 Publications 7

Chapter 2 Data and Web Mining 9

2.1 Data Mining and Knowledge Discovery 9

2.2 Text Mining and Information Retrieval 18

2.3 WordNet 25

2.4 E-mail Classification 28

2.5 Interesting Knowledge 31

2.6 Web Mining 37

2.7 Summary 51

Chapter 3 Immune Systems 53

3.1 Artificial Immune Systems in Context: Biologically Inspired Paradigms 53

3.2 Immunity 55

3.3 Artificial Immune Systems 66

3.4 Framework for Artificial Immune Systems 67

3.5 Artificial Immune Systems for Data Mining and Other Applications 80
3.6 Summary 95

Chapter 4 AISEC – an Artificial Immune System for E-mail Classification 96

4.1 Introduction 96
4.2 Motivations 97
4.3 An Overview of the Artificial Immune System for E-mail Classification (AISEC) Algorithm 98
4.4 The ASIEC System in Detail 102
4.5 Computational Complexity Analysis 109
4.6 Experimental Results 112
4.7 Sensitivity Analysis of AISEC Parameters 119
4.8 Using the Parameter Analysis 134
4.9 Investigating Effect of Class Distribution 136

Chapter 5 AISIID – an Artificial Immune System for Interesting Information Discovery 139

5.1 Introduction 139
5.2 Motivation 140
5.3 Overview of AISIID 143
5.4 Algorithm Description 147
5.5 Pseudocode 167
5.6 Visualisation of Search Progress 172
5.7 Summary 176

Chapter 6 Analysis of AISIID 177

6.1 Objective Tests 178
6.2 Observations Upon Termination 190
6.3 Subjective Tests 194

Chapter 7 Further Work and Conclusion 210

7.1 Assessment of AISEC 210
7.2 Future Work: Improving AISEC 211
7.3 Assessment of AISIID 213
7.4 Future Work: Improving AISIID 215
7.5 Reflections on AIS for Web Mining 218
7.6 Interestingness Revisited 219

7.7 Summary 221

References 223

Appendix A Stopwords 250

Appendix B Roulette Wheel Selection 251

B.1 Pseudocode 252

References 252

Appendix C Statistical Testing 253

C.1 Student's t-test 254

C.2 Large Sample Method 256

References 257

Appendix D Investigation into TFIDF Ranking 258

D.1 Investigation Method 258

D.2 Results and Discussion 259

D.3 Summary 261

URLs of the Web Pages from Table D.1 261

References 262

Appendix E User Test URLs 263

List of Tables

Table 4.1. Parameters used for testing AISEC 113
Table 4.2. Predictive accuracy for AISEC and Naïve Bayes 114
Table 4.3. Assessment of performance with randomised data 117
Table 4.4. Number of e-mails per class in data sets 122
Table 4.5. AISEC parameter configuration 122
Table 4.6. Optimised parameter set 135
Table 4.7. Result of tests using optimised parameters 135
Table 5.1. Parameters and legal ranges for AISIID 167
Table 6.1. Parameter values for AISIID tests 179
Table 6.2. Summary of results for experiments varying $K_{suppress}$ 182
Table 6.3. Summary of test users and their subjects 201
Table 6.4. Mean subjective interestingness scores for AISIID user tests 201
Table D.1. Full results of TFIDF test 259
Table D.2. Summarised Results 260

List of Figures

Figure 2.1. General layout of an IF-THEN rule 16
Figure 2.2. Two common forms for conditions in an IF-THEN rule 16
Figure 2.3. Example chart describing the relationship between amount of training and predictive accuracy over a test set 18
Figure 2.4. Example Confusion Matrix 24
Figure 2.5. A graphical representation of the spidering process. 43
Figure 3.1. Primary and secondary immune response. 57
Figure 3.2. Diagram of a B-cell. Numerous antibodies can be seen on its surface and the Fab and Fc regions of an antibody are labelled. 59
Figure 3.3. A T-cell's maturation in the thymus 63
Figure 3.4. Anatomy of an antibody according to the Jernian Immune Network Theory. 64
Figure 3.5. Degrees of affinity between a lymphocyte (L) and an antigen (Ag) 66
Figure 3.6. Analogy between B-cell receptor and artificial immune cell feature vector. 69
Figure 3.7. Diagrammatic depiction of recognition region and cross reactivity threshold. 71
Figure 3.8. Diagrammatic representation of B-cell maturation process. 78
Figure 4.1. High level view of the AISEC system after initialisation 100
Figure 4.2. Recognition and classification regions surrounding a B-cell 101
Figure 4.3. Structure of B-cell vector 103
Figure 4.4. Change in classification accuracy by e-mails classified 116
Figure 4.5. Change in B-cell population over time 118
Figure 4.6. Influence of Classification Threshold (`Kc`) 127
Figure 4.7. Influence of Affinity Threshold (`Ka`) 128
Figure 4.8. Influence of Clone Constant (`Kl`) 129
Figure 4.9. Influence of Mutate Constant (`Km`) 130
Figure 4.10. Influence of Initial Naive B-cell stimulation (`Ksb`) 130

Figure 4.11. Influence of Initial Memory Cell Stimulation (Ksm) 131
Figure 4.12. Influence of the Number of Initial Memory Cells Generated by Initialisation (Kt) 132
Figure 4.13. Influence of the Number of Initialisation Examples (Kts) 133
Figure 4.14. Change in predictive accuracy with changing class distribution 137
Figure 5.1. AISIID system flowchart 150
Figure 5.2. Example RWV and corresponding ITV 151
Figure 5.3. Generation of mini-document from webpage 160
Figure 5.4. AISIID visualisation of cell distribution 173
Figure 5.5. AISIID visualisation of interestingness distribution 173
Figure 5.6. Analysing AISIID spider technique 174
Figure 5.7. Example of good spidering technique (1) 175
Figure 5.8. Example of good Spidering technique (2). 176
Figure 6.1. Starting URL set 178
Figure 6.2. Chart showing the variation in population size for three different values of Ksuppress 181
Figure 6.3. Bar chart showing the mean number of iterations required for the system to reach a terminating condition, when varying $K_{suppress}$ 182
Figure 6.4. Chart showing variation in mean population affinity for two different population sorting methods. 184
Figure 6.5. Chart showing variation in mean population interestingness for two different population sorting methods. 185
Figure 6.6. Chart showing the maximum graph distance with increasing iteration number for two population ordering strategies 187
Figure 6.7. Chart showing the mean node distance with increasing iteration number for two population ordering strategies 188
Figure 6.8. Chart showing the mean cell position with increasing iteration number for two population ordering strategies 189
Figure 6.9. Histogram showing relationship between distance of page from start pages and number of cells finding pages at that distance. 191
Figure 6.10. Changing mean affinity and mean interestingness with distance from start pages. 192
Figure 6.11. Mean affinity of pages at different distances from the start pages (bars) with maximum affinity of these pages shown as lines. 193
Figure 6.12. Chart showing mean scores for each system under test 202

Figure C.1. Example distributions. 253
Figure C.2 Representation of one-tail (A) and two-tail (B) tests. 254

Chapter 1 Introduction

1.1 Motivation and Background

With the ever growing wealth information on the internet, effective tools for distinguishing between interesting and non-interesting material are becoming very important. The mining of textual data is a common web mining task, that is, performing data mining tasks over the text found on web pages. This type of mining, called web content mining is becoming increasingly necessary as finding information on the internet is almost impossible without automated assistance. While simple, well researched, information processing techniques have proven efficient at filtering or discovering *relevant* information, these traditional techniques currently not tailored for discovering *interesting* information.

The discovery of interesting information on the web is conspicuous by its absence in the research literature, yet it is a very real issue. While the results returned from a web page search will often be relevant, the user may be overwhelmed with an unmanageable number of search results. This thesis makes the assumption that a situation where fewer results are returned to the user, but each one contains information a user did not know, would be of greater use to a user than quantity of results. No current system is available to fulfil this role effectively. In addition to this, there exists a paradox with regard to current keyword based search techniques and the discovery of interesting information. Interesting information is surprising or unexpected to the user, but the user is required to specify search terms and these must be based on that user's existing knowledge. Therefore it is inevitable that a user's search for unexpected knowledge is hampered when existing knowledge is required to initiate a search. It is believed that this problem needs addressing.

Web content mining can supply a user with the information that he or she seeks but how should such a system be realised? There are a number of attributes of the web that make this a taxing task. Clearly the system must be both adaptable and robust. It must

be adaptable for two reasons, firstly the content of the web is forever changing and secondly so are the expectations of the user. It must be robust as web pages do not conform to a set template, they are full of spelling mistakes, advertisements, and huge amounts of irrelevant noise. Any web mining system that is to retrieve an acceptable set of results from a search must adapt to the conditions and ignore noise. This must be done while searching a vast space. However, the natural immune system exhibits many properties that are of interest to this area of web mining. Of particular interest is the dynamic nature of the immune system when compared with the dynamic nature of mining information from the web. Previous work has shown that a type of computer learning algorithm, an Artificial Immune System (AIS), has attributes that fulfil these criteria (de Castro & Timmis, 2002a).

The implementation of computer algorithms based on immune principles and components, or AIS, have become an increasingly popular machine-learning paradigm. Inspired by the mammalian immune system, AIS seek to use observed immune components and processes as metaphors to produce algorithms. These algorithms encapsulate a number of desirable properties of the natural immune system and are turned towards solving problems in a vast collection of domains (de Castro & Timmis, 2002a). There are a number of motivations for using the immune system as inspiration for both data mining and web mining algorithms which include recognition, diversity, memory, self regulation, and learning (Dasgupta, 1999). Being based on an AIS algorithm, by its very nature the system will preserve generalization and forget little used information. Thus giving a system such as this the ability to adapt to changing user preferences and underlying data.

The preceding paragraphs have expanded the title of this thesis - Web mining with an artificial immune system: focusing on the discovery of interesting information. Mining is the search for knowledge from some information resource, in this case pages of text stored on the web. It aims to enable a user to find interesting web pages which are hoped to be of more use to a user than simply relevant pages, and the method by which this goal may be achieved is in the use of AIS.

This thesis is concerned with the design, implementation and evaluation of two AIS algorithms. Both algorithms perform data mining tasks over sets of documents where each set has a notion of dynamics associated with it. In the first case, a passive filtering task is performed in which the system attempts to classify e-mail into two classes – those which will be of interest to a user and those which will not, based on previous experience. The second task is to discover, identify and rank pages drawn from the Web

that are considered "interesting" to a user. While at first glance it would appear that both e-mail and web pages are quite different, it could be argued that both are simply text documents but with certain metadata associated with them. E-mail has headers and web pages contain hyperlinks. Both of these are exploited during the investigations. Thus the main experimental part of the thesis takes the form of two case studies. Both serve to investigate AIS in domains currently unexplored in the literature, whist also contributing to the web mining field.

Firstly, the e-mail classifier AISEC (Artificial Immune System for E-mail Classification), investigates the behaviour of an AIS in a text mining scenario but with regard to continuous classification of interesting e-mail. Secondly the web mining system AISIID (Artificial Immune system for Interesting Information Discovery) is an investigation into the discovery of interesting information on the web.

1.2 Challenges

Investigations in this thesis involve two particularly challenging areas: 1) the identification of interesting information; and 2) the ability of a solution to perform well in a dynamic domain.

The notion of interestingness has been studied, for some years in the field of data mining. However, this has predominantly been undertaken in the field of classification and association rules, in order to discover rules (and therefore relationships) thought to be unknown to a user. Finding interesting information within a page of text is more removed from this than most would imagine. A page of text does not neatly fit into a template, unlike rule based classification or association where interestingness is gauged over a set of elements, all rules with the same form: "IF(x) AND(y) THEN(z)". Text is much more unstructured and tends to contain much more noise and irrelevant information than the structured dataset minded by conventional algorithms. This may be one reason why such a task has not been researched widely, the task seems insurmountable. Thus with little research regarding the discovery of interesting information from documents in the literature, the challenge of creating such a system becomes amplified as it is not possible to adapt another system.

Classification in a dynamic scenario is also a challenge. In the scenario investigated, mining data from the web, the data source is dynamic. This is in contrast to the more traditional data mining scenario where a model is trained once on a set of data and then set to classify unseen data. In this dynamic scenario the class unseen data should be assigned changes. In the web mining scenario this could be as user's opinions change.

This is a challenge because it requires classification algorithms to have extra layers of complexity. The algorithm must constantly update its internal representation of the class distribution and it must do this in a robust way so that it does not start making mistakes. How does the algorithm know whether a given data point was noise, or symptomatic of a change in the class of that data? Clearly the answer to this is not straightforward. Again, a more practical challenge also arises due to the relative sparseness of literature about dynamic classification (as opposed to the traditional classification over static datasets). However, it believed that an artificial immune system may already offer hope in this area. Its internal representation is already dynamic and so it is thought that it may naturally lend itself to this scenario.

Web mining, and more specifically web content mining is a challenge in itself. The book (Baeza-Yates & Ribeiro-Neto, 1999) lists a number of generic challenges associated with web mining. These challenges can be summarised as the points below. In an attempt to illustrate at this early stage that the chosen solution, an AIS, may be able to adequately meet this challenging web mining environment, each point is answered using general characteristics of artificial immune systems. These general characteristics have previously been discussed in (Dasgupta, 1999) and (de Castro & Timmis, 2002a).

- **Distributed data** – The data on the web spans countless computers. The immune system is naturally distributed. Just like the internet this distribution provides the system with disposability and diversity.
- **High percentage of volatile data** – New computers and data can be added or removed from the internet easily; likewise immune cells are constantly undergoing cell death and reproduction. The ability for both to cope with this situation shows both are adaptive, resilient and robust.
- **Large volume** – The size of the web is incredible and is constantly growing, making it difficult for systems to mine. The immune system too is made from countless numbers of cells. As each type of cell has a specialised function and works independently the system functions efficiently.
- **Quality of data** – The ease with which anyone may publish to the web can raise questions regarding its quality. Errors and omissions are common. The immune system however is noise tolerant, such that absolute matching is not required to trigger a response. Such noise tolerance is essential to an algorithm mining low quality data. The learning characteristics of the immune system are invaluable in this case. The immune system quickly

learns the new characteristics of invaders when they mutate, likewise a web mining system may learn to correctly classify documents even with errors, or conceptually, mutated words.

Thus, there are one general and two specific sets of challenges that need to be tackled for a positive outcome of this thesis. The general challenges of web content mining, and the more specific challenges of classification of changing data and identification of interesting information in a textual setting.

1.3 Goals and Contributions

The aim of this thesis is to investigate the application of AIS to web content mining, with regard to the identification of interesting information. There is a bias in the world of information retrieval to concentrate on maximising recall, precision and predictive accuracy. By contrast there are much fewer researchers working on the discovery of interesting knowledge and so greater potential for valuable research in this area. The problem identified and investigated by this thesis is therefore somewhat unexplored in itself. The identification of information that is interesting to a user in a web mining scenario is only investigated in one other reference as will be discussed later. The data mining algorithms produced work in somewhat dynamic domains, a scenario in which artificial immune systems are thought to demonstrate an advantage over other techniques.

This thesis will detail AIS solutions using inspiration from the literature. The quality of these solutions will be assessed by comparing against another suitable comparison system. Then the impact of this research, with regard to contribution to the fields of AIS and data mining, will be discussed. Contributions of subsequent chapters include the following:

1. A review of the literature regarding interestingness both in terms of its root domain, that of classification and association rules, and the identification of a comparison system. (Chapter 2)
2. A review of the AIS literature including previous instances of AIS being turned to text mining and web content mining tasks. (Chapter 3)
3. The design, implementation and evaluation of an artificial immune system for e-mail classification. (Chapter 4)
4. A description of the artificial immune system for discovery of interesting information. (Chapter 5)

5. An investigation into the characteristics of the web mining system for discovery of interesting information and a comparison between it and the comparison system identified earlier, with respect to subjective opinions of users. (Chapter 6)
6. A greater understanding of the comparison system and the identification of possible improvements, as revealed by the user tests. (Chapter 6)
7. Suggestions regarding the current understanding of interesting information discovery and the future of identifying interesting documents, based on the outcome of the user tests. (Chapter 7)

1.4 Thesis Outline

The structure of this thesis is as follows. Chapter 2 and Chapter 3 reveal the background information. This literature review serves two purposes. Firstly it serves to impart technical knowledge to the reader to allow for comprehension of the later chapters. Secondly it serves to provide evidence that the use of an AIS and the chosen problem domain are both justified in the context presented in this thesis. As this thesis covers the two somewhat disparate topics of data mining and artificial immune systems which are, at this stage in the thesis unrelated, the subjects are separated for the sake of clarity. For this reason, Chapter 3 does not follow in terms of content from Chapter 2.

Chapter 2 deals in detail with Data Mining. Taking a top-down approach, this begins by describing the most general data mining process before describing the more specific subjects of information retrieval, which then leads to text mining, which in turn leads to web mining. The notion of interesting knowledge is introduced with regard to information retrieval and a detailed review of the literature is performed.

Chapter 3 introduces Artificial Immune Systems (AIS) and combined with Chapter 2 forms a single background section. This chapter introduces AIS from a computing perspective. The referenced papers tend to contain the relevant immunological concepts which are often used as inspiration for the design of AIS. Therefore this chapter deals with the practices that engineers have used to exploit the biology and sets this in a re-useable and flexible framework model. Literature regarding the use of AIS for classification, e-mail classification and web mining is reviewed extensively.

Chapter 4 and Chapter 5 describe AIS algorithms for identification of interesting information in two text mining related scenarios, both of which take inspiration from the immunological process of clonal selection. Firstly Chapter 4 describes and evaluates AISEC, an Artificial Immune System for E-mail Classification. The chapter is

motivated by describing the reasons why an AIS is especially useful in the scenario of e-mail classification. It then describes the implementation of AISEC in detail using pseudocode. From this pseudocode the runtime complexity of the algorithm is analysed. AISEC is then tested both to investigate its quality at the task against a number of other classification techniques, and also to observe its characteristics as an algorithm.

Chapter 5 contains a detailed description of an AIS for web mining. Called AISIID (Artificial Immune System for Interesting Information Discovery), the algorithm works to discover interesting information on internet web pages. A number of aspects of AISIID are described at some length, many being novel to the realm of AIS. Then the AISIID algorithm is detailed in pseudocode.

Chapter 6 follows on from Chapter 5 by describing the results of tests performed on an implementation of the AISIID algorithm. An investigation into the characteristics of the AISIID algorithm is undertaken. This aims to ascertain whether certain aspects, such as the manner in which cells spread out over the search and population control are working as expected. A user study is then performed which has the aim of assessing the quality of the output in a subjective manner. The AISIID algorithm is compared with the only other known algorithm for ranking of web pages by interestingness, WebCompare, by allowing users to assess the output of both algorithms. Pages found by the popular search engine Google are also considered to give a baseline for analysis. Some unexpected conclusions on the relative merits of the three page ranking techniques are drawn from this study and it is thought that these conclusions make an important contribution to the field.

Chapter 7 contains concluding remarks and a description of future improvements and directions for both AISEC and AISIID. The chapter reflects on and summarises the results of the previous chapters. This final chapter also offers some ideas of future work for each system that it is hoped may be picked up and acted upon by an interested researcher. The observations from the previous chapter are revisited and interpreted providing a useful insight into the field of data mining, in particular the discovery of interesting information from the web. Finally the thesis concludes by offering some thoughts on the use of intelligent systems for web mining in the long term.

1.5 Publications

The following publications were written during the research for this thesis:

- Secker, A., Freitas, A. A., & Timmis, J. (2003). *AISEC: an Artificial Immune System for E-mail Classification.* Congress on Evolutionary Computation 2003 (CEC2003), Canberra, Australia. Vol. 1. IEEE, pp. 131-138

- Secker, A., Freitas, A. A., & Timmis, J. (2003). *A Danger Theory Inspired Approach to Web Mining.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 156-167

- Secker, A., Freitas, A. A., & Timmis, J. (2005). Towards A Danger Theory Inspired Artificial Immune System for Web Mining. In Scime, A. (Ed.), *Web mining: Applications and Techniques* (pp. 145-168). London: Idea Group Publishing.

Chapter 2 Data and Web Mining

This chapter gives a comprehensive overview of data mining topics which will be encountered during the remainder of the thesis, focusing on both data mining and web mining concepts and methods.

2.1 Data Mining and Knowledge Discovery

Data mining has been defined as:

Data mining is the analysis of observational data sets to find unexpected relationships and to summarise the data in novel ways that are both useful and understandable to the data owner (Hand et al., 2001)

Data mining seeks to take large amounts of raw data and transform it into a form humans can easily understand and ultimately use for decision making. In general, data mining has two related streams for very broad kinds of tasks, those of modelling and pattern discovery. Modelling involves the summarisation of large structures in the data. Examples of such modelling tasks are:

- Segmentation analysis
- Regression analysis
- Time series decompositions

Techniques used to solve modelling tasks tend to be mature as they have been developed over many years, and also tend to be statistical in nature. The second stream of data mining is that of pattern discovery. Techniques for this kind of task generally detect certain local features within a data set. In this context, local means that few data points are involved, with these tending to summarise a small data subset compared with the vast amount of data summarised by modelling. (Hand & Zhang, 2005) describes some techniques for discovering these local structures amongst large repositories of data.

As discovered relationships or patterns tend to be interpreted by a human in the context of a particular task, it can be stated that data mining is the principal step towards *knowledge discovery*. Other steps in this process may include data cleaning and preparation before data mining, and post processing after the data mining stage. The full knowledge discovery process tends to be interactive. (Smith, 2005) explains the interactive nature of the process and also outlines a semi-automated pre-processing task.

The knowledge discovery process is about asking the right questions to get the right answers and therefore a domain expert should be involved at every step to ask those questions. This process can be summarised as follows (Fayyad et al., 1996):

- Data cleaning and interrogation
- Data selection and transformation
- Data mining
- Evaluation
- Knowledge presentation

The first two steps are pre-processing stages, whilst the last two are post-processing stages. The data mining step in the middle is the part of the knowledge discovery process that actually extracts the knowledge from data and can draw on strategies from machine learning, statistics and other areas to solve a given data mining task. (Freitas, 2002a) describes three desirable properties discovered knowledge should include. These are that the knowledge should be a) accurate, b) comprehensible and c) interesting, although it is noted that the relative importance of each of these is dependent upon both the domain and the user. Generally, where applications involve strategic decisions to be made by a human user, rather than a machine, the comprehensibility of the discovered knowledge is important. Often the knowledge will be used to support user decisions, and if that knowledge is presented in a manner not understood by that user, the decision making process can become impeded. In addition, if a user cannot validate the result of a data mining algorithm the process of refinement and iteration may become frustrated.

As ever increasing amounts of data are stored in repositories such as databases, there is an ever increasing desire to leverage the otherwise "hidden" patterns in the data for commercial gain or otherwise. The actual step of data mining may use one or more data mining algorithms. The actual task of the data mining algorithm will change depending on the general task for the knowledge discovery process, but there are two main types of task: those of classification and clustering.

In essence, clustering can be regarded as the process of partitioning data into one or more groups maximising intra-cluster distances and minimising inter-cluster distances. That is, data records within one cluster tend to be more alike than data records not in that cluster. Clustering is not the concern of this thesis and so the interested reader is directed towards the literature such as (Merkin, 1996).

2.1.1 Classification and Prediction

In its simplest form, the goal of the classification task is to assign a class to an instance of a data set based on a set of characteristics (attributes or features) expressed by that instance. Therefore each instance of the data set consists of two parts: a single *goal* attribute to be predicted and a collection of other attributes called predictor attributes, the values of which are used to determine a predicted class (value of the goal attribute). For this process to begin, it is necessary to first give the prediction system a number of examples, collectively called a *training set,* where both the values of the predictor attributes and the goal attribute are known. From this it is the duty of the system to infer the relationship between the predictor attributes and the goal attribute. Hence, when an example is presented with the predictor attributes known and its class unknown, the system may apply the relationships it has previously learnt and predict the value of the goal attribute. This learning stage is called the *training phase* of the classification algorithm. These relationships may be expressed in the form of decision trees (Quinlan, 1987; Witten & Frank, 1999a; Witten & Frank, 1999b) or prediction rules, although a detailed discussion of these and other representations is beyond the scope of this thesis.

At this time, the classification system is able to run on unseen data where the true class of an instance is unknown. However, before this occurs it is usual to take the system through a *testing phase*, the aim of which is to discover how well the system is performing by measuring the *predictive accuracy* of the learnt relationships. This is done using another set of instances, called the *test set,* where the value of the goal attribute is known to the user but unknown to the system for the purposes of classification. It is essential here that no data instances are shared between the test set and the original training set, so is usual practice to partition the dataset into a training set and a test set before the training begins. The system then applies the learnt relationships to the instances of the test set to predict each instance's class. After predicting this class, the system is then allowed to see the true class of that instance. If the system predicted the correct class it is counted as a correct classification attempt. As a measure of the quality of the learnt relationship, the predictive accuracy may be

computed. This is simply the number of times the system predicted the correct class divided by the total number of classification attempts (and therefore the number of instances in the test set). The higher this predictive accuracy, the better the system is working. A 100% "predictive" accuracy could trivially be achieved over the training data by implementing a lookup table, but this strategy would be likely to achieve a low accuracy when confronted with unseen data instances, such as those in the test set.

In (Freitas, 2000; Freitas, 2002a) the author discusses how the classification task as described above is ill defined and non-deterministic in the sense that once a relationship has been inferred in the training set by the system between a set of predictor attributes and a goal attribute, this relationship cannot be guaranteed to continue to provide high predictive accuracy on unseen data. In (Freitas, 2000) the author states "*we are essentially using data about the past to induce rules about the future... Clearly, predicting the future is a non-deterministic problem*" (p.66). The author argues that there may be a virtually infinite number of hypotheses (e.g. candidate rules) consistent with a training set. Hence when a classification algorithm is searching for the best hypothesis (model), in addition to taking into account the hypothesis' consistency with the data, it must have an inductive bias that favours one hypothesis over another based on another criterion. In particular humans, data analysts and users, as well as good rule induction algorithms have a tendency to be biased towards the simplest one. However, there is no guarantee that this simplest hypothesis will make correct predictions over the (as yet unseen) test set (Domingos, 1998).

Below is a discussion of three classification techniques, namely nearest neighbour (or instance based), Bayesian and rule-based classification. Among the large numbers of classification techniques available in the literature, these three were chosen because they are the most relevant for this thesis and they will be referred to in subsequent chapters.

2.1.2 Instance Based (Nearest Neighbour) Classification

The *K-nearest neighbour* (KNN), *instance based* learning (IBL) or *lazy learning* approach to classification (Aha et al., 1991; Aha, 1992; Lee, 1994) is of particular significance to this study as many parallels may be drawn between it and attributes of an Artificial Immune System (AIS), the topic of the following chapter. Most notably the reliance on representation, distance metric and the simplicity with which a KNN can update based on changes in data, and so support incremental learning (Aha, 1992).

In essence it uses the available training set (or a subset of that) to classify new data instances, no explicit training stage is needed. To classify these yet unseen instances,

the *K-nearest* training instances (where K is a user defined parameter) are retrieved and the test instance assigned the class of the majority of those K training items. KNN algorithms may also be referred to as one-nearest neighbour, 1NN, algorithms when K=1. In the above description the word nearest is in italics because nearest does not necessarily imply proximity in Euclidean space, but rather proximity in solution space based on some appropriate distance measure. Indeed, *Euclidean distance* may be employed to measure the similarity between two data instances if appropriate, but this distance measure may not be the best for a given problem. Different distance measures have different inductive biases making them suitable to different kinds of data sets. (Saltzberg, 1991) contains a discussion of such distance metrics and presents a comparison of these over three data sets. Of specific interest to this thesis, the distance metric of *cosine distance* (section 2.2) is particularly common when performing distance measures between text documents.

In addition to introducing a particular bias, the representation used to store individual instances may require one distance measure over another. Hamming distance is only applicable over binary attributes for example. Discussion of representational issues for KNN classifiers may also be found in (Kibler et al., 1989; Saltzberg, 1991).

Many strategies may be employed to improve the performance of nearest neighbour algorithms. Firstly, each attribute in data instances may be appropriately weighted based on a notion of relevance to the classification. A highly relevant attribute should have a large influence on the classification, although an effective application of weights may present a complex optimisation problem in itself. If this is not done specifically, the data should at least be normalised such that one attribute with a large range or high values does not dominate another attribute with a small range or small values. The second improvement may come from reducing the size of the training set, while preserving the classification accuracy. This will improve the time taken to perform the classification as, in general, a data instance must be compared with every training instance for a classification to be performed. (Zhang, 1992) describes how a concept may be represented by a small number of *typical instances* of that concept. The nearest neighbour algorithm may therefore only need to compare a test instance with this vastly reduced set of training instances, thus reducing the time taken to classify an instance and, as generality is preserved, even increasing predictive accuracy over some data sets.

The third way to improve performance is reducing the attributes used for classification, preserving only the most relevant attributes via an attribute selection

method. In the case of (Rozsypal & Kubat, 2001) the authors use a genetic algorithm to both select typical instances and select the most relevant attributes.

It should be noted that Memory Based Reasoning (MBR) (Stanfill & Waltz, 1986; Creecy et al., 1998) has certain commonalities with Instance Based Learning (IBL), as do Case Based Reasoning (CBR) systems (Liao et al., 1998). Although these topics will not be covered further in this thesis.

2.1.3 Naïve Bayesian Classification

Naïve Bayesian classifiers (Weiss & Kulikowski, 1991; Mitchell, 1997; Friedman & Kohavi, 2002) are a very popular technique used for classification and one that has been effective over many years.

As described previously, in the classification task of data mining, the goal is to assign a class to an instance based on the values of a number of predictor attributes. In the naïve Bayesian scenario, the probabilities of an instance belonging to each possible class are estimated based on the training data, and the instance is assigned the class that is most probable. As Bayesian classifiers have roots in statistical mathematics, they possess properties that are mathematically provable, and therefore desirable for many applications. One of these is it can be shown that, in theory, a Bayesian classifier will reach the smallest possible classification error given a sufficiently large training set. Although in practice this may not be the case due to the need for unrealistic simplifying assumptions described later. In addition to this, probabilistic methods may be employed to deal with missing values and asymmetric loss functions. That is, situations where the cost of misclassifying examples of one class may far outweigh the cost of misclassifying examples of another. For example, classifying an important e-mail as spam and removing it is a lot more harmful than to allow spam e-mail into the user's inbox (Diao et al., 2000).

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \tag{2.1}$$

$$v_{mp} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2 ... a_n) \tag{2.2}$$

$$v_{mp} = \arg\max_{v_j \in V} \frac{P(a_1, a_2 ... a_n \mid v_j)P(v_j)}{P(a_1, a_2 ... a_n)} \tag{2.3}$$

$$v_{mp} = \arg\max_{v_j \in V} P(a_1, a_2 ... a_n \mid v_j)P(v_j) \tag{2.4}$$

The Bayes theorem (Manning & Schutze, 1999) is the cornerstone of Bayesian learning. Equation 2.1 to Equation 2.4 describe the steps taken to derive the equation used for the naïve Bayesian classifier from the Bayes theorem and an equation to return the most probable class given a set of features. As described by (Mitchell, 1997), the probability of observing hypothesis h given the training data D may be given by Equation 2.1. In Bayesian learning, the most probable class v_{mp} is assigned from a finite set, V, based on a set of attribute values $<a_1,a_2...a_n>$ as described by Equation 2.2. The Bayes theorem, Equation 2.1, and the equation to determine the most probable class Equation 2.2 can be combined as shown in Equation 2.3, to produce Equation 2.4, since the denominator of Equation 2.3 is just a normalization factor which can be ignored without affecting the choice of class V_{mp}.

In Equation 2.4, $P(v_j)$ can be estimated simply by counting the frequency with which each class appears in the training data. However the first term is a lot harder to determine. In practice, it is not necessary to see every possible instance in the problem space a number of times in order to provide reliable estimates. The naïve Bayes classifier introduces the assumption that attribute values are conditionally independent as shown in (Kononenko, 1991; Pazzani, 1996). Therefore the probability of observing $a_1,a_2...a_n$ is the product of the probabilities of observing each attribute independently. This results in the naïve Bayesian classifier as defined in Equation 2.5.

$$v_{NB} = \underset{v_j \in V}{\text{argmax}} \; P(v_j) \prod_i P(a_i \mid v_j) \qquad \textbf{2.5}$$

The terms in Equation 2.5 are usually calculated using frequency counts of attribute values over the training data. However, it is possible that an attribute value that does not occur in any example of a given class during training is encountered during testing. When calculating the probability of this class, assuming the frequency count to simply be 0 (i.e. $P(a|b) = 0/100 = 0$) would rule out this class entirely, as this zero term results in the calculated probability for this class always evaluating to 0. A number of methods have been suggested for substituting a suitable probability for this value, although each is associated with its own form of bias. Some implementations simply ignore this term, but a common and straightforward strategy is to replace the probability with a small, non-zero number. Examples of this would be replacement with $1/n$, where n is the number of training examples, which has the advantage that this represents the increasing certainty that this element must have an almost-zero value with the increasing size of the training set. The probability may also be replaced by $1/m$, where m is the number of

attributes. This strategy is found employed later in this thesis. A worked example of naïve Bayesian classification, used for classifying documents, can be found in (Mitchell, 1997).

2.1.4 Rule Induction

Rule induction algorithms (Witten & Frank, 1999a) have been around for many years and their popularity during this period cannot be overstated. One of the reasons why they may be used so widely is the clear and concise way they may summarise discovered knowledge. The task of a rule induction algorithm is to capture knowledge from a data set in the form of IF-THEN rules as demonstrated in Figure 2.1:

```
IF (condition1) AND (condition2) ... (condition) THEN (consequent)
```

Figure 2.1. General layout of an IF-THEN rule

The antecedent of a rule is a conjunction of a number of conditions, each containing at least one predictor attribute. This conjunction will predict the value of a goal attribute, the consequent, which is a value drawn from a finite and discrete set of classes. Classification rules normally use either one of two types of predictor conditions in their antecedent, as shown in Figure 2.2. The first type involves an attribute and a value of the domain of that attribute. The second type involves two attributes. These two types of condition are often called propositional and first-order logic conditions respectively.

```
1. (gender = male) or (age > 26)
2. (income > expenditure)
```

Figure 2.2. Two common forms for conditions in an IF-THEN rule

Generally, a rule induction algorithm consist of two stages, those of rule construction and rule pruning. Rule construction typically consists of a greedy procedure where a rule is initialised with an empty antecedent and the algorithm adds one condition at a time to the rule until a stopping criterion is satisfied. At each step the condition to be added is chosen by using a certain evaluation function. Rule pruning is an important procedure to prevent overfitting. This can be plainly seen given the fact that a rule induction can achieve an accuracy of 100% over a training set simply by constructing one unique rule per training record, although it is intuitive that this would not be useful when run on unknown class data in the test set. Pruning therefore

eliminates the rules deemed to be too specific, generalising the rule set and allowing it to work better over data unseen during training. Generalisation is important to all classification algorithms, a short discussion on this can be found in section 2.1.5.

It is beyond the scope of this section to describe rule induction algorithms in detail but for the interested reader a popular method is C5.0 (and its predecessor, C4.5). There has also been work in the field of rule induction with biologically-inspired algorithms for example (Freitas, 2002b; Parpinelli et al., 2002; Zhou et al., 2003).

It is worth noting that classification rules rather than *association rules* have been described so far in this section, and throughout the rest of the thesis. Association rules seek to summarise the underlying data by representing sets of items that frequently occur together in the same transaction. It is a well defined and deterministic task, where the accuracy of each discovered rule can be calculated using the measures of *support* and *confidence* (Leng, 2005). Such measures however, are not measures of predictive accuracy because association rules do not involve prediction. In contrast, classification rules involve a non-deterministic task of prediction as discussed earlier (Freitas, 2000).

2.1.5 Overfitting and Underfitting

Overfitting and underfitting refers to the level of generalisation of the relationships generated by a classification system during the training phase. Overfitting is described in (Quinlan & Cameron-Jones, 1995) as "*the construction of a theory tailored to the* [training] *data that has a high but misleading apparent accuracy*". It is accepted that the training data will never be a perfect, unbiased representation of the test set data. Spurious or "fluke" relationships not present in the test set will almost always be present in this training set and vice versa. Learning these fluke relationships would decrease the classification accuracy of the system when run on test data, a phenomenon called overfitting, the converse of which is underfitting. When underfitting occurs, the system has assumed too many relationships it has discovered are due to these fluke relationships. These relationships have therefore been discarded (or not discovered in the first place), but if a relationship was genuinely representative of the data then without it, the predictive accuracy on the test set will be lower than it could have been. Generally a reduction in training results in underfitting, whereas training more than necessary results in overfitting (Figure 2.3). One goal in the production of classification systems is an automated balance between overfitting and underfitting. The amount of training needed to strike this balance (point A in Figure 2.3) will differ from dataset to dataset and from system to system, and so too will the gradients G_1 and G_2.

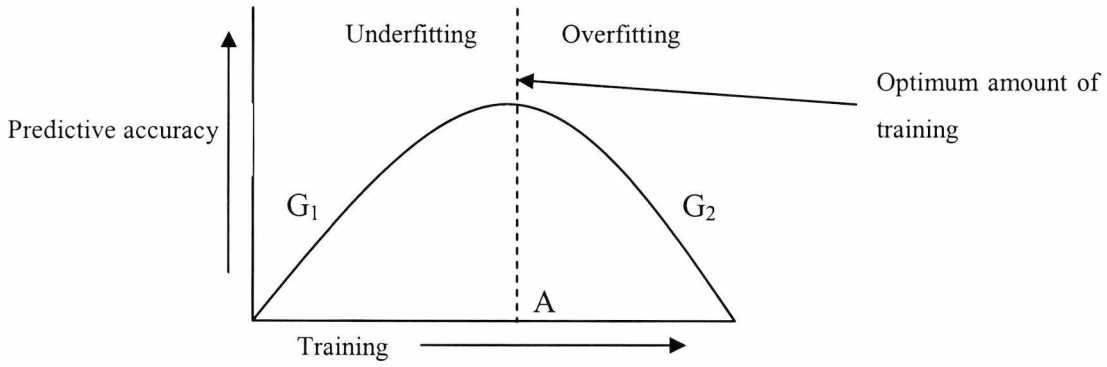

Figure 2.3. Example chart describing the relationship between amount of training and predictive accuracy over a test set

It has been noted in the literature that generalisation introduces an inductive bias – "*any overfitting avoidance strategy amounts to a form of bias and as such may degrade performance instead of improving it*" (Schaffer, 1993). As a bias it follows that the effectiveness of overfitting avoidance strategies may be domain dependent, and the most appropriate overfitting avoidance strategy for the target domain should be selected in a case-by-case basis. An illustration here is the use of Occam's razor[1]. That is, the use of the simplest model should in turn lead to the greatest generalisation and therefore, apparently, the greatest predictive accuracy. However, this standpoint is critiqued in (Webb, 1996) based on experimental evidence, with alternatives being discussed in (Webb, 1994). In addition, (Domingos, 1998) argues that a wrong interpretation of Occam's razor has often been used in data mining.

2.2 Text Mining and Information Retrieval

While more "traditional" data mining tends to work with numerical or categorical records drawn from databases, it is quite reasonable to discover patterns and relationships within and between text documents (Feldman, 2002; Weiss et al., 2005). The discovery of such patterns and relationships is the area of text mining.

One large portion of the text mining subject area is concerned with *information retrieval*. In its basic form, Information Retrieval (IR) is mainly concerned with

[1] In its broadest form, Occam's Razor states that entities should not be multiplied without necessity, hence, when multiple explanations are available for a phenomenon, the simplest explanation should be preferred.

retrieving one document from a set of documents, rather than inferring relationships. This is a subject area that has been researched for several decades. Indeed, it was a mature subject area long before the advent of enormous text repositories such as the web; see (Faloutsos & Oard, 1995) for a survey. Even though some information retrieval systems were constructed before the web existed, many essential aspects of information retrieval systems are still used extensively when mining the web. Therefore, a number of basic IR concepts are examined in the following subsection including document pre-processing and evaluation, which will be used later in this thesis.

2.2.1 Text Pre-Processing

To begin a typical IR procedure, the document must first be pre-processed. This is primarily to reduce the document into a form that is more efficient to represent and process further. A number of the pre-processing tasks aim to reduce the amount of data that must be stored when storing a document or collection of documents, whilst at the same time keeping any loss of information to a bare minimum.

To begin, the text may be *tokenised*. This procedure involves splitting the text into words and removing punctuation, thus transforming a document of words, numbers and other symbols to an ordered set of tokens. Identifying exactly what is a word, and therefore a candidate token varies from algorithm to algorithm. Usually a block of characters surrounded by spaces becomes a token. When dealing with punctuation, some algorithms will combine hyphenated words such as "e-mail" into one word "e-mail" but some will produce two tokens "e" and "mail". Likewise when dealing with numbers, some will discard tokens comprised entirely of numbers such as "13" but keep tokens comprising of numbers and letters, i.e. "13th". Some may discard the numbers from the beginning or end of a token, so "13th" would become "th". Whatever tokenising an algorithm performs it is important that no information that would otherwise be useful is, discarded.

After tokenisation, the removal of *stopwords* from the set of tokens may be wise. Stopwords are the set of words, such as function words or connectives, that bring no meaning to a document such as "a", "it", "is", "the", "on" etc. These types of word are usually the most common throughout a document and so removal of these will have the effect of greatly reducing the number of words in the token set. There are two important points with this procedure. If, at a later time, the proximity of words is important, a placeholder must be introduced to demonstrate a word has been missed out. Hence phrases such as "bacon and eggs" can still be retrieved. It is also important to use a

correctly constructed stopword list. An example here is that of the word "can". In one context, such as the phrase "I can dive", it is a stopword whereas in the phrase "can of bitter" it most certainly is not. Therefore the word "can" should not be used as a stopword in this instance.

After stopword removal stemming may be applied to the remaining tokens. This procedure involves reducing a word to its base word, or stem, thus further reducing the number of tokens and increasing efficiency. For example, the words "drinking", "drinks" and "drinkers" may all be reduced to the single word "drink" while retaining almost all of the original meaning. Again, like stopword removal, stemming must be handled with a certain amount of care. The words "universe" and "university" could both become reduced to "univers". However, modern stemming algorithms tend to be mature and so are rarely susceptible to mistakes. One well known example is the Porter stemming algorithm (Porter, 1997), which uses a set of rules applied to a word in a sequence to stem. Recent research has indicated that in some situations stemming can increase the accuracy of document classification algorithms (Benbrahim & Bramer, 2005).

2.2.2 Computing Similarity Between Documents

A popular representation for documents is the boolean model where each document is represented by a vector where each element of this vector is a boolean value indicating the presence or absence of a word (token) in a document. When attempting to retrieve documents from a corpus using a query and the boolean model, all documents containing one or more tokens (or terms) can be reliably retrieved using a straightforward query. However, after the retrieve operation has completed, there is little information to rank the results before presenting these to the user. All results are equally valid because all results contain all search terms, and the boolean model does not store information regarding how many times these occur.

One way to improve this situation would be to store a count with each token, indicating how many times that token occurs in the document. Naively then, when a query is submitted, the results may be ranked taking into consideration the number of occurrences of the token in the document: the higher the frequency with which a word occurs in a document, the higher the relevance of that document. This count is called the term frequency and is commonly stored with the token. However, simply using this term frequency for ranking is prone to errors for two reasons:

1. Different documents are different lengths. By its nature a shorter document will on average contain smaller term frequencies for its terms compared with a longer document, thus biasing the ranking towards longer documents.
2. Some words are more common than others. Within documents some terms naturally occur with a much higher frequency. Thus, when searching for two terms in a query, one term may completely dominate the other biasing the result in favour of the common term.

A solution to point 1 above is to normalise the term frequency by the document's length, thus practically eliminating this bias as an issue, although it still leaves point 2. Point 2 can be eliminated by taking into account the frequency of the term in the collection of documents as a whole. Thus, if the term is common in one document but uncommon in the collection of documents, it can be thought to be more characteristic of that document; whereas if the frequency of the term is high in both the document and the collection as a whole it is reasonable to assume the word generally occurs with a high frequency and so, when ranking calculations are performed, should be penalised accordingly.

The two ideas of normalising term frequency by document length and taking into account the frequency of the word over the set of documents is a mainstay of IR called Term Frequency/Inverse Document Frequency, or TFIDF. There are a number of weights that can be associated with an individual term in a vector representation but by far the most common is this TFIDF metric. As the notion of weighting words with TFIDF is important in text mining, some details are presented below regarding the calculation of a TFIDF value for a term and, when a document is represented by <term, weight> pairings, how it is possible to determine a measure of document similarity.

Consider a single document which may be represented by a set of <term, weight> pairs. A document can be thought of as representing a vector in n-dimensional space, where n is the number of terms in the document (thus, document space has n axes) and that term's weight corresponds to the value of co-ordinates of the document in the corresponding dimension. This is therefore known as the *vector space model* of document representation. To compute the term weight of term t in document d in terms of TFIDF, Equation 2.6 can be used.

$$\texttt{weight}_{t,d} = TF_{d,t} \times IDF_{D,t} \qquad \textbf{2.6}$$

Term frequency, $TF_{d,t}$ is a normalised score of the number of times term t occurs in document d. The normalisation is commonly done by the number of terms in document d to normalise this figure by document length, see Equation 2.7, or by the maximum frequency of any word found in the document. In Equation 2.7, $F_{d,t}$ is the raw count of the number of time term t occurs in document d.

Inverse document frequency ($IDF_{D,t}$) is derived from document frequency. Document frequency (DF_t) is the number of documents in collection D in which term t occurs. The inverse of this is required thus penalising terms for occurring in many documents in the collection. Both TF ad IDF are commonly damped using logarithms, where the exact nature of the damping term ($\log_2$, $\log_{10}$ etc.) is implementation dependent, thus giving the substitutions for Equation 2.6, shown in Equation 2.7 and Equation 2.8 respectively.

$$TF_{d,t} = 1 + \log\left(\frac{F_{d,t}}{|d|}\right) \quad \textbf{2.7}$$

$$IDF_{D,t} = \log\frac{|D|}{DF_t} \quad \textbf{2.8}$$

Once TFIDF values have been determined for each term, it is possible to determine the proximity of one document to another in multi-dimensional space. The simplest way to do this is by the use of Euclidean distance. The well known formula $a^2 + b^2 = c^2$ generalises to n dimensions and so it is possible to compute the distance between two document vectors (x and y) in Euclidean space using Equation 2.9.

$$\mathtt{dist}_{x,y} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \textbf{2.9}$$

One, possibly more common way of determining measure of distance between two documents in vector space is using the cosine measure of distance. In general, when comparing two n-dimensional vectors, or documents, the distance between them can be calculated by Equation 2.10 where x and y are documents and x_i and y_i are the weights of word i on document x or word i in document y respectively.

$$\mathtt{dist}_{x,y} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad \textbf{2.10}$$

2.2.3 Assessing Quality

With regard to information retrieval, there are a number of objective and subjective measures of quality for a classification system. A high score in one criterion does not necessarily mean a high score in the other while the criterion to maximise will change from domain to domain.

To begin with objective measures, the traditional measures of predictive accuracy, precision and recall are common. Given a class of document, all documents with content of that class are referred to as "positive documents", all others negative, no matter how many different classes there are. The accuracy of a predictive system over a test set is given in Equation 2.11:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2.11}$$

Note that predictive accuracy must only be computed over a test set, no training data can be involved in this calculation. For information retrieval applications there is usually a large amount of negative data, and therefore a classifier may achieve a high accuracy simply by predicting all records as belonging to that negative class. It is therefore desirable to introduce metrics that examine the errors made by the classifier. Precision and recall are such measures:

- Recall is the proportion of relevant results returned with respect to the total number of relevant results.
- Precision is the proportion of the results returned that are actually relevant to the search query.

$$\text{Precision} = \frac{\text{Number of correct positive predictions}}{\text{Number of positive predictions}} \tag{2.12}$$

$$\text{Recall} = \frac{\text{Number of correct positive predictions}}{\text{Number of positive documents in the set}} \tag{2.13}$$

A perfect IR system would maximise both recall and precision, thus the IR system would retrieve everything relevant to the search, and likewise everything it retrieves is relevant. As is often the case there is a trade off between precision and recall. To increase recall more documents can be retrieved, but this increases the risk that the documents are retrieved will be irrelevant, thus reducing precision. Because document collections are often large, a high precision is typically indicative of a better classifier,

that is the positive decisions are usually correct. The class distributions in typical IR scenarios are often skewed, so too are the costs associated with classifying or misclassifying positive or negative examples. For this reason a confusion matrix, shown in Figure 2.4, may be desirable for illustration of classifier performance.

The confusion matrix consists of four cells:

1. **TP (True positive)**: The number of documents classified as positive that were positive.
2. **FP (False Positive)**: The number of documents classified as positive but were negative.
3. **TN (True negative)**: The number of documents classified as negative that were negative.
4. **FN (False negative)**: The number of documents classified as negative but were positive.

		Actual Class: Positive	Actual Class: Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 2.4. Example Confusion Matrix

Other than the objective measures of classifier performance above, the quality of a classification output can also be evaluated by more subjective measures. That is, any output ideally should have:

1. High predictive accuracy (described above)
2. High comprehensibility
3. High degree of perceived interestingness

It is important that a user is able to question why a decision was made by the algorithm. This allows the user an extra level of information, other than the algorithm output. At one end of the spectrum, rule-based classifiers tend to produce comprehensible output and some objective measures exist to evaluate the comprehensibility of rule produced. Fewer rules and a shorter rule length tend to increase comprehensibility for the user. At the other end, the output of algorithms such

as neural networks tend to have low comprehensibility, as it can be almost impossible to illustrate to the user why a particular decision is made.

Interestingness is another characteristic that discovered knowledge should possess. An in-depth look at knowledge interestingness is contained in section 2.5.

2.3 WordNet

WordNet (Fellbaum, 1998) is an electronic repository of both words and phrases. In the context of this thesis it has been used to help text mining systems and so is examined here. WordNet is used extensively in AISIID, the system detailed in Chapter 5 and Chapter 6. Just as important as the words themselves in WordNet are stored relationships between words. In its current version – version 2.0 (available online at (WordNet, 2004), WordNet contains over 144,000 unique words/phrases. WordNet is described as an attempt to map the human understanding of words and the relationships between them. In WordNet all words (and phrases) are tagged with their part of speech (POS), noun, adjective, verb, etc. thus allowing an efficient lookup mechanism for any word's legal parts of speech.

As a brief aside, part of speech tagging is an important concept in furthering the accuracy of information retrieval systems. (Manning & Schutze, 1999) contains a chapter dedicated solely to part of speech (POS) tagging. Two popular POS taggers are the Brill tagger (Brill, 1992) and QTAG (Tufis & Mason, 1998). Other than containing POS information, each WordNet word also belongs in a set of synonyms, words that all mean a similar thing, called a *synset*, for example the sets {police man, police officer} or {buy, purchase}. There are a number of relationships defined between synsets. Of particular note are the following:

1. Generalisation. Called a *hypernym* relationship, Y is a *hypernym* of X if every X is a (kind of) Y. For example a "robin" is a "bird" which is an "animal". Not all hypernyms share the same root in the hierarchy. Instead there are currently 25 unique hypernym roots. Sometimes it is necessary for all nouns to share a common root, in which case a virtual root may be defined containing all real roots as immediate children.
2. Specialisation. Called a *hyponym* relationship. Y is a hyponym of X if every Y is a (kind of) X. Therefore a "bird" is a hyponym of "animal". Note that due to the quirks of language the hypernym/hyponym relationship is not necessarily symmetrical.

3. Opposites. Called an *antonym* relationship. For example "hot" and "cold" are antonyms of each other and this relationship is symmetrical.

Some lesser-used WordNet relationships include the meronym/holonym relationship. That is, 'X' is a holonym of 'Y' if Y is a part of X. The converse is true of a meronym. Therefore "tree" is a holonym of "bark", "trunk" and "branch". It follows therefore that the latter three are meronym of "tree".

It should be noted however that not all relationships are available for all parts of speech. For example, the hyponym/hypernyms relationship is defined only over nouns and the antonym relationship is defined only over adjectives. These relationships defined over any other parts of speech make no sense in language.

As well as these relationships, a single word, for example "borrow", may be stored in many forms such as "borrows" and "borrowing", thus allowing authors to use WordNet to perform a certain amount of simple stemming if needed.

WordNet has proven to be a useful tool in text mining research, allowing authors of algorithms flexibility and computational intelligence in the textual domain. Of particular note is (Voorhees, 1998), in which WordNet is used to improve the performance of an information retrieval system and the paper (Holden & Freitas, 2004), in which the authors describe a web page classification system using an ant colony algorithm for classification but relying heavily on WordNet for processing of web pages. Nouns in the text are identified using their parts of speech as recorded in WordNet. The authors hypothesise that the nouns on a page can capture a significant proportion of the subject or context of that page and so the page is reduced to a series of nouns. WordNet is then used to generalise the nouns contained in the page using the hypernym semantic relationship, thus both reducing the number of and increasing the generality of the attributes describing a page.

Other than classification, WordNet has been used to aid the clustering of text documents. In (Hotho et al., 2003) the authors state that representing documents as a set of words is insufficient for clustering as it ignores the conceptual similarity between the words. Their algorithm uses a standard hierarchical clustering technique combined with WordNet to cluster news articles with generally good results, although it was noted that word sense disambiguation must be implemented for the results to improve. Given that a single word in language can have more than one meaning (polysemy) or a set of words can have a single meaning (synonymy). Tagging a polysemous word with the correct sense is the task of word sense disambiguation and this was investigated in

(Voorhees, 1993) and (Leacock & Chodorow, 1998). In the former, the author uses a number of word relationships such as antonym, hypernym/hyponym, and meronym/holonym to disambiguate the senses of nouns, although this approach was shown to work poorly compared with the use of word stems.

In (Silla et al., 2003) the authors use WordNet to enhance the automatic detection of topics in text documents and summarise that document, the objective being to shorten a given text to a summary such that reading time for a human user is reduced but the main concepts of the text are still kept. Nouns as identified by WordNet and the generalising hypernym relation is used in a similar manner to (Holden & Freitas, 2004), discussed previously. Text summarisation using WordNet was also investigated in (Chaves, 2001), although no experimental results were available.

One research instance relevant to this thesis is the use of WordNet to evaluate the novelty (section 2.5) of rules generated from text. (Basu et al., 2001) used WordNet to evaluate the novelty of rules mined from a set of descriptions of books from Amazon (Amazon, 2005). Once again, using the hypernym relation between synsets in WordNet, the investigation seeks to use the notion of hypernym hierarchy to assess rule novelty. A notion of semantic distance between two words is used, where the semantic distance is the shortest path connecting those two words using the hypernym hierarchy in WordNet. Recall, all nouns in WordNet are part of the hypernym tree hierarchy and a common root can be constructed between all natural roots. The semantic distance between words in the antecedent of the rule and that in the consequent can be computed and compared with the average semantic distance between all other words in the hierarchy. This semantic distance is defined as a weighted shortest path connecting two words using the hypernym relationship in WordNet. The rule generation algorithm DiscoTEX is run over a collection of documents and the average semantic distance between the words in the antecedent and the consequent of the rule is computed. The greater this distance the less the words are related and therefore the higher the novelty of the rule. The example given is that IF (beer) then (nappies) is naturally more of a novel rule than IF (beer) THEN (crisps), as the deepest common ancestor of both "beer" and "nappies" is further from these elements as the tree is traversed than the deepest common ancestor of "beer" and "crisps".

Human users were asked to assess the novelty of the rules that were produced. It was found that the correlation between a human and the computer was similar to the extent to which the human evaluators agreed on the interestingness of rules between themselves.

2.4 E-mail Classification

A chapter of this thesis is concerned with turning an AIS algorithm towards the classification of electronic mail (e-mail). There have been a number of strategies for this task discussed in the literature and the systems proposed broadly fall into two groups: spam filters and e-mail organizers. Both are relevant and so are discussed further in the following subsections.

2.4.1 Spam Filters

Spam (Graham, 2003) is a common term used to describe e-mail that is unsolicited, sent in bulk and usually with a commercial objective. These systems typically classify incoming messages into only two classes, legitimate e-mail and spam e-mail, before these e-mails reach the user client. Two techniques which have been common are collaborative methods, in which many users share their knowledge of junk e-mail to construct a central 'blacklist', and machine-learning based, in which statistical or related machine learning techniques are exposed to spam e-mail and learn the characteristics of this type of e-mail. However, spam e-mail is constantly changing in content and style. For this reason, machine learning techniques are increasingly employed to tackle the problem of spam e-mail. Collaborative methods are widely regarded as too slow to react to changes in spam e-mail content and do not generally use machine learning techniques leaving machine learning as the focus of the remainder of this section.

In (Fawcett, 2003), the author lists four reasons why spam filtering is inherently a hard problem to overcome:

1. Skewed and changing class distributions
2. Unequal and uncertain error costs
3. A disjunctive target concept comprising superimposed phenomena with temporal characteristics
4. Intelligent and adaptive adversaries

Fawcett argues that the way forward in spam filtering is for researchers to begin developing their algorithms on real world data rather than pre-constructed corpuses of spam e-mails, the strategy employed later in this thesis.

Typically all spam filters hide spam messages in some way from the user, but for this to be acceptable safeguards may be usually put in place to ensure false classification of legitimate e-mail (which may be important to the user) is not removed accidentally. (Androutsopoulos et al., 2000a; Androutsopoulos et al., 2000b) is an example in which

this asymmetric loss function (point (2) in the list above) is accounted for. The authors compare a naïve Bayesian approach of spam removal to a memory based approach and assume that discarding one single legitimate e-mail is as bad as classifying 999 spam e-mails as legitimate, and the classifiers used are biased accordingly. (Katirai, 1999) compares a Bayesian approach with a Genetic Programming (GP) method of spam filtering. The GP algorithm performed comparably with the Bayesian method although the resulting GP algorithm was liable to overfitting.

Recently (Cunningham et al., 2003) investigated a case-based approach to spam filtering with the added feature that it may track concept drift. As explained in (Klinkenberg & Renz, 1998), this is a phenomenon where a concept will change or drift over time. In the case of text mining this is especially prevalent in web mining and the removal of spam e-mail. The content of spam e-mail changes over time, for example one month the user may be inundated by software advertisements, whereas the next month this may be replaced by the prevalence of spam about loans. So too may the user's legitimate e-mail interests drift. The user may get a new job for example, where he or she must now deal with software. If the filter does not keep up with this changing concept drift, as spam on the subject of software was prevalent in the past the algorithm may incorrectly begin tagging this now legitimate e-mail as spam. This concept tracking may be done automatically, but is often more effective when the user has at least partial input into the operation (Klinkenberg, 1999).

2.4.2 E-Mail Organisers

E-mail organizers differ from spam filters in that they may often work with more than two classes of e-mail, and the job of this type of classifier tends to be to assign a folder to a message based on the message content. For example, assigning the labels "work" or "friends" to a message and assigning it the appropriate folder. As this type of classifier is not concerned with the specialised attributes of spam filtering, it is a close analogue to that described later in the thesis, so a broad review is attempted below.

Two e-mail organisation systems from the literature are MailCat (Segal & Kephart, 1999), which was integrated into the Lotus Notes client, and ifile (Rennie, 2000), which may integrate into the EXMH mail client. MailCat uses a Term Frequency Inverse Document Frequency (TFIDF) approach to class assignment. By contrast, ifile uses a naïve Bayesian technique to sort messages into folders. Four users tested the ifile system and the results show that users could expect a typical classification accuracy of between 85% and 90%.

This Bayesian classification strategy is found to be common in the literature. For example (Diao et al., 2000) compares a naïve Bayesian system against the C4.5 decision tree algorithm and it was found that, although C4.5 can classify e-mail with greater accuracy, the Bayesian system was more robust overall. Similarly, in (Yang & Park, 2002) Naïve Bayes is compared with a simple TFIDF weighting scheme combined with cosine distance. The paper concludes that a simple Bayesian algorithm is better than a TFIDF/cosine approach in almost all cases, even without feature selection. This TFIDF approach is also investigated in (Brutlag & Meek, 2000), who also compare this to discriminant classifiers and a classifier based on a language model approach. The results showed that none of these three techniques was constantly superior and that the accuracy varies more between mail stores than the tested classifiers.

In comparison to the above systems, (Manaco et al., 2002) describe an e-mail classifier that works primarily on a clustering technique and so marks it as fairly unique in the literature. The advantage of such a technique was that the system would work in an unsupervised way: a distinct advantage over other supervised systems. The recall and precision of some tests were reported but the system was not compared with another, so the effectiveness of this solution remains unproven. One other solution proposed in the literature that moves away from the more common rule based or naïve Bayesian techniques is the system described in (Inoue & Ralescu, 1999). The authors introduce the notion of perceptual text categorisation to the e-mail classification literature, a technique based on fuzzy sets and user perception of the topics. The advantage of this is that a single e-mail may be assigned to a number of different topics at the same time. The results are also somewhat unclear as the class assigned to an item is not a binary choice. Therefore the quality of the final result is reported as four categories: "all correct", "partially correct (>50%)", "partially correct (<50%)" and "all incorrect". Again this system is not compared with another technique, although the method is somewhat unique and so there may be little in the literature to correctly compare against. A review of a number of research based and practical systems for both spam e-mail removal and more general e-mail organization can be found in (Crawford et al., 2001).

A review of the literature regarding the use of artificial immune systems is reserved for the following chapter.

2.5 Interesting Knowledge

Compared with other areas of data mining research such as clustering and classification, there has been little work done on the discovery of interesting patterns in textual data. This is rather surprising as, when performing a traditional rule discovery task, the more interesting the rules discovered, the better they are.

Interestingness (Hilderman & Hamilton, 2001), along with accuracy and comprehensibility, is a measure applied to gauge the quality of a rule which has been generated by a classification system. Interestingness measures can be divided into two groups: objective and subjective. Subjective measures are based on the expectations of the user, whereas objective measures estimate interestingness based on the distribution of the data under scrutiny. Subjective measures have the advantage that they can be tailored to individual users. However they do have the disadvantage that a user is expected to give his or her expectations on a dataset before the measure can be calculated, and are thus somewhat domain dependent. Determining user expectations may become a laborious task depending on the data under scrutiny. Objective measures tend to be less effective estimates of a user's interest in generated rules but are largely domain independent and may be automated.

One particular example of a subjective approach is that of the production of *general impressions* (Liu et al., 1997). In this scheme, the user is asked to specify a number of these general impressions corresponding to high level representations of the rules they expect the system to generate. The hypothesis here is that while the user is not expected to know precise rules, they can give general impressions about what they expect. Discovered rules can then be compared with these general impressions and if the consequent of the generated rule does not match the consequent of the general impression where the antecedents do match (or vice versa) the rule may well be of interest representing novel, unexpected knowledge to the user.

Objective measures use data-driven factors to estimate rule interestingness. In this scenario the system may assume a user will already be aware of certain simple trends present in the dataset. These may include rules with one single antecedent, one consequent and an abnormally high correlation between these in the data. The system can therefore attempt to produce rules a user is not expected to already know by analyzing the class distribution associated with a rule and reporting only those which seem to go against the norm. As an example, if the class distribution suggested that the rule IF(salary='high') THEN(credit='good') was expected by the user because 90% of the data fits this pattern, then the rule IF(salary='high') THEN(credit=bad) may be

considered surprising. (Silberschatz & Tuzhilin, 1996; Liu et al., 1999) are both concerned with measuring interestingness, whilst (Hilderman & Hamilton, 2000) gives five general principles which may be applied to objective interestingness calculations.

2.5.1 Interesting Knowledge in Data Mining

In section 2.1, a number of measures of quality for the output of a data mining system were described. It was stated that a desirable property of discovered knowledge is that this knowledge be interesting. When the interestingness of mined knowledge is mentioned in the literature it is almost exclusively mentioned with regard to rules, be they association rules or classification rules. Rules are by their nature easily interpreted by a human user. This is in contrast to many other data mining paradigms such as neural networks or support vector machines. Both may be used to discover knowledge that is accurate but neither can easily inform the user why they may classify an example as a certain class, and so comprehensiveness suffers. When a user can interrogate the system and find out why an example was given a certain class, that user then has the possibility of discovering a new pattern in the data or one that contradicts the user's expectation. Assuming, however, that the terms by which the system may be interrogated is meaningful to the user. If this is the case, the user has the chance of being surprised, and therefore will find the rule interesting. Recall a rule that may be in the form:

```
IF (antecedent) THEN (consequent)
```

An attribute's value and the effects it may have on the consequent class can be seen immediately.

For the remainder of this subsection, the concept of interestingness is described in terms of rules only. While this is not the only domain interestingness has been researched, it makes a significant proportion of the literature and is sufficient to motivate an explanation of interestingness in terms of text mining in the next section.

To take the three criteria for a good knowledge discovery system once again, it is relatively straightforward to ensure that discovered rules are accurate and by their very nature, discovered rules may be comprehensible, but assessing a rule in terms of interestingness is a lot harder. Take as an illustration the following rule discovered by a medical database found as an example in both (Freitas, 2002a; Carvalho et al., 2003):

```
IF(patient is pregnant) THEN (patient gender = female)
```

While this rule is likely to be 100% accurate and comprehensible, it is in no way interesting as anyone using a medical database would have known this already. In rule discovery systems where interestingness is a factor, a technique for selecting interesting rules from all discovered rules is often applied as part of a post processing stage to complement statistical factors aimed at improving the rule set's accuracy.

With reference to the example shown above, the statement was made that this rule would not be interesting because is would be obvious to the user. There are two ways to detect this situation:

1. Ask the user what they are expecting to find. Anything else may be interesting.
2. Examine the dataset. If the distribution of the data suggests that a certain relationship may be either very simple, incredibly prevalent, or both, then it is reasonable to assume the user will already be aware of this relationship.

These two strategies allow the division of interestingness measures into two groups, namely user-driven and data-driven. A significant amount of the literature refers to these two measures as subjective and objective measures respectively (Carvalho et al., 2003). Both strategies have their pros and cons as summarised below:

User-driven:

Pros:

- Can directly take into account users beliefs and expectations.
- Has access to more direct, relevant, and domain specific information to allow for making better judgements.

Cons:

- User may be reluctant to spend time specifying previous his or her knowledge.
- User may have more knowledge than it is reasonable to specify.
- This approach is domain dependent and user dependent. Within the same application domain one user's knowledge and experience will differ from another's.

Data Driven

Pros:

- More generic than user-driven approach, therefore domain and user independent.
- Relieves user of responsibility of specifying expectations.
- More likely to satisfy a group of potential users, rather than one specific user.

Cons:

- There is an increased risk of discovering rules that are not interesting because the user's knowledge is not taken into account.

The two types of approach are not mutually exclusive and it is not uncommon for both systems to be used in combination. A short review of rule interestingness is contained in (Freitas, 2002a), while (Tan et al., 2002) contains information on 21 different rule interesting measures with regard to association rules. (Tan et al., 2002) describes mainly data-driven approaches and attempts to provide a guide to choosing the right approach in different domains, but also alludes to the advantages of user-driven approaches by commending the selection of measures using rankings made by domain experts. In the next section, the areas of user-driven and data-driven interestingness measures are reviewed separately.

2.5.2 User Driven Approaches

(Silberschatz & Tuzhilin, 1996) discusses the theory behind user-driven measures of interestingness, identifying that a user will find a pattern interesting if it falls into one of the following categories:

1. Unexpected or surprising, as described previously.
2. Actionable. A rule is interesting if the user may do something advantageous with it.

A rule is interesting if it is actionable as in general the mining task is performed in order to discover knowledge which will allow a user to take a specific action. The unexpectedness of a rule is defined in the context of user beliefs. That is, if the rule contradicts a user's belief it is in some way surprising. The authors define two types of belief:

1. Hard beliefs are constraints that cannot be changed with new evidence and usually point to errors in data acquisition when they occur.
2. Soft beliefs are those that the user is willing to change when presented with evidence to the contrary, and are usually the types of belief the algorithm is attempting to contradict when discovering surprising rules.

Silberschatz & Tuzhilin discuss four methods of assigning degrees to soft beliefs and compare the different approaches. It is also stated that the harder a soft belief is held then the more surprised a human may be when that belief is contradicted.

A straightforward method of user-driven interestingness measurement is the use of general impressions, as first described by Liu (Liu & Hsu, 1996; Liu et al., 1997; Liu et al., 1999). In (Liu & Hsu, 1996) a post-processing system for a rule discovery algorithm is described. The inputs to this post-processor to a rule induction algorithm will be a set of fuzzy rules prepared by a domain expert. Fuzzy rules are used as they are regarded simple for a human to construct. For example, a user may be able to describe a range of a numerical attribute values as "medium" rather than having to put specific bounds on what "medium" actually is. This has the effect of summarising the expert's knowledge. The rules generated can then be evaluated against the fuzzy rules specified by the expert. The generated rules regarded as the least similar are expected to be the most interesting.

This work was updated in (Liu et al., 1997), as it was noted that in the previous paper "*too much reliance is being placed on the user's ability to supply the set of fuzzy expectations*". Instead, it argues that in many domains users simply do not know enough about the data under scrutiny to specify fuzzy rules, but they can supply what the authors of the paper term *general impressions*. The general impressions about a domain supplied by a user are constructed using a specification language. A user may specify how he or she may expect a single attribute to affect the predicted class and also how combinations of attributes may affect the predicted class.

(Liu et al., 1999) reviews the idea of general impressions and furthers the work of (Liu et al., 1997) by evaluating the technique both in terms of result and in terms of efficiency. The results showed most rules were unanticipated with respect to the set of general impressions supplied by the user. The authors also note that in general "*the discovered rules serve to inform the user that there are some other attributes there, other than those mentioned in the expected rules, that effect the outcome*".

The work on general impressions was updated once more in (Liu et al., 2000) extending the general impression idea, now called *general patterns*, to include the possibility of specifying special cases. This had the twofold benefit of reducing the amount of output by summarising using general rules, while still identifying the interesting pattern by the use of special cases.

2.5.3 Data Driven Approaches in the Literature

Data driven (or objective) measures of rule surprisingness tend to analyse the structure of a dataset and report findings that are both simple in structure and differ significantly from the norm. For example, there is likely to be a high correlation in a banking database between a salary being high and a loan being granted. This contains only one antecedent and so it is simple to understand. Due to its simplicity, it is expected that a domain expert should know this rule already and so it should be considered uninteresting. However, consider the discovered rule:

```
IF (salary = high) AND (bank account = no) THEN (loan = no)
```

This is more interesting to the expert as a high salary now no longer automatically leads to a loan being accepted. A data-driven measure therefore must look at all antecedents and discover which antecedents, whether interacting with others or not, tend to produce a given consequent. It may then look for rules where that antecedent is encountered but a different consequent is predicted and assume this will be interesting.

Two papers that give good overviews of the area of data-driven measures of interestingness are (Tan et al., 2002), which is rather technical, while (Carvalho et al., 2003) provides more of a conceptual discussion. In the latter, the authors set out to review four objective rule surprisingness measures and make an enlightening conclusion that "*finding the best rule surprisingness measure for an application is harder than finding the best algorithm (in terms of classification accuracy) for a given dataset*".

Exception rules, one of the four measures investigated in the above paper, are described in (Suzuki & Zytkow, 2000). The algorithm described will find every occurrence of an exception rule, that is, a rule that provides a significant deviation from rules considered obvious. It is pointed out that these rules are often pruned away as they are considered to be noise, but if the noise in the data set is low, they do represent an actual relationship, and so can often be the most interesting rules discovered.

(Freitas, 1998) notes how rule surprisingness measures may be based on a) small disjuncts (rules) and b) the level of surprisingness associated with individual attributes using information theory. With regard to small disjuncts, which are rules covering few examples; such disjuncts are more prone to be noise in the data set rather than a truly interesting rule. However, they have the potential to be more surprising than large disjuncts, i.e. rules with a large number of examples. The second measure uses information gain of attributes to compute rule surprisingness. The more relevant an attribute is to a class, the higher its information gain. The premise here is that the user is likely to know these relationships already, thus the user is more likely to be surprised if he or she is shown short rules with attributes having individually low information gain although the combination of all attributes in the rule has a good predictive power due to attribute interaction.

(Hilderman & Hamilton, 2000) makes a significant contribution in the area by developing five principles that any data-driven measure of interestingness should satisfy. 15 different diversity measures are evaluated against these five criteria. Mathematical proof is used to identify the measures that fulfil the principles. Thus allowing an informed choice to me made when selecting a measure.

2.6 Web Mining

Since its beginnings as a small number of interconnected computers used for academic data exchange the World Wide Web has become mankind's largest repository of information. The term hypertext was reported as far back as 1965, but the World Wide Web as it is known today can be traced back to the early 1990s when a worker for CERN, Tim Berners-Lee, wrote the first program with a Graphical User Interface (GUI) to navigate between bi-directional links in a collection of documents. This software was named "World Wide Web". Although the act of navigating between documents in this way was nothing new, the addition of a graphical user interface was. This new user interface was combined with another innovation to be devised at CERN, that of HTTP: HyperText Transfer Protocol. These two innovations allowed easier transfer of hypertext documents and combined that with a more user-friendly way of displaying and navigating among these. During 1993 the amount of HTTP traffic grew ten-fold (Chakrabarti, 2003) and over the coming years the meaning of "World Wide Web" changed from describing a piece of software to describing the actual "web" made by these interlinked hypertext documents.

Before continuing it is worth defining some terminology. The web, short for "world wide web", is the collection of interlinked pages written in the HTML language and accessed using the internet. As well as actual content, HTML allows the insertion of hyperlinks. For the purposes of this thesis, these are pointers from one HTML document to another document also found on the web. It is estimated that 75% of the internet is comprised of such pages rather than other media.

The task of information retrieval using the web is challenging for a number of reasons. Not least of which is the size of the search domain, for example at the time of writing, Google (Google, 2006b), regarded as the web's most popular search engine, indexes over 8,000,000,000 individual web pages. Web mining is a specialised form of data mining where the data comes directly from the Web, or user interaction with it. Web Mining may be defined as (Kosala & Blockeel, 2000):

"The use of Data Mining techniques to automatically discover and extract information from web documents and services."

After a study of the literature (at the time of the article's writing), Kosala and Blockeel further decompose web mining into four areas:

1. Resource finding (retrieving intended web documents)
2. Information selection and pre-processing (automatically selecting and pre-processing specific information from retrieved web pages)
3. Generalisation (Automatic discovery of general patterns at individual websites and across multiple sites)
4. Analysis (validation and interpretation of the minded patterns)

As the web contains an enormous amount of diverse information there are a number of tasks a user may perform to mine data, retrieved both from the web and caused by user interaction with it. The concern of this thesis is mainly with the mining of content, although concepts gleaned from the mining of structure are also used, and as such are detailed below.

2.6.1 Mining Web Usage

Each individual web server typically records each individual data item requested, whether it is a webpage, graphic or any other piece of data. This unordered data is called a clickstream and is where usage mining generally begins. It is from this raw clickstream that a usage mining system must try and identify individual users' browsing

habits and therefore mining usage is as much about the pre-processing of this raw data as the techniques used to mine it. Mining usage consists of a number of tasks to pre-process this data and discover knowledge from it. The pre-processing steps are filtering, de-spidering and user identification with the goals being sessionisation and path completion. For further information on this topic the reader is directed to (Srivastava et al., 2000; Linoff & Berry, 2001; Chakrabarti, 2003), while (Yang et al., 2001) describes how such a technique may be used to improve internet pre-fetching and caching.

2.6.2 Mining Web Structure

The interconnected structure of the internet can be regarded as a directed graph. Each page is a node and each hyperlink an edge. The structure of the graph created may lead to insights about the area of the internet described by that graph, some of the most striking examples of which are currently produced by the OPTE project (OTPE, 2005). One example of the use of structure analysis is the generation of a measure of importance for a particular page or site. By linking from one page to another, a human is making the judgement that the target page is worth linking to. Therefore each hyperlink to that site is effectively a recommendation of that site. "*a site that links to many authorities is a hub; a site that is linked to by many hubs is an authority*" (Linoff & Berry, 2001). Using this strategy allows search engines to rank pages in order of estimated importance when returning results of a keyword search.

One other use of structure mining is that of understanding local internet structure and tailoring it to influence user behaviour. This is a completely different use to that above and is often practiced by commercial websites to increase sales. Destination pages are the pages which tend to be at the end of a number of navigation pages (those that give hyperlinks to others with little content), and are where the visitor will spend the majority of the time. These destination pages offer the website's content and as such the websites structure may be mined to determine if users may easily navigate to these. Structure mining in this way is not very informative and usage data is commonly mined and combined with structural data for a more useful overview of the site.

2.6.3 Mining Web Content

The main focus of this research is in the mining of content from the web. Most would associate this action with searching the web, and to some extent this is true. Web search techniques can indeed be classed as web content mining, but the area is much more diverse than simple web searching. Content mining aims to extract/mine useful

knowledge from web page contents. In (Liu & Chang, 2005) the author gives 12 separate reasons why web content mining is a challenge. These points can be reduced as follows:

1. The data on the web is diverse and growing rapidly with pages being added, removed and updated constantly.
2. Data of different types can co-exist such as text, pictures, sounds, etc. and web pages may present the same information in different formats, using different styles. For this reason, much of the information on the web is redundant.
3. Web information is semi-structured as it not only includes information but also layout information and hyperlinks to other pages.
4. The web contains a lot of noisy data. One page may contain a lot of information that is not on the topic of that page, such as navigation panels and advertisements.
5. The deep web (pages behind passwords and suchlike) tends not to be accessible to automated search tools.

It is worth making the distinction between the normal web (the shallow web) and what has been termed the deep web. The deep web is the portion of web pages that are for some reason inaccessible to normal search engines. A straightforward example would be a portion of a website that requires a password to log in. Similar to this is the situation in which filling out some details on a web form may yield dynamically generated results. The page is constructed dynamically from information retrieved from a database. Normal automated search tools cannot fill out that web form. This will leave a portion of the site undiscovered. The shallow web is the portion of the web where no such hurdle to access exists.

2.6.4 Document and Webpage Classification

A typical task in the web mining literature is that of document or web page classification. That is determining the class of a web page drawn from a finite set of classes based on the text the document contains. From the literature, two motivations can be identified:

1. Producing hierarchical topic directories.
2. Finding pages a user will be interested in or find relevant.

The result of situation (1) may have many goal classes, and is made more complex by the fact that these topic directories, such as Yahoo! (Yahoo! 2005) or Google Directory (Google, 2005a), tend to be arranged in a hierarchy. Given a tree of topics where each node is a topic and the documents become more specialised as the tree is traversed from the root, a document may be best suited to a general node, or it may be more suited to one of that node's children or grandchildren. The classification of items in hierarchies is an active area of research, but not the concern of this thesis.

At its core, situation (2) can be achieved with a keyword search over a set of boolean vectors. That is, there exists the situation where only two classes are present – documents of interest and those not of interest. A search engine will make this classification each time a query is submitted. This thesis, however, is concerned with classification based on a much more complex set of user preferences, interests or opinions, etc., and these cannot be specified by a small number of keywords, rather they must be learned.

The following concentrates on web page classification, that is classification as examined previously, but using documents retrieved from the web rather than data drawn from databases or plain text documents. The majority of approaches use models such as vector space to represent web pages and then use standard distance metrics (section 2.2.2) combined with classic classification algorithms (section 2.1) to perform the classification. Techniques other than those detailed above include the use of support vector machines (Sun et al., 2002) and ant colony algorithms (Holden & Freitas, 2004).

2.6.5 Use of Metadata

One of the primary differences between normal text and documents found on the web is that web documents tend to come with attached metadata. For example, elements in the HTML of web pages such as hyperlinks or headings can give extra clues about the relevance of certain content and as such be used to enhance classification. Hypertext is a form of semi-structured document and therefore, ideally, an algorithm would exploit this structure to increase classification accuracy as investigated in (Yi & Sundaresan, 2000).

(Chakrabarti, 2000) provides an overview of how hypertext can be used for data mining. Uses identified not only include information retrieval but also social network analysis (Kumar et al., 2002). (Ghani et al., 2001) is a typical example of hypertext exploitation in which the extra information typically available in the hypertext tags is used to increase classifier performance. The author states that using text only can yield

sub-optimal performance but standard text classification algorithms can successfully be used to classify hypertext using metadata. It is noted that the use of hyperlinks alone, one form of hypertext metadata, can harm accuracy as they can be noisy. Indeed, in (Laware, 2005) it is stated that metadata is currently "*non-existent, ill defined or erroneously labelled*". Thus for continued accurate web search more well defined metadata is required.

One exploitation of the structure of web documents is the discovery of blocks on the page that hold the content, rather than noise. (Lin & Ho, 2002) and (Song et al., 2004) are examples of this. (Lin & Ho, 2002) assume content is contained within specific cells of a table, while (Rowe, 2005) attempts to identify captions associated with multimedia to enhance retrieval of multimedia objects.

2.6.6 Spidering

Spidering (also known as crawling) is the action of retrieving and locally storing web pages ready for further processing. For this reason, spidering is an integral part of content mining and structure mining. The web, being fundamentally decentralised, has no index, there is no catalogue of all documents contained on the web. Therefore an algorithm that will construct this index is required, this algorithm is known as a spider (also called a crawler or web-bot, bot, or robot).

A spider relies on the hyperlinks that authors have inserted into web pages to make its way from page to page. In its simplest incarnation, the spider may begin with one single page in memory. This page is assumed to contain a number of outgoing hyperlinks to other pages. Each outgoing hyperlink will be added to the tail of a queue while the hyperlink at the head of the queue is then removed and accessed. This new page may be stored or processed and will in all likelihood also contain a number of hyperlinks. These new hyperlinks are checked, to determine if the spider has accessed them before, and therefore need not follow them again. All un-followed hyperlinks are again added to the queue. This process then repeats until the queue of unvisited hyperlinks is empty, although in practice this may never happen (Figure 2.5). The book (Chakrabarti, 2003) contains further information about crawler implementation.

This basic crawling algorithm is, in theory, all that is needed to retrieve just about every page in the web, but writing a large scale system able to retrieve and process/store a significant percentage of the web is a challenging task both in terms of hardware and algorithm design. Firstly, the queue of unseen links will grow very quickly and can reach an enormous size. Elements of the queue must be able to be checked very fast so

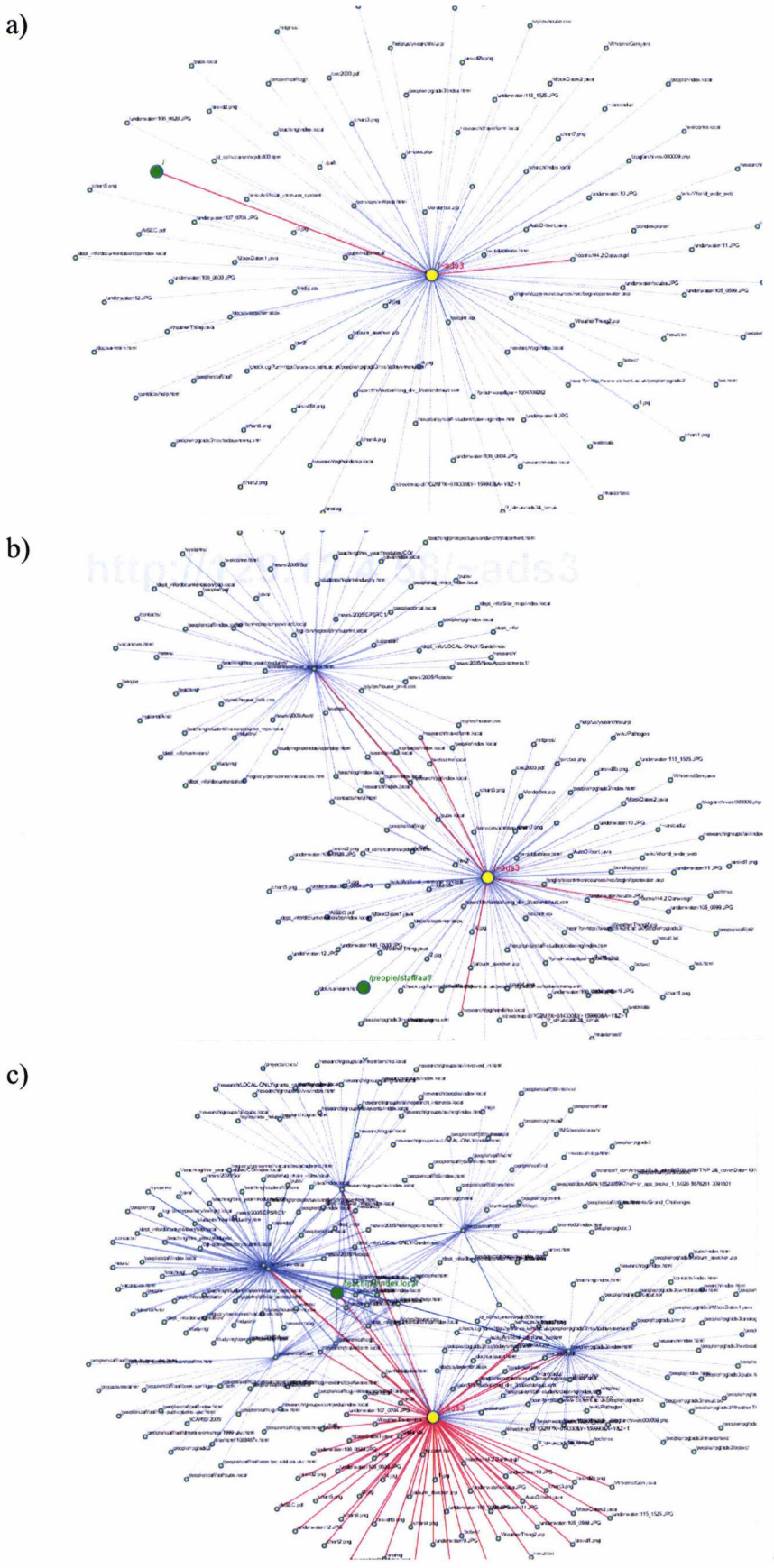

Figure 2.5. A graphical representation of the spidering process. a) Spider begins by determining all outgoing hyperlinks from a starting page. All outgoing hyperlinks are added to a queue. b) The head of the queue is removed and the target of the hyperlink is parsed. All outgoing hyperlinks from this page are added to the queue. c) The pages referred to by all outgoing hyperlinks from the start page have now been retrieved and all outgoing hyperlinks from these have been added to the queue. Already the queue now contains well over 100 hyperlinks yet to be followed.

duplicate entries do not occur and the head must be quickly accessible. Secondly an industrial sized spider must consider the latency of the network impacting on performance and therefore the submission of parallel requests is normal. However, mechanisms must be in place such that a single target machine is not swamped with requests. For reasons of efficiency when adding pages to the unseen page queue and possibly for the accuracy of the derived information, a spider should have some mechanism for identifying duplicated pages. There are some techniques to account for this but these are beyond the scope of this document. The interested reader is directed towards (Chakrabarti, 2003).

2.6.7 Topic-Specific Spidering

When a user is attempting to build a more specialised data repository, it may not be feasible to use a general crawler as described above. A general crawler will retrieve vast amounts of data, almost all of which will not be on a specific topic no matter what the starting page. The most efficient way to gather data only on a specific topic is by *focused crawling* or *intelligent crawling*, that is only retrieving pages on a particular topic while trying not to traverse off-topic areas and is performed using a *topic-specific crawler*. In practical terms most systems described in the literature use focused crawling to rank hyperlinks in the order in that they are to be retrieved, thus improving the quality of the crawl by retrieving the most relevant pages quickly. Many of these crawlers rely on Chakrabarti's "radius-1 hypothesis" *"If page u is positive and u links to v, then the probability that v is positive is higher than a randomly chosen web page is positive"* (Chakrabarti, 2003).

Chakrabarti has been active in the field, coining the term focused crawling in (Chakrabarti et al., 1999). Chakrabarti created a system that is first trained in a supervised manner to recognise pages on a particular topic. When a webpage is retrieved, if the classifier regards that page as on topic all its outgoing hyperlinks are added to the work queue, otherwise it is discarded. Further to this, a Bayesian classifier may be trained to make judgements on relevance about a page. Using this, all outgoing hyperlinks could be assigned a measure of relevance the same as the page, and these are added to a priority queue rather than a first-in first-out (FIFO) queue as before. Thus the most relevant hyperlinks are analysed first. Chakrabarti goes on to produce results over a number of experiments using this crawler, noting that a random crawl (adding random hyperlinks from a page to the work queue) results in a topic being lost within 100 page

fetches whereas a focused crawl as described above acquires relevant pages even after thousands of page fetches.

Two recent topic-specific crawlers no longer rely just on this simplistic view. For example, (Wu & Hsu, 2005) investigated a topic specific crawler that measures graph distance between a given page and the given topic to determine the order in which pages should be visited.

Both the above systems still rely on hyperlinks for navigation, but as noted in (Aggarwal, 2004) "*Recent studies have shown the topical correlations in hyperlinks are quite noisy and may not always show the constancy necessary for a reliable discovery process*". Hence the authors combine content mining techniques with the analysis of user behaviours drawn from web logs, usage mining. The resultant paradigm they called collaborative crawling. Tests were performed using a crawler based on content mining only, usage mining only and combined content and usage. The combined crawler outperformed the others in every test performed.

(Wu & Hsu, 2005) state that "*some sets of off-topic documents often lead to highly relevant documents*" thus "*an optimally focused crawler should sacrifice visiting several off topic pages in order to reach the highly relevant pages among the hyperlinks*".

2.6.8 Web Personalisation and Filtering

Web personalisation is the process of changing or filtering the contents of web pages to suit an individual. The concept of personalisation is examined in (Wright, 2002), who conjectures that there are three ways to create personalising filters: content-based approaches (e.g. web content mining), collaborative approaches and hybrids of these two. In this section, as with the rest of the thesis, only the content based approach is of interest. In (Sheth, 1994) a literature review discovers just three papers at that time had been published dealing with learning agents for information filtering.

The literature tends to agree that, as a general rule, relevance feedback (repeatedly asking users to rank pages and refining the search based on this feedback) should be avoided (Chan, 1999; Chen et al., 2001). This strategy may work for research systems to allow authors to test the accuracy of algorithms but no system where users must rank webpages has been found to have progressed much past the research stage, the automated inference of this data is therefore imperative. This situation is explicitly investigated in (Chen et al., 2001). It involves the notion of non-invasive learning using a proxy to detect surfing habits thus allowing tracking of concept drift.

The parallels between Usenet news filtering and web personalisation are discussed in (Pretschner & Gauch, 1999). This report describes a number of Usenet news filters and then reviews a number of web personalisation services and web search assistants. 50 separate systems are summarised in tabular form and as such represents a good summary of the sort of work happening in this subject area at that time. Not many of those detailed are based on learning algorithms however, while it is also stated that "*due to a lack of data, a comparison of the systems with respect to performance is currently impossible*".

(Aggarwal & Yu, 2002) describes a system to produce a personalised summary of news with a user being shown news articles that may be of interest but also incorporating a method of detecting sudden variations in news patterns and alerting the user. Although (Aggarwal & Yu, 2002) refers to "interest" in terms of relevance rather than in terms of unexpectedness or novelty, etc., the sudden variations of topic could well be surprising to a user. Therefore, it is believed that the paper implicitly proposes the use of interestingness albeit not describing it as such. (Widyantoro et al., 1999) also attempts a solution to the problem of personalising news.

(Shahabi & Chen, 2003) is a position paper that argues, in the context of search personalisation, user preference may be used to enhance search in the following ways: personalised page importance, query refinement and personalised metasearch. The investigations later in this thesis both personalise page importance and allow a user to refine a query.

One strand of webpage classification that has been researched steadily has been that of website filtering or web personalisation. To this end there are some parallels with e-mail filtering (as discussed in section 2.4) in that the system is attempting to learn user preference, then either filter out everything else, in the case of e-mail, or actively seek to find new documents and make recommendations. The most significant are reviewed in the following section in rough chronological order as to show progression. It should be noted here that automated personalisation is not the subject of this thesis. Some literature deals with users personalising their own web portals – front doors to web sites. Using "check box personalisation" a user can add or remove certain content from that portal. One learning system for such portals is described in (Aggarwal & Yu, 2002) however, check box personalisation is almost exclusively done by the user without computational intelligence and so will not be examined further.

WebWatcher (Armstrong et al., 1995; Joachims et al., 1996) could be considered one of the first personalisation systems. Although it had somewhat limited scope it set a

certain precedent for future work. WebWatcher was specialised for one website only: the school of computer science at Carnegie Mellon University, but learned from multiple users. The system ran, server-side, as an interactive guide for the site. First assessing the user's information need then highlighting appropriate hyperlinks as the user browsed, thus guiding the user to goal. This has the added advantage of allowing the user to absorb information on the way to that goal, unlike traditional web search. However the problem here would be that this may not scale well. Such a "tour guide" through a relatively small intranet site is feasible but the internet has vastly more paths from one point to another.

WebMate (Chen & Sycara, 1998) is another personalisation system helping with both browsing and searching. It learns user interests and continuously updates these. It automatically provides documents matching those interests and helps the user refine searches to increase the relevant documents returned when searching. The method is straightforward in that it uses a number of TFIDF vectors (recall section 2.2.2) to keep track of the users preferred features in the user's interest domains. WebMate was experimentally evaluated but the results showed the average classification accuracy was just 30%. This was attributed mainly to noise in the webpages (advertisements etc.). But it was clear more sophisticated efforts at determining blocks of content in web pages are needed.

(Chan, 1999) makes the statement again that "*web personalisation products must be non-invasive for the user*". Although it may be the first paper to mention interestingness in the context of webpage filtering, it fails to use the term interestingness in the data mining sense (of novelty, surprisingness or unexpectedness, as defined in the following section). Rather interestingness is used to denote relevance.

The aforementioned papers make use of techniques such a statistical or rule induction approach to learning, but a variety of techniques have been used. One is used in (Tanudjaja & Mui, 2002), called Persona. Focused more on a personalised web search, Persona is a graph based search technique which, importantly, recognises the significance of synonyms and hypernyms in web search and mentions WordNet in relation to tackling this problem. The authors concluded Persona produced "*reasonable results*" although no figures were given and any use of WordNet seems to have been ignored in the analysis.

Amalthaea (Moukas, 1996; Moukas, 1997; Moukas & Maes, 1998) learns user interests from the user's browsing history. It then creates a closed eco system populated with agents where each agent must compete for resources in that environment. Two

types of agent are used, those for information filtering and those for information discovery. The fittest agents reproduce using crossover and mutation, while the least fit are removed from the population. Resources are made available based on the overall fitness of the population. Although the usual vector space model is used with features weighted using a variation of TFIDF, one of the most novel aspects of this system is the manner in which a user gives feedback to the system. Credit is indirectly assigned (or removed) by a user based on that user's action when acting on the results. This, in turn, directly influences the fitness of the agents originally retrieving that information. Although not mentioned explicitly in the text, it can be inferred that the system is able to track concept drift (Cunningham et al., 2003). Tests were performed and the system evaluated, although again not in a way that may be directly compared to any other. The system was tested in a somewhat artificial manner with a simulated user whose interest changes over time. As the first biologically inspired web filtering system it is of particular interest, but is not directly linked to the subject of this thesis.

The research in personalisation is, of course, still continuing. Recently, Markellou published a book chapter reviewing the past year's research in the area and analysis of current trends to make some predictions at future directions (Markellou et al., 2005).

2.6.9 Interestingness and Web Pages

(Liu et al., 2001) is directly engaged with the task of evaluating the interestingness of web pages and can be seen as one of the main motivations for the work in this thesis. The paper makes a number of statements regarding the inadequacy of existing search techniques to retrieve interesting information as opposed to information regarded only as relevant. The paper is set in the context of a business where a user may want to discover unanticipated information of a competitor's website. It argues that unexpected information is often of great interest to a user and existing web extraction techniques are unsuitable for this type of information extraction. Unexpectedness is defined as:

"*A piece of information is unexpected if it is relevant but unknown to the user or it contradicts the user's existing beliefs or expectations.*"

This can be seen to be equivalent to the characterisations of interestingness in section 2.5. The use of the term "*relevant*" in the above definition is important as not every piece of unknown information is interesting.

The authors of (Liu et al., 2001) implemented a number of metrics to assess the interestingness of a web page, the combination of which was called WebCompare. After

pre-processing, documents are represented as points in vector-space using TFIDF to weight features, or as sets of "concepts" where a concept is a set of keywords that occur together in a page above a certain user-specified minimum support threshold. In the following explanations, a known web site is referred to as U (the User site), whereas that site which is to be mined is referred to as C (the Competitor site). Given the sets C and U, u_j is a page drawn from site U and likewise c_i is a page drawn from the competitor site, C. All calculations involving term frequency use a normalised term frequency computed as shown in Equation 2.14.

$$tf_{i,j} = \frac{f_{i,j}}{\max f_{i,j}} \qquad \textbf{2.14}$$

1. **Finding a corresponding C page of a U page**. The user wishes to find the equivalent of a U page on the competitor's site. For this the cosine distance between all pages in C and the user page is computed and the closest one in vector space chosen. This is performed using the method in section 2.2.2 and will not be repeated here.
2. **Finding unexpected terms in a *C* page with respect to a *U* page**. This will allow a user to discover features on the competitor's page that may be related to those on the user page but not mentioned specifically on the latter page. As such, it is thought that these terms will be surprising to the user. The unexpectedness of each term is ranked using a score computed by the formula below, where $tf_{r,i}$ is the term frequency of the rth term in document c_i, normalised by the maximum term frequency in document i. $tf_{r,j}$ is the normalised term frequency of the corresponding term in u_j.

$$\text{unexpT}_{r,i,t} = \begin{cases} 1 - \dfrac{tf_{r,j}}{tf_{r,i}} & \text{if } \dfrac{tf_{r,j}}{tf_{r,i}} \le 1 \\ 0 \text{ Otherwise} \end{cases} \qquad \textbf{2.15}$$

3. **Finding unexpected pages in C with respect to U.** Finding pages in C that are most unexpected compared with U can be useful, as it can tell the user that the competitor site contains pages not found in their own site. All pages in C are combined into to a single document, so too are all pages in U. The term unexpectedness of every term in C is computed with respect to U, then

for each page, *P*, the sum of term unexpectedness on that page is computed and normalised by the page length, as follows:

$$\mathrm{unexpP}_i = \frac{\sum_{r=1}^{m} \mathrm{unexpT}_{r,c,u}}{m} \qquad 2.16$$

Where $unexpP_i$ is calculated as in Equation 2.15, *r* refers to the specific term in sets *C* and *U*. *m* is a normalising factor and is the total number of terms shared between the elements of sets *C* and *U*. Therefore it is also the maximum value for *r*.

4. **Finding unexpected concepts in a C page with respect to a U page**. Concepts (often called co-locations) can often be more revealing than sets of keywords alone. The co-location of words such as "data mining" rather than the words "data" and "mining" reoccurring apart on a page may confirm that page is on a particular subject. Association rule mining using the Apriori algorithm is used to discover these concepts.
5. **Finding unexpected outgoing links**. These hyperlinks may indicate a useful resource as yet unknown to the user. The unexpected hyperlinks are simply those found in *C* but not in *U*.

Although the measures used to compute the above aspects of unexpectedness are objective measures, the opinion of a user is always required to validate the assumptions made in constructing them. Therefore, the evaluation of any system such as this will always be made in a subjective way by a user. Coupled with the fact that no comparable system could be found, evaluation was difficult. WebCompare was received positively by three users asked for their opinion. The reported comments were positive towards the WebCompare system, with some useful observations being reported. These included opinions such as the system allowed a user to browse more deeply into the site rather than becoming impatient and stopping browsing on high-level pages, or similarly allowing the summarization of long pages with keywords. Thus, users who would otherwise grow impatient with a long page were prompted to read it in detail as they then had the motivation to spend time. Some advantages of this tool were cited as the summarising capability allowing a user to focus on the relevant aspects of a competitor site and the way in which such an automated system is less likely to miss important

concepts compared to manual browsing. This system is revisited in some detail in Chapter 6.

2.7 Summary

This chapter has described in some detail the areas of data mining, text mining, discovery of interesting information and web document mining. As examples of application areas e-mail classification and web personalisation have been described and the literature reviewed in some detail. Inspired by WebCompare, as reviewed above, it is one of the tasks of this thesis to create a web mining system that will return interesting (relevant and novel, surprising or unexpected) web pages to a user. Using this chapter as inspiration, the following motivational statement can be made:

Current web search techniques are geared towards finding relevant pages on the web. In some situations pages that are highly relevant may not be interesting for a user. For a page to be interesting the information it contains must be novel, unexpected or contrary to a user's previously held beliefs. Current search techniques do not cater for this scenario, as the very nature of the method by which users currently specify search criteria is not able to cater for this situation. A user will find it hard to find unexpected pages as a keyword search. After all, how can a user discover something unexpected when by its nature a user cannot specify search criteria (i.e. keywords) for knowledge the user does not yet possess? It is also virtually impossible for a keyword-based system to make accurate assumptions about what a user knows already and what he or she does not know, as this information cannot be summarised by a small number of keywords.

To summarise further, this thesis takes the view of (Liu et al., 2001) and hypothesise that a user will find a web page interesting if that page is relevant to the user's search and also fulfils al least one of the following criteria. The content of the page is:

1. Novel
2. Surprising
3. Unexpected

Given a user performing a search over a number of web pages and having certain prior knowledge of the search domain:

1. A novel page is relevant to the search and provides the user with information he or she did not know already.
2. A surprising page contains information which is relevant to the search again but some way contradictory to the user's current beliefs.

3. An unexpected page is related to the search domain providing the user with otherwise unknown information relevant to the search but is outside the core search domain. While the subject would therefore be only related to the search, it would contain very little of the user's prior knowledge. An example here would be the user performing a keyword search over a set of documents and having results returned that do not contain any of the original search terms.

One of the goals of this thesis is to construct a system to mine from the web pages that the user will find interesting. That is, the user may consider them novel, surprising or unexpected. Rather than relying on the predefined objective measures of interestingness as used by WebCompare, it is believed that the use of adaptive machine learning, as introduced in the following chapter, may be advantageous. The specification of prior knowledge can be done by giving the search algorithm a small set of web pages that are assumed typical of this user's prior knowledge. It is then the job of the algorithm to try and learn what the user may find interesting, search for it, and refine its hypotheses if necessary.

Chapter 3 Immune Systems

In this chapter, natural and Artificial Immune Systems (AIS) are described in some detail and continue the foundation on which to base the remainder of the thesis. This chapter begins with a motivational description of the natural immune system. The attention of the chapter then shifts to describing algorithms rooted in the immunological world. This information is necessary to allow further discussion of AIS. When describing the artificial systems, the engineering view is emphasised. With this knowledge, the chapter is concluded with a review of data mining found in current and previous AIS research.

3.1 Artificial Immune Systems in Context: Biologically Inspired Paradigms.

In general, artificial immune systems (AIS) are one of many biologically inspired algorithms. As their name suggests, biologically inspired algorithms are computer algorithms where the motivation for their principles and attributes comes from the quest for adaptability and robustness of biological paradigms. Examples of these biologically inspired algorithms include the well known discipline of neural networks (Zeidenberg, 1990), the inspiration of which is the natural brain; ant colony optimisation (Dorigo & Stützle, 2004), which uses the principles found in the behaviour of ants to solve problems; evolutionary algorithms (Holland, 1992; Mitchell, 1996), which use the principles found in the Darwinian theory of evolution to evolve solutions to problems; and artificial immune systems, which use the principles and processes of the natural immune system to solve problems. For an introduction to biological and naturally inspired algorithms the reader is directed to (Bentley, 2001).

The theory of evolution underpins much of the material in this chapter, and indeed throughout this thesis. Charles Darwin proposed a number of hypotheses concerning the development of species within their environment based on the following observations of the natural world:

1. The number of offspring tend to be larger than the number of parents.
2. The number of individuals in the species tends to be approximately constant.
3. From (1) and (2) there will be competition to survive.
4. There are genetic variations within the same species of individual.

Darwin hypothesised that this genetic variation combined with the competition to survive leads to *natural selection*. That is, those individuals better suited to survival in their environment will have an increased chance of surviving and passing on their genetic material. Later three additional hypotheses were added, leading to neo-Darwinism:

5. Any sort of continual variation must be responsible for the introduction of novel information in the genetic material of the organism.
6. There is no limit to the variation that can occur.
7. Natural selection allows the preservation of new information corresponding to better adaptation.

This notion of neo-Darwinism is regarded as a more suitable explanation for the observed natural selection phenomenon, and so will be used as a synonym for evolution and natural selection from here on.

Of the biologically inspired algorithms mentioned previously, AIS in the form of clonal selection is often compared with genetic algorithms (de Jong, 1999). Both clonal selection, a type of AIS algorithm, and genetic algorithms can be considered under the general topic of evolutionary algorithms. Evolutionary algorithms (EAs) have at their basis the notion of evolution as described by Darwin and use the processes described in the evolutionary theory as metaphors for algorithms. They encode potential solutions in the form of individuals, a population of which evolve in a closed environment towards a solution. There are many types of EA, but all have some elements in common. In (Freitas, 2002a) these are stated as:

- A population of individuals, each element of which is a candidate solution.
- A selection method based on a measure of the quality of the candidate solution represented by that individual.
- Generation of new individuals by a method of inheritance from existing individuals. These new individuals are generated by applying probabilistic operators to individuals of the current generation.

These three points are described similarly in (de Castro & Timmis, 2002a) as reproduction with inheritance, genetic variation and natural selection.

To take a Genetic Algorithm (GA) as a specific example, each individual will encode a potential solution or, more generally, a position in the problem space in that individual's genetic make up. Two parent individuals are then selected probabilistically, with the chance of choosing each based on their fitness. The notion of fitness is defined by use of a fitness function, the purpose of which is simply to evaluate an individual's suitability to its environment. Thus, if the better individuals tend to be selected more often to reproduce, then there is a tendency to increase the average fitness of the population over time. There are a number of ways to perform this probabilistic selection, including tournament selection and roulette wheel selection, the latter detailed in Appendix B. Genetic variation is introduced in a GA, mainly by the crossover operation which consists of crossing the genetic material in one parent individual with the other, thus producing two offspring. There are a number of ways of achieving this switch of genetic material such as single point and multi-point crossover and uniform crossover. It is not of concern here to go into the details of crossover mechanisms. One other form of genetic variation is that of mutation, which is simply the choosing of one or more elements in the individual's genetic make up and swapping it for another legal value. This has the effect of introducing new genetic sequences which in turn may prevent a population converging at a local optimum when a global optimum is the search goal. The clonal selection algorithm can also be considered one example of an EA and the commonalities will be made explicit later in this chapter.

Now AIS have been placed in some context with other biologically inspired paradigms, the area of AIS can be explained in more detail. This begins with a review of the natural immune system, on which AIS are based.

3.2 Immunity

During a review of the literature it has been noted that from the beginning of AIS as a subject area authors of publications feel compelled to describe the natural immune system. Much of the immunology found in these papers tends to be more complex than necessary. In this subsection, basic immunology is reviewed but only at a level to aid understanding of the remainder of the thesis. For a more detailed explanation the reader is directed towards the literature such as (Nossal, 1994; Sompayrac, 1999) from which, in combination with (de Castro & Timmis, 2002a), much of this section is derived. The aim is to cover the immunology to such an extent that a reader may appreciate the

individual examples of AIS as reviewed near the end of this chapter and is given enough explanation to allow an understanding of the two AIS algorithms presented in later chapters.

This section begins with a short summary of the innate immune system. This is not commonly used in the artificial domain and as such will be glossed over. The innate immune system, however, is thought to influence the function of the adaptive immune system and as such is worthy of a few paragraphs in review.

3.2.1 Innate Immunity

The innate immune system (Germain, 2004), as its name suggests, does not change over time and as such has not attracted much attention in the AIS community. However, this is beginning to change, especially with the current interest in Danger Theory (Matzinger, 2002a) as can be seen by the current plethora of papers such as (Aickelin & Cayzer, 2002; Aickelin et al., 2003; Secker et al., 2003b; Greensmith et al., 2005; Twycross & Aickelin, 2005).

The innate immune system is set to recognise and attack a small number of common invaders. The innate immune system will destroy most *pathogens* (potentially harmful invaders) on first encounter and due to this efficiency the adaptive immune system is only required a tiny fraction of the time. The adaptive immune system can take some time to begin an effective reaction so it is the innate immune system's task to react quickly and to keep an attack under control until the adaptive system can react efficiently. Cells of the innate system can often act as antigen presenting cells (APCs). That is, they recognise entities that may be pathogenic and present these in the proper way to cells of the adaptive system. As such, they play a vital role in triggering an adaptive response (Janeway, 1994) and the role of innate immune components as APCs is referred to a number of times in the following sections.

3.2.2 Adaptive immunity

Once activated, the innate immune system typically stays activated for just a few days. In contrast the adaptive system can stay activated for weeks. It is the adaptive immune system's task to remove a pathogen when the innate immune system has been overwhelmed or is otherwise ineffective. This may be because the innate immune system cannot generate an attack specific enough to an invading pathogen. In this case the adaptive immune system is activated because it may adapt to protect against pathogens. Unlike the innate system this adaptive response is specific and also, unlike

the innate system, the adaptive system retains memory of the event and so may recognise the same pathogen when it is presented again, this time mounting a much quicker response (secondary response) as shown in Figure 3.1.

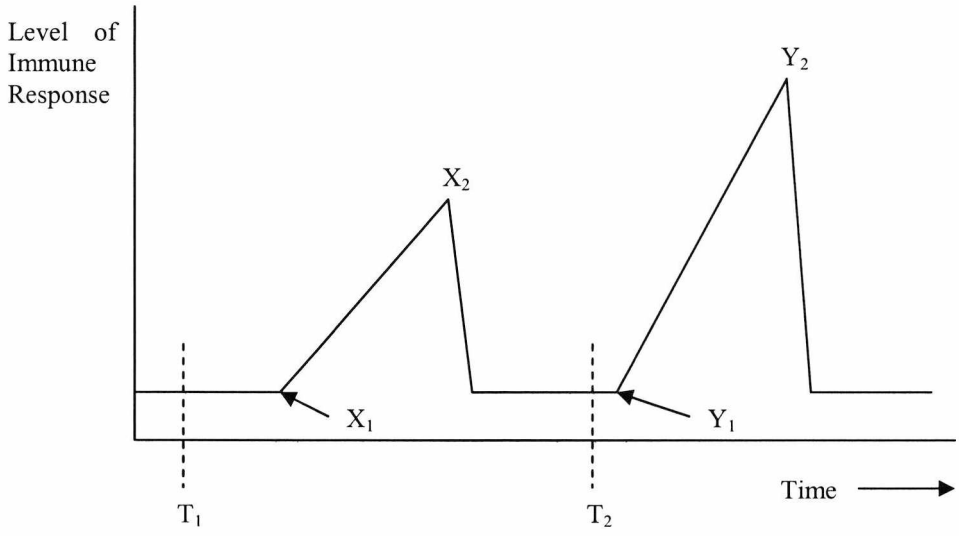

Figure 3.1. Primary and secondary immune response. Unfamiliar antigen introduced at time T_1 producing the response at X_2 but notice the lag between T_1 and X_1. The same antigen introduced at T_2 notice the response begins almost immediately at Y_1 and the magnitude of the response Y_2 is greater than that at X_2.

The first indications that such an adaptive system existed were during experimental vaccinations performed by Edward Jenner in the 1790s. There were many theories put forward over the years to explain Jenner's observations until around 1890 when E. von Behring and S. Kitasato demonstrated that protection induced by vaccination was due to the appearance of protecting elements in the bloodstream and named these agents *antibodies*. It is this model which underlies most of the theory today. More recently, Burnett proposed the clonal selection or clonal expansion principle (Burnett, 1959). Given each cell (now known as a B-cell) expresses only one type of antibody, upon stimulus by an antigen, the cell will begin to proliferate and secrete antibodies, a process called clonal expansion which was formalised in 1959.

It is now accepted that there are two types of cells in the adaptive immune system. These are collectively called *lymphocytes*. These two types of cell are called *B-cells* and *T-cells*. Both have slightly different functions but when called upon to protect the host, both work closely together. Lymphocytes express receptors which are found on the surface of these immune cells. Any molecule capable of binding to one of these receptors by chemical interactions is called an *antigen*. It is this binding that causes a lymphocyte to become activated. The match between the receptor and antigen need not be exact and so when a binding takes place it does so with a strength called an *affinity*.

If this affinity is above a certain threshold the bound immune cell becomes activated in some way. With this now defined it is possible to go into some detail regarding first the production of and then the behaviour of T-cells and B-cells.

3.2.2.1 Creation of Lymphocytes: Gene Libraries

Human DNA encodes in the order of 30,000 genes, but a human body can make B-cells with more than 100,000,000 unique antibody receptor proteins. It stands to reason therefore that human genetic material cannot encode these proteins in the usual way (Janeway, 1994). The enormous diversity of natural antibody shapes is achieved by the way fragments of genetic material are used to encode the receptor and are joined together in individual lymphocytes as they develop, in order to code for the shape of the antibody receptor. This is a process is called *gene rearrangement.* The genes that encode the receptor of the cell are inherited by the cell as gene fragments. These fragments are inherited randomly and are then concatenated to generate diversity. In addition to this, random DNA bases are joined to the ends of these pieces promoting further diversity.

However, this process is not entirely straightforward. Antibody molecules are made from two different regions, called the light and heavy chains. These in turn require different genes to encode them to preserve their structure. Hence, the inherited DNA sequences cannot be concatenated entirely randomly, instead the gene fragments are taken from specific *gene libraries* (Cayzer et al., 2005). These are libraries of genetic material that can encode a particular segment of the receptor. The result of this mechanism is a genetic code that, while coding for a legally sequenced protein, that protein will have sections that are of random shapes, promoting diversity of the lymphocyte set in the body. This random process will not always result in a legally shaped receptor being produced, and it has been calculated that only one cell in nine will produce a legal genetic sequence. The remaining eight out of these nine cells will be removed, as they are of no use.

Given that a T-cell may bind to any cell and, when T-cells are produced their receptor is in a random configuration, why do T-cells not bind to cells of the host? In the natural system, the body is purged of these *autoreactive* cells before they are able to circulate and initiate an *auto-immune response* by a process called *negative selection* (Forrest, Perelson, Allen, & Cherukuri, 1994). This process is important in the context of both natural and artificial immune systems but a full explanation is reserved for section 3.2.2.4.

3.2.2.2 B-cells, Antibodies and Clonal Selection

Like all immune cells, the B-cells are generated in the bone marrow as described above. A natural B-cell will express in the order of 10^5 antibodies (and therefore receptors) on its surface, each one of a shape encoded by the B-cell's genetic makeup and each one of a single shape common to that B-cell. All antibodies produced by a singe B-cell will therefore bind to the same set of molecular patterns. B-cell antibodies are said to be divalent and bi-functional. They are divalent because they may bind to two antigens at once, hence the two "arms" of the Fab region (see Figure 3.2) and bi-functional because as well as the Fab region binding to antigenic patterns, the Fc portion may bind to special receptors on the surface of other immune cells. Unlike the killer T-cell (section 3.2.2.3) it is not the antibody's job to destroy pathogens, instead antibodies will simply bind to an antigen.

This binding may fulfil two roles. The primary role is to tag the antigen for destruction by cells of both the innate and adaptive immune system. It is said that the antibodies opsonise the pathogen. When antibodies opsonise an invader they do so by binding with their Fab regions, leaving the Fc regions available for binding with receptors on the surface of other immune cells such as macrophages. Thus illustrating just one way of many by which the innate and adaptive immune systems interact.

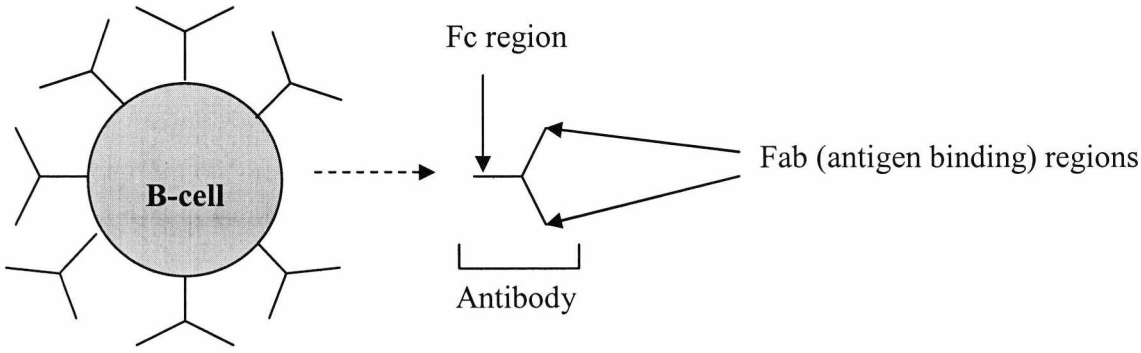

Figure 3.2. Diagram of a B-cell. Numerous antibodies can be seen on its surface and the Fab and Fc regions of an antibody are labelled.

Importantly, in the case of viruses, an antibody binding to a virus cell may have the additional effect of stopping the virus working altogether. In this case the antibody interferes with the cellular receptor on the surface of the virus, neutralising the virus cell by rendering it unable to bind with the surface of a host cell. As a result, the virus cell can no longer make its way inside the host cell and is already tagged for destruction by phagocytes (components of the innate system efficient at destroying pathogens).

When a B-cell binds to an antigenic pattern (and therefore to its *cognate* antigen) the B-cell will proliferate to create many more identical B-cells. It takes around twelve hours for a B-cell to grow and divide into two. Once stimulated the period of proliferation can last around a week. Thus a single cell can produce 2^{14} or around 16,000 cells are produced, all with the same shaped receptor. However, the higher the affinity between the B-cell receptor and the antigen, the greater the rate of proliferation. Given higher affinity B-cells produce more clones than lower affinity ones, the high affinity clones tend to dominate. This process is called the *clonal selection* principle. It is believed to be correct and is a process similar to Darwin's natural selection. The clonal selection principle has its own algorithm in AIS (de Castro & Von Zuben, 2000) and is common in the literature (section 3.5) and is the process much of the AIS work in this thesis is built upon.

Once B-cells have undergone proliferation they begin maturation, a process that occurs in three steps: isotype switching, affinity maturation and a decision to become a plasma or memory cell. Generally AIS are not concerned with isotype switching, but the remaining two pathways are important and inspire ideas used in numerous algorithms. *Somatic hypermutation* as part of the *affinity maturation* step can be a significant part of an AIS algorithm. Somatic hypermutation occurs in B-cells to mutate the part of the B-cell's genetic code that determines the shape of the antibody receptor. The mutation may increase or decrease the affinity of the antibody receptor with the B-cell's cognate antigen. A B-cell will only continue proliferating during this stage if it is continually re-stimulated and so an increase in affinity between the B-cell and the antigen causes the B-cell to become more stimulated, thus increasing the rate at which it proliferates. There is a positive feedback mechanism working to exert strong selective pressure to create better and better B-cells. This process is called affinity maturation. The final step of the affinity maturation process is the choice between the B-cell becoming a memory cell or a plasma cell. Plasma cells are antibody factories and secrete antibodies at a phenomenal rate and so do not last long. The other choice is to become a memory cell. Most memory cells have higher than average affinity with their cognate antigen, making them prime candidates to remember this antigen for the future. These memory cells, as well as being fine-tuned for a specific kind of antigen require less of a stimulation to become activated in the future, thus increasing the speed and efficiency of the immune response when the pathogen is next encountered (recall Figure 3.1). Some evidence suggests that memory cells are simply long lived while others indicate a memory cell is short lived but periodically proliferate when re-stimulated, possibly by other cells in the

immune system expressing antigen-like patterns. For more details of these immune memory theories see section 3.2.2.5.

3.2.2.3 T-cells

T-cells are similar to B-cells in appearance, and like B-cells they are created in the bone marrow (Weissman & Cooper, 1994). They also display a receptor on their surface, called a T-cell receptor (TCR), which is produced using the same gene-rearrangement technique as B-cell receptors. The TCR uses the same binding mechanisms for attaching to other cells as an antibody. However, unlike B-cells they do not mature in the bone marrow but instead migrate to the thymus (Figure 3.3). It is in the thymus that a process of negative selection occurs (Marrack & Kappler, 1994), the purpose of which is to delete any T-cells capable of reacting against the host and thus generating a *repertoire* of cells tolerant of self. Negative selection is explained in some detail in section 3.2.2.4.

Also, in contrast to B-cells, the TCRs may only recognise proteins whereas an antibody's binding region can recognise any organic molecule. In addition, an antibody receptor may bind to two cognate antigens at once, whilst the TCR may only bind to one. Upon activation the TCR stays attached to surface of T-cell. There are thought to be three different types of T-cell: the *killer T-cell*, which is the subject of the following paragraph, the *helper T-cell* and the *suppressor T-cell*, which may suppress the activation of another T-cell once the pathogen has been removed.

B-cells, and especially killer T-cells, are effective weapons against living cells and so the immune system has evolved to keep these under control. That is, T and B-cells require activation before they may function. One route to activation is via the helper T-cell. A helper T-cell will become activated when it is stimulated by its TCR binding with its cognate antigen when this is properly presented by an APC such as a macrophage (another member of the innate immune system). The T-helper needs one other stimulation to become activated. This signal is the recognition of the self MHC protein expressed on the surface of any APC and non-specific and so will activate any lymphocyte. Once a T-helper cell is activated it will proliferate and build up a great many clones all with the same shaped TCR. This activation takes some time, from eight to twenty four hours, during which time the APC will bond to the T-helper cell (Sompayrac, 1999). Once activated, as well as binding with other lymphocytes to activate them, a T-helper cell acts as a *cytokine* factory. Cytokines are chemical messengers that regulate the immune response. These messages may be relayed between different elements of the immune system, up-regulating the response of cells when

needed and down-regulating when the response is no longer required. Cytokines are incredibly important in the immune response but rarely used in AIS although one exception is the work of Tarakanov (Tarakanov et al., 2003).

3.2.2.4 Negative Selection and Two-Signal Activation

Given that a T-cell's TCR can be produced in a fairly random shape, and this TCR is capable of binding to any suitably shaped molecule while killer T-cells are potentially very efficient, there must be a process to ensure no T-cells are allowed to mature if their TCR may match proteins expressed by the host organism. For the immune system to behave properly it should distinguish between these patterns, *self-antigens*, from everything else, *nonself-antigens*. Understanding how the immune system discriminates between these, and therefore resists *autoimmune disease* (Steinman, 1994) is called the self/nonself discrimination problem (Tauber, 2002).

Once created, immature T-cells mature in a part of the body called the thymus where the chance of encountering a foreign antigen is thought to be negligible. The interior of the thymus is still somewhat mysterious to immunologists but it is thought to present innumerable proteins typical of those expressed by the host. When an immature T-cell enters from the blood stream it begins to proliferate and mutate (similar to the clonal expansion of B-cells but on a much reduced scale). The thymus is populated by large numbers of antigen presenting cells and, as the thymus is protected by a blood-thymic barrier, it is assumed that these APCs will only present proteins representative of the host, *not* any foreign material. Given these assumptions, when in the thymus an immature T-cell binding with any protein pattern has matched a self-protein, the T-cell then dies by *apoptosis* (programmed cell death) thus purging the thymus of any T-cells capable of causing auto immune reaction. So it can be assumed that the only cells able to leave as mature cells are those that are not capable of binding while in the thymus. They are only capable of binding with *non-self.*

While immature and in the thymus, a process of positive selection also takes place. A T-cell can only recognise antigen if presented by an *antigen presenting cell* (APC). These APCs express self-MHC on their surface and so for a T-cell to match, bind and become activated it must be able to recognize the MHC unique to the host. Any T-cell not exhibiting significant interaction with this self MHC molecule will also die by apoptosis as it could never become activated when fully mature. With positive and negative selection combined, only these T-cells capable of binding to self MHC but not

any other self-protein become mature T-cells. The full process is represented diagrammatically in Figure 3.3.

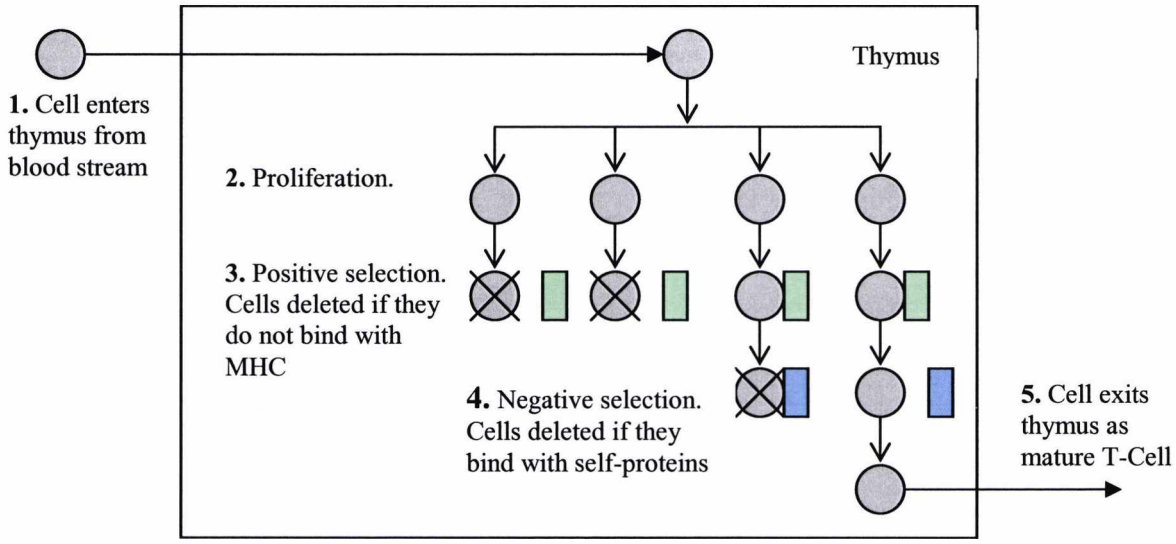

Figure 3.3. A T-cell's maturation in the thymus

It should be noted that it is not possible to present every single unique self protein in the thymus and so some self-reactive cells will escape into the periphery. The thymus is one small part of the host and it would be impossible to remove all potentially self-reactive cells based on negative selection alone. There are a number of processes designed to counter this, one of which is the two-signal model as, described in (Matzinger, 2002b), which is used for purging self reactive cells once they have left the thymus. A T-cell needs two signals to become activated. The recognition of an antigen by a T-cell is said to be signal one. The second signal or co-stimulation signal is a confirmation and is given to the T-cell by an APC upon proper presentation of the antigen. If the T-cell has received signal one in the absence of signal two, it has bound to an antigen not presented by an APC and therefore assumed to be part of the host. The T-cell is removed. This extra layer of protection allows potentially self-reactive cells that have survived negative selection to exist in the periphery without beginning an autoimmune reaction. These basic negative selection mechanisms have been challenged by the Self Assertion viewpoint (Bersini, 2002) and the Danger Theory (Matzinger, 1998) although these are not yet widely used in AIS.

Like T-cells, B-cells must also be prevented from matching to antigenic patterns of the host. It is thought that B-cells, which do not migrate to the thymus when immature, undergo a similar process to T-cells in the bone marrow, but most of their maturation is done in the periphery. Like T-cells, it is thought that a binding in the absence of a

second signal, in this case that given by a T-helper cell, may induced apoptosis (cell death). It is thought that a sustained, weak stimulation such as that given by self antigen is required to promote tolerance in B-cells.

3.2.2.5 Immune Networks and Memory

This subsection expands the notion of immune memory, as described in the final paragraph of section 3.2.2.2, and seen in Figure 3.1. There are two main competing theories regarding the mechanisms with which the immune system remembers past encounters. The first mechanism, long-lived memory B-cells, is part of the clonal selection principle. To reiterate, certain B-cells with higher than average affinities with antigens may as part of their maturation process become one of these long-lived cells.

The second is the *immune network theory*. This theory, originally proposed by Jerne in 1974 (Jerne, 1974), sees the immune system as a more dynamic system than in the memory B-cell/clonal selection principle, in which the immune system is generally at rest and activated only when invaded by an antigen. The immune network (or *idiotypic network*) theory proposes that the immune system presents dynamic behaviour even in the absence of stimulus, and proposes memory is an emergent property of this system.

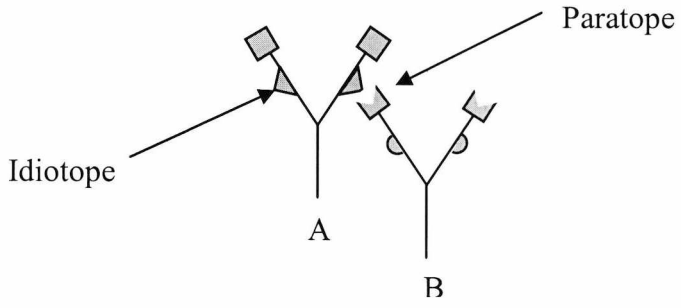

Figure 3.4. Anatomy of an antibody according to the Jernian Immune Network Theory. Here antibody B is stimulated by the binding via its paratope to the idiotope of antibody A. Antibody A is likewise suppressed by this action.

This theory was proposed to explain the observation that the immune systems of one species of animal can be stimulated to recognise parts of antibody molecules made by other species of animal. Jerne hypothesised that within an immune system, any antibody molecule could be recognised by a set of other antibody molecules. To explain this, Jerne hypothesised that each antibody consisted of two regions called a *paratope* and an *idiotope* (Figure 3.4). These regions may not necessarily be the same shape but the idiotope should represent a pattern expressed by an antigen to be of use. It was thought

that one antibody might be stimulated by that antibody's paratope binding to its complementary idiotope on another antibody. After this binding occurs it is thought that the stimulation will cause the antibody to proliferate, producing offspring with the same receptor shape such that when the parent cell dies, the information it carried would not be lost. The converse of this is thought to be true of an antibody binding via its idiotope, in that this action will lead to tolerance and suppression. Thus the immune system may be thought of as a vast interconnected network of cells stimulating and suppressing each other to maintain an immune memory.

The use of this theory in AIS allows the creation of AIS algorithms such as that described in (Timmis, 2000) among many others. Further explanation of the immune network theory and AIS is reserved for section 3.4.3.3 and 3.5.2.1.

3.2.3 Adaptive Immunity and Engineering

In the previous subsection, the immune principles required for an understanding of AIS were reviewed. It can be hard to tease from that text the manner in which the immune principles may be put to work in an engineering scenario. It is therefore the task of these few paragraphs to make clear the bridge between the natural world and engineering by highlighting the main concepts required for an understanding of the following section.

The most important concept is that that a lymphocyte may bind to another cell via a specifically shaped receptor molecules displayed on the surface of that lymphocyte. Different shaped *surface receptor molecules* (SRM) will bind to different shaped patterns on the target cell and as the mechanism of this binding is chemical attraction, the binding may take place with differing strengths. The total set of shapes an immune system can recognise is referred to as the *potential repertoire*, while the set of shapes it can actually recognise (i.e. there are cells in the host at that moment with the correct set of SRMs) is the *actual repertoire*. If there is a high degree of match or *complementarity* between the surface receptor molecule and the antigen, the bind is strong and the two are said to bind with high *affinity*, whilst a slightly mismatched pair of molecules may still bind but with a lower affinity.

It is from this that the notion of the immune system performing a pattern recognition task is born (an important concept in AIS). This concept is illustrated in Figure 3.5 in which diagram A depicts the best match between the surface patterns of the lymphocyte (L) and the antigen (Ag), and therefore represents a bind with high affinity. Diagram B represents an antigen with a pattern less complementary to the shape offered by the antibody receptor, a binding may still occur but with lower affinity. Diagram C

represents a bad match between the two surface molecules and in this case a binding may not occur at all. Anything a lymphocyte may bind to is called an *antigen*. Any antigen an antibody may bind to is referred to as that antibody's *cognate* antigen. Generally the binding with an antigen will activate the lymphocyte cell in some way.

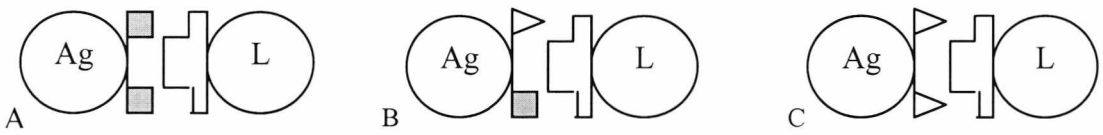

Figure 3.5. Degrees of affinity between a lymphocyte (L) and an antigen (Ag)

The pattern recognition mechanisms above can be combined with the clonal selection mechanisms to perform problem solving. That is, a set or population, of these immune cells can assess their affinities with antigens and if the affinity it high (the match is good) clone or copy themselves. If this cloning is implemented with a rate proportional to the degree of match then there is pressure on the population to converge towards the best match. When this copying of cells is combined with mutation, new areas of the space begin to be explored, allowing for the potential for better and better matches to be made. Combined with the selective pressure, this allows a population of artificial immune cells to become better and better at matching the patterns presented to it.

3.3 Artificial Immune Systems

The field of Artificial Immune Systems (AIS) can be regarded as a machine learning paradigm. Machine learning is the ability of a computer to accomplish a task by learning from data or experience, without having to be specifically programmed to complete that task. Rather the program running allows the computer to achieve the task of learning. Artificial immune systems themselves are defined, in (de Castro & Timmis, 2002a) as:

"*Adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving*"

De Castro and Timmis take this definition of an AIS and explicitly state three criteria that must be included in any AIS algorithm:

1. It must include at minimum a model of an immune component such as a lymphocyte.

2. It must be designed incorporating ideas from theoretical and/or experimental biology.
3. It must be aimed towards problem solving.

Based on these criteria, de Castro and Timmis identify the first work in the field of AIS, which came in 1986, when (Farmer et al., 1986) proposed a model for the immune network theory, stating "*the immune system is capable of learning and memory and pattern recognition. By employing genetic operators on a time scale fast enough to observe experimentally the immune system is able to recognise novel shapes without reprogramming*" thus starting the interest in AIS with regard to machine learning. Around the same time Hoffmann, motivated by a problem regarding neural networks, explored the similarities and differences between the nervous and immune systems (Hoffman, 1986), while Varela et al. contrasted the immune system and neural networks (Varela et al., 1988). Since then, AIS have proved a remarkably adaptive and appealing paradigm. The immune system has been shown not only to be a pattern recognition system, but there are many more attributes possessed by an immune system and therefore by a correctly constructed AIS, such as learning, memory and self-organisation. From an engineering perspective, properties such as anomaly detection, fault tolerance, distributivity and robustness are all appealing.

3.4 Framework for Artificial Immune Systems

If an AIS is to be used for engineering, rather than simulation, principles and processes must be extracted and generalised from the physical workings of the natural immune system. de Castro coined the term "immune engineering" for this process, defined as:

"*A meta-synthesis process used to extract ideas from the immune system in order to build novel computational tools and solve complex problems*" (de Castro, 2001).

The generic process of engineering requires certain fundamental principles and processes. For this reason, (de Castro & Timmis, 2002a; de Castro & Timmis, 2003) proposed an engineering framework to aid this process. The core of the framework for engineering AIS consists of just three elements:

1. A representation for the components of the system.
2. A set of mechanisms to evaluate the interaction of components with the environment and each other (an affinity function).
3. Procedures of adaptation.

This can be seen to be a layered approach, as the problem domain will influence the representation which, in combination with the domain, will influence the choice of affinity function. This layered approach was further generalised by Stepney et al. A generic conceptual framework is proposed (Stepney et al., 2004) and used (Newborough & Stepney, 2005). In the following subsections, the layered approach of de Castro and Timmis is taken and aspects of each of the three elements of the framework are described in turn.

3.4.1 Representation

For an AIS to function it is imperative to have a way to represent data. Representation schemes abstract from the biological notion of cell receptors to allow this data to be encoded. Recall from section 3.2.3, one component of a natural immune system is a cell's *receptor*. An AIS will typically involve the manipulation of a set of these artificial lymphocytes with the shape of an artificial receptor being described by a set of features. Recall that each natural immune cell, be it a B-cell or a T-cell, will posses a specifically shaped receptor. T-cells have a T-cell receptor (TCR) while B-cells express antibodies, but receptor encoding is of interest, and so no distinction is usually made between these and the more generic term, receptor[2], is commonly used. The shape of a lymphocyte's receptor is encoded using its genetic information. An artificial immune cell may include information in a similar way, where its characteristics are encoded by a vector of attributes, or features (Figure 3.6). In the artificial domain, antigens and antibodies generally share the same encoding scheme, which allows a measure of similarity or distance to be computed between them (see following section). In the case of Figure 3.6, both antigen and lymphocyte receptors are described in binary, however this may not always be the case. For problems that are not so basic, it is quite possible to store the "phenotype" of a receptor in the feature vector, thus abstracting further from the biology.

Each artificial cell will represent a point in the solution space or search space; a notion biologists refer to as a location in *shape space*. The notion of shape space was introduced by (Perelson & Oster, 1979) while performing an investigation into the size of immune repertoire, and has been adopted in theoretical immunology to model and

[2] There are cases of algorithms where this distinction is made and the two types of cell do behave differently, but this is rare.

study the interactions between immune components. (Perelson & Oster, 1979) proposed it is possible to describe the shape of an antibody receptor m with a set of L physical parameters (length, width, height, electric charge etc. in the binding site). Thus a point in L dimensional space describes the shape of the receptor. Mathematically $m=<m_1,m_2...m_L>$ is a point in shape space, $m \in S^L$. The entire landscape, including all legal sets of values (values of m) is called shape-space. Given an immune system with N different antibodies, the shape-space for that immune system contains N points lying within a volume, V, of finite size as only certain lengths, widths, electric charges, etc. for certain features are possible. In order to bind to a molecular pattern, a receptor must have significant proportions of its surface complementary in shape with that pattern, however the match does not need to be exact.

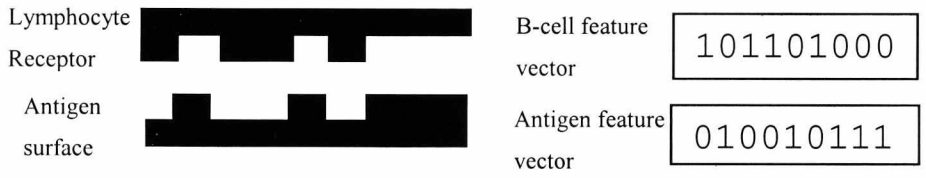

Figure 3.6. Analogy between B-cell receptor and artificial immune cell feature vector. The feature vector for the artificial cells could, for example, be a Boolean representation representing the presence or absence of words in a document.

The notion of shape space has been adopted in the AIS community as it can be thought of as a synonym for solution space when translated into the artificial environment. Likewise, a set of L parameters may specify a potential solution to a computing problem. Thus, an antibody may be described in a similar way. i.e. $Ab=<Ab_1,Ab_2...Ab_L>$ to represent a point in space. This representation is referred to as a *feature vector* or *attribute string*. The actual anatomy of this encoding is dictated by the problem domain. The antibody Ab may contain a set of

- Numbers (real numbers, integers, etc.).
- Symbols, built from a finite alphabet such as binary digits known as a Hamming shape-space.

In Hamming space it is possible to calculate the size of the potential repertoire N, (total number of unique points that can be represented in shape space) using Equation 3.1.

$$N = k^L \qquad \textbf{3.1}$$

Where k is the size of the alphabet (in binary Hamming space for example, K=2) and L is the length of the attribute string. The encoding scheme chosen will dictate the legal ways in which two cells are compared by an affinity function as examined in the following section.

3.4.2 Affinity

Artificial affinity measures abstract from the notion of a lymphocyte receptor binding to an antigenic pattern to allow for a means by which to assess the degree of match between a candidate solution and the data. In order to bind to an antigenic pattern, a natural receptor must have a significant proportion of its feature vector matching with the antigen, however the match does not need to be exact. Given an artificial cell, the point in space it represents will be surrounded by a volume, within which the antibody may interact with an antigen. This surrounding volume's size is characterised by ε, called the cross *reactivity threshold*, and the area inside this volume is called the *recognition region*. A mathematical function may be employed to determine whether two items lie within each other's recognition region, and determine the strength of matching if they do. (Hart, 2005) investigates three different shaped recognition regions with regard to AIS, although this is specifically with regard to immune networks.

This mathematical function, commonly called *distance measure* or *affinity function*, may be defined to determine a measure of similarity between an immune cell and an antigen or between two immune cells. The affinity function performs a mapping between two *attribute strings* (or feature vectors) to a number that represents the *affinity* or *degree of match* between those strings. Recall how this reflects the natural system where *regions of complementarity* are needed to provide enough physical binding force between an antibody's receptor and an antigen to pull these two cells together. In Figure 3.6, the region of complementarity extends over the entire length of the receptor. The match between the receptor and antigen need not be exact. These terms are shown in Figure 3.7.

It is possible to view affinity as a general term that relates the quality of one element of the set of artificial cells to a pattern external to that set. If the value calculated by the affinity function is greater than the threshold ε, when the function measures similarity (or distance) the antigen is said to be within the recognition region of the immune cell or that the lymphocyte will *recognise* the antigen. Given this threshold, in combination with Equation 3.1, it is possible to calculate the minimum number of antibodies required to cover the whole shape space, although details are out of the scope here.

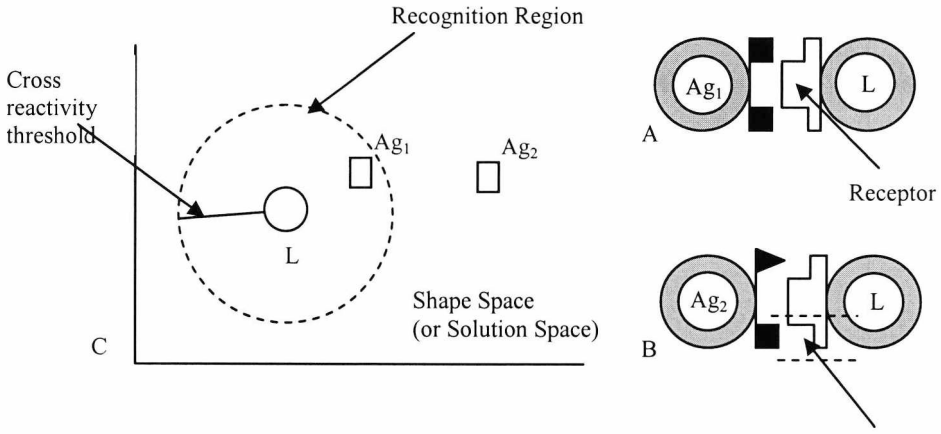

Figure 3.7. Diagrammatic depiction of recognition region and cross reactivity threshold. (A) depicts a lymphocyte (L) binding with high affinity to an antigen (Ag1), whereas (B) depicts a binding between an antigen (Ag_2) with fewer regions of complementarity compared with the same lymphocyte. This results in a bind with lower affinity, and so L may not become activated by Ag_2. (C) shows the relative positions of L and the complement of Ag_1 and Ag_2 in shape space. Ag_1 is recognized by L as the affinity between the two is higher than the affinity threshold.

Although in the natural world complementary molecular patterns will bind, in AIS it is common for the affinity between two attribute strings to be proportional to the *similarity* between them, therefore inversely proportional to the distance between them. Numerous standard measures of distance can be used for this of which three examples are shown below. Given that antibody $Ab=<Ab_1,Ab_2...Ab_L>$ must be compared with antigen $Ag=<Ag_1,Ag_2...Ag_L>$, the assessment of affinity when using a numerical attribute string may be done by Euclidean distance (in Euclidean space), or Manhattan distance (in Manhattan space). This is shown in Equation 3.2 and Equation 3.3 respectively, where D denotes distance.

$$D = \sqrt{\sum_{i=1}^{L} (Ab_i - Ag_i)^2} \qquad 3.2$$

$$D = \sum_{i=1}^{L} |Ab_i - Ag_i| \qquad 3.3$$

A popular distance measure is that of Hamming distance. Hamming distance is a count of the number of positions in two strings of equal length for which the corresponding elements are different. A distance (D) between two binary vectors in Hamming space can be computed by Equation 3.4.

$$D = \sum_{i=1}^{L} \delta_i \quad \text{where} \quad \delta_i = \begin{cases} 1 \text{ if } Ab_i \neq Ag_i \\ 0 \text{ Otherwise} \end{cases} \tag{3.4}$$

Hamming space also affords a more biologically plausible manner in which to assess distance. Recall from section 3.2.2, for an antibody/antigen binding to occur regions of complementarity are required; this can be modelled by the *r-contiguous bits* measure. This may be of use in a situation where contiguous regions of complementarity in the artificial cell feature vector should be favoured. Equation 3.5 shows the r-contiguous bits distance measure where D_H is the Hamming distance as computed by 3.4 and l_i is the length of each complementary region with two or more complementary bits.

$$D = D_H + \sum_i 2^{l_i}, \tag{3.5}$$

Other similarity measures in Hamming space include those of (Harmer & Lamont, 2000) and (Rogers & Tanimoto, 1960).

Finally, it should be noted that a *binding function* may be used to bias the result from any of these distance equations. Where it is not required that the binding strength varies in proportion to affinity this function may, for example, penalise a particularly low or high value for affinity as the algorithm demands. For example a binding function that only allows activation once ε is reached, where D_H is the Hamming distance as calculated before, is shown in Equation 3.6.

$$D = \begin{cases} 1 \text{ if } D_H > \varepsilon \\ 0 \text{ Otherwise} \end{cases} \tag{3.6}$$

In this section it has been stated that AIS algorithms tend to have a set notion of a cross reactivity threshold, also called recognition threshold, affinity threshold and activation threshold. That is, any affinity value greater than this threshold (and therefore distance value smaller than this threshold) will characterise a recognition event. The processes described in the following section may depend either on whether the recognition event occurred or on the calculated value of affinity.

3.4.3 Processes

Having described the artificial representation of a lymphocyte and a notion of similarity between lymphocytes and antigens, the processes that manipulate populations of lymphocytes can now be described. These processes abstract from the natural methods by which lymphocyte populations change over time and form the basis by which artificial populations adapt to changing conditions and stimuli, generally offering the system adaptability and learning mechanisms. The literature describes a diverse selection of AIS algorithms, not only with regard to the problems that they solve, but also in the mechanisms used. This can be attributed to the vast array of mechanisms displayed by the natural immune system, which offers authors a wide choice when determining how to best solve a problem using immune principles. These mechanisms fall roughly into four groups:

1. Bone marrow
2. Negative selection
3. Immune network
4. Clonal selection

Some AIS algorithms use exclusively one of these techniques, although it is not uncommon for some to share characteristics. For example, immune network models tend to involve cloning procedures which form the basis of clonal selection algorithms. Clonal selection algorithms may use negative selection techniques to control the initial generation of cells. All will need to generate an initial set of cells, which might be thought of as a bone marrow model to a greater or lesser extent. In the following sections, discussion of negative selection and immune network models will be short with the reader being referred to the literature for all but the basic characteristics of these. By contrast, the introduction to clonal selection algorithms will be more comprehensive since this thesis uses this type of algorithm extensively. Before these three are described, the following subsection describes a technique to generate initial populations of cells.

3.4.3.1 Bone Marrow

Bone marrow processes are those by which immune cells are created and as such computational abstractions of these allow for the creation of artificial cells. Bone marrow algorithms are used during the initialisation of the population or to replace cells that have otherwise been removed from the population, unlike negative selection,

immune networks or clonal selection they do not manipulate the population as a whole. All natural immune cells are created in the bone marrow; hence a process to create new artificial cells is often referred to as a bone marrow process. The goal of such a process is to produce a randomly generated feature vector. There are two constraints generally placed on the generation of this vector:

1. The vector must be of legal length.
2. The elements of the vector must be drawn from the correct alphabet or legal range.

Recall from section 3.2.2.1 that gene libraries are libraries of genetic material that can encode a particular segment of the receptor which inspires a gene library algorithm. It is possible that a cell feature vector only allows certain values or certain values in certain positions. A gene library algorithm will accomplish this by randomly assigning a value of the correct type to a particular point in a feature vector. In its simplest form a bone marrow model will use a random number generator to randomly produce a string conforming to both these constraints. It is quite possible, that the alphabet is more complex than just holding a binary digit, a symbolic value or real valued number. In this case a gene library may be used. These gene libraries are used where the feature vector requires a predefined structure or random generation of the feature vector is otherwise inappropriate (Cayzer et al., 2005). Gene libraries are used later this thesis and further specific implementation details will be given where necessary.

3.4.3.2 Negative Selection

Recall from section 3.2.2.4, negative selection is a process by which newly generated cells are censored such that only those incapable of reacting against the host are allowed to mature. Computational abstraction of this natural process results in algorithms that are able to produce a set of artificial detectors that do not match a certain class of data. The process of negative selection is not the concern of this thesis, but it is used by a number of algorithms reviewed later and is therefore briefly summarised as follows.

This process of negative selection is a major component in a number of AIS algorithms. The goal of these negative selection algorithms is to generate a set of detectors (antibodies) that do not match any element in a set of known patterns (antigens). In this class of algorithm, a set of detectors are compared against a set of patterns corresponding to self and the resulting affinities evaluated. Any detector with an affinity to an element of the self set, above a threshold, will be eliminated. Thus only

detectors capable of recognizing only non-self examples are left. The negative selection algorithm as described in (Forrest et al., 1994) can be summarized as follows. It should be noted that this is a generic negative selection algorithm whose uses are not just constrained to the network intrusion detection problem as outlined in (Forrest et al., 1994).

```
1. While mature detector set is less that required size, do:
      a. Initialise: Randomly generate detectors, and place them in
         a set of immature detectors.
      b. Affinity: Determine the affinity of all immature detectors
         with all elements of a set of known patterns.
      c. Generate repertoire: If the affinity of an immature
         detector is greater than a threshold then discard
         detector, otherwise add to set of mature detectors.
```

Pseudocode 3.1. Negative selection algorithm from (Forrest et al., 1994).

A *positive selection* algorithm works in a similar way. Positive selection is the mechanism found in the natural system to only allow lymphocytes capable of recognising self MHC to pass into maturity. In the case of positive selection, the immature detectors are kept only if they are similar to an element in a set of known patterns and thus produces a set complementary to a negative selection mechanism. Negative selection has been criticised when used in isolation for data mining (Freitas & Timmis, 2003), further discussion of this is reserved for later.

3.4.3.3 Immune Network

Immune network models attempt to depict a population of cells undergoing constant activity, thus computational abstractions grant dynamics to a population of cells allowing the population to react to external stimuli. The immune network theory is based on the notion that cells are continually recognising each other; the idiotope of some being recognised by the paratope of others thus stimulating the cell that has recognised the idiotope. This theory in terms of the natural system has been somewhat discredited, even being described as "absurd" (Langman & Cohn, 1986). However, it is the theory that has received the most theoretical modelling, as it is possible to model the interaction within populations of cells using mathematics. This modelling has been attempted in two separate ways: continuous and discrete. The continuous models of Jerne (Jerne, 1974), Farmer (Farmer et al., 1986) and Varela (Varela et al., 1988) uses differential equations to model the way lymphocyte concentrations may vary based on stimulation, suppression, cell generation and cell death. Of these three approaches the model of Farmer is the most significant to AIS, as (Farmer et al., 1986) describes the

mathematical modelling of the dynamics of populations of bit string lymphocytes in Hamming space (section 3.4.2) thus it has many commonalities with the field of AIS.

$$s_i = \sum_{j=1}^{M}\left(1 - D_{i,j}\right) + \sum_{k=1}^{n}\left(1 - D_{i,k}\right) - \sum_{k=1}^{n} D_{i,k} \qquad 3.7$$

Based on this, (Timmis & Neal, 2000) proposed the immune network based learning algorithm, RAIN, which uses the Equation 3.7 to determine a stimulation level (s) for each cell (i) in the population during each iteration of the algorithm. In this equation, M is the number of antigens, n is the number of connected B-cells and $D_{i,j}$ is the distance between antigen j and B-cell i.

```
1. Initialise: Create an initial network
2. Antigenic presentation: for each antigen do
     2.1. Clonal selection and network interactions: determine
          stimulation level of each cell according to Equation 3.7
     2.2. Metadynamics: remove cells with low stimulation level until
          the number of resources allocated is less than the maximum
          number allowed
     2.3. Clonal expansion: most stimulated cells reproduce in numbers
          proportional to their stimulation level
     2.4. Somatic hypermutation: mutate each clone with a rate
          inversely proportional to its stimulation level.
     2.5. Network construction: select mutated clones to incorporate
          into the network
3. Cycle: repeat step 2 until stopping criterion is met.
```

Pseudocode 3.2. RAIN, an example of an artificial immune network algorithm.

Individual cells with a high stimulation are selected to undergo clonal expansion and those which have a low stimulation are selected to be removed from the population. To deter unbounded increases in population size, a notion of resources is used to control the size of the population. In this case there are a predefined number of resources in the network for which each cell must compete. These resources are allocated according to a cell's stimulation level. Notice in Pseudocode 3.2 how clonal expansion, a concept to be described in the following section, is used. Although unlike the clonal selections algorithms described later, RAIN and other immune network models use the idea of cell proximity or neighbourhoods within the population of cells to assess stimulation, whereas in most clonal selection algorithms the stimulatory pressure tends to be external only (via a presented antigen). The RAIN algorithm is summarised in Pseudocode 3.2. The stopping criterion for such an algorithm is generally domain dependent, although typically the algorithm will cycle for a set number of iterations. A

review of the uses of network based algorithms for data mining can be found in section 3.5.

3.4.3.4 Clonal Selection

The abstraction of the natural clonal selection and hypermutation mechanisms allow for a population of artificial cells to change and adapt over time and thus often form the basic learning mechanism in a clonal selection based AIS algorithm. Upon activation, the two types of lymphocyte (B-cells and T-cells) will behave differently. Considering first a B-Cell, whose task is to tag an antigen for destruction by another immune cell. This B-cell must bind to the antigen, and stay bound until the antigen can be destroyed. It is quite possible for no B-cells in the body to have a high affinity with this antigen. For this reason, an activated B-cell will begin a process of cloning and receptor mutation called *clonal selection*. Strong selective pressures during this proliferation process have the effect of maximizing affinity with the antigen, and so increasing the effectiveness of the immune response. In AIS, an activated immune cell may adapt to new data in a similar way. Upon activation the artificial cell may undergo a process of cloning with a rate proportional to the antigenic affinity. Each new clone is mutated with a rate inversely proportional to the affinity with the antigen. Both of these processes have the goal of gradually moving cells in the population closer to the antigen within the solution space. An adaptation process such as this is a common paradigm found in many Evolutionary Algorithms (EAs). After activation, a small number of clones with high affinities will survive to provide some memory of the event in the form of *memory cells*. How this is done in the natural world is sill a point for debate, whether it be simply long lived cells, or the more complex *immune network* (Jerne, 1974). Indeed (Robins & Garrett, 2005) lists six such theories and even evaluates them via simulations. The process described above is illustrated in Figure 3.8.

It should be noted here that, in the natural system, B-cells undergo mutation and so most clonal selection algorithms tend to refer to a population of B-cells as populations of these show adaptability. This mutation changes the shape of the cell receptor and, combined with the fact that only receptors capable of recognising antigen will proliferate and so exerts strong selective pressure on the population. However, the mutation is believed to be guided in B-cells that is, the greater the match between the receptor and antigen the smaller the rate of mutation. This results in the B-cells with receptors presenting higher affinities with the antigen cloning more, thus increasing the average affinity of the population with the antigen. This process of cloning and

mutation is called *affinity maturation*. It can be seen that this process is analogous to neo-Darwinian evolution as the affinity maturation process encompasses reproduction with inheritance (cloning), genetic variation (mutation) and natural selection (selection) for reproduction based on affinity.

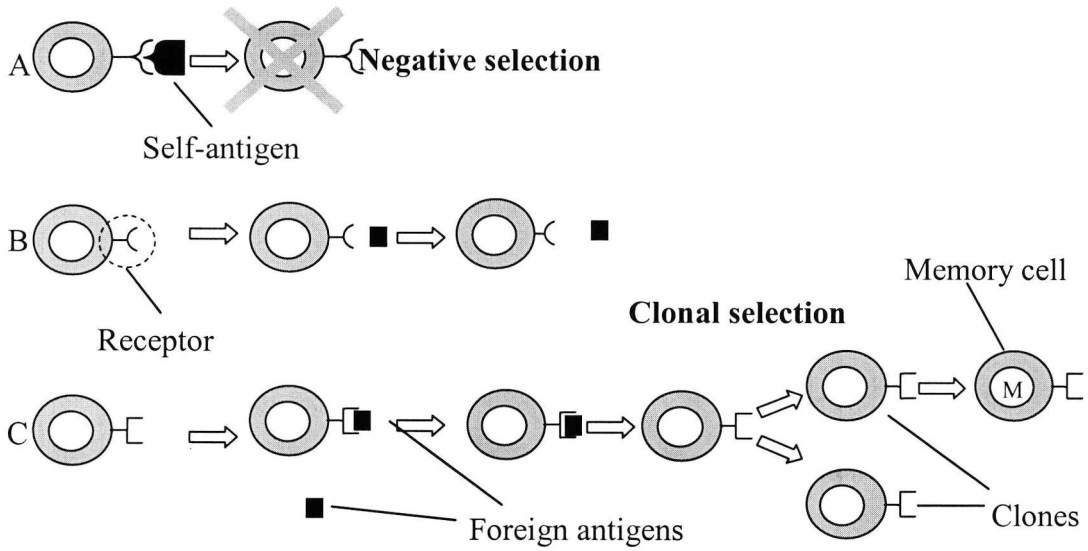

Figure 3.8. Diagrammatic representation of B-cell maturation process. (A) Cell receptor matches pattern belonging to the host, cell is removed by negative selection. (B) Cell receptor does not match antigenic shape and so cell is left un-stimulated. (C) Cell receptor matches foreign antigenic shape (unlike cell in B) and so is selected for cloning. Cell becomes activated and produces clones, some of which become memory cells. A modified version of a diagram taken with permission from (de Castro & Timmis, 2002a).

Two principles can be identified as important to a clonal selection algorithm:

1. Many cells may be selected to perform affinity-proportional cloning.
2. The mutation rate of each clone is inversely proportional to the cell's antigenic affinity.

Thus, if a subset of all immune cells are required to reproduce, then a method of selection for cloning is required, so too is a method of mutation. It is common for a cell to clone only if it has affinity with an antigen over a specific threshold, or for the single best or top n members of a population to clone (when ranked by affinity).

When cloning occurs, mutation also tends to take place on the clones. Mutation is important as it promotes diversity in the population and so encourages exploration. It will reduce the chance that a population will converge prematurely to a local optimum. Recall that mutation in a clonal selection algorithm is typically performed with a rate inversely proportional to affinity with an antigen. The most common scheme for mutation is a *random mutation* but other kinds of mutation such as *inversive mutation* may also be used. A random mutation will simply pick a point in the cell's feature

vector at random and change that value for one drawn from a legal set of values. The inversive scheme will choose two random points in the vector and swap the values contained there. If locational bias is a positive trait in the representation, multipoint inversive mutation may be used in which two or more consecutive elements are swapped with two or more consecutive elements from another position. It should be noted that unlike random mutation this, scheme cannot introduce new gene values into individuals, and therefore the population, by itself and as such should be used with care.

Regarding a random mutation, consideration of the representation is important. When the representation encodes a binary or other symbolic shape space, the new element should be drawn from the correct alphabet. A gene-library-like process may be employed to ensure only legal values are inserted. When mutating real numbers it is possible that minimum/maximum values may need to be respected. If integer shape space is used both these must be taken into account. Mutation of such a genotype must be done with care, if the values of the phenotype are bounded but still representable by that genotype. For example, when representing a number in binary, a single bit mutation in a highly significant bit location can cause the phenotype (number) to be out of legal range.

The scheme of encoding may also cause issues concerning the variable mutation rate common in clonal selection algorithms. The purpose of this is to provide a localised search around a local or global optimum, however the resultant position in space is highly dependent on the ordering of the bits in the genotype. Changing the value of the single bit at the start of a binary number will have a much greater effect than changing the one at the end (assuming that the most significant bit is encoded first in the string rather than using Gray encoding for example). It may not be sensible therefore to randomly change bits in such a string with no regard for the position. A representation which is the phenotype itself may be preferred. In this case one may randomly add or subtract smaller and smaller values from a random position in the vector as the mutation rate decreases, and vice-versa.

CLONALG (de Castro & Von Zuben, 2000) is an example of a generic clonal selection algorithm which has been widely used with numerous references in the literature (de Castro & Von Zuben, 2001; White & Garret, 2003; Cutello et al., 2005). CONALG is thought to be applicable to tasks such as pattern recognition and optimisation and due to its generic nature is highlighted here as it contains all the principles of a clonal selection algorithm, with few implementation dependent specifics. The pseudocode for CLONALG is presented in Pseudocode 3.3.

```
1. Initialise: Create a random population of individuals
2. Antigenic Presentation: For each antigenic pattern, do
    2.1. Affinity Evaluation: Present antigen to each member of
         population and determine affinity.
    2.2. Clonal selection and expansion: Select n highest affinity
         elements of population. Clone these with rates proportional
         to affinity.
    2.3. Affinity maturation: mutate all clones with rates inversely
         proportional to affinity and add them to population
    2.4. Memory: keep element of population with highest affinity to
         antigen
    2.5. Metadynamics: replace the m lowest affinity elements of
         population with new ones.
3. Cycle: Repeat step 2 until stopping criterion is met
```

Pseudocode 3.3: CLONALG an example of a clonal selection algorithm

One further useful attribute of the clonal selection algorithms is that of immune memory. Recall activated immune cells may differentiate into memory cells, which are long lived cells of utmost importance in future responses to antigenic pattern that are similar to the original stimulus (section 3.2.2.2). It is common to exploit the notion of a memory cell for generalisation and data reduction in an algorithm as will be seen later on.

3.5 Artificial Immune Systems for Data Mining and Other Applications

Since the beginning of the field, AIS have been used for a variety of purposes in a variety of domains (de Castro & Timmis, 2002a; Dasgupta et al., 2003). Among these, three main topic areas can be identified into which the majority of these fit. The three are:

1. Anomaly detection
2. Optimisation
3. Data mining (classification/clustering)

Anomaly detection is briefly reviewed in section 3.5.1. While not the main topic of this thesis, a review of anomaly detection is of special significance interest as it formed the root of AIS as a subject area and allows a short review of the use of negative selection. The main topic of this thesis is data mining, and this is reviewed with regard to AIS in section 3.5.2 and section 3.5.3, in which the latter examines a popular AIS used for data mining. This is followed by the most relevant review in section 3.5.6

which deals with web and text mining using AIS. Optimisation is not covered in this thesis.

3.5.1 Anomaly Detection

Intrusion detection, as summarised in (Aickelin et al., 2004), and the more general area of *anomaly detection* (de Castro & Timmis, 2002b) have historically found favour in the AIS community. The literature also abounds with such disparate topics as hardware fault tolerance in refrigeration units (Taylor & Corne, 2003) or Automatic Teller Machines (Ayara et al., 2005), financial fraud detection (Kim et al., 2003) and aircraft fault detection (Dasgupta et al., 2004) which are all types of anomaly detection.

The idea of a computer network immune system was one of the first uses of an AIS, presumably because it is logical to make the connection from the immune system protecting a biological host to an AIS protecting a computer. For this reason, research into AIS began in earnest in the mid 90s in the realm of computer security for intrusion detection and suchlike. At this time, the negative selection type algorithm proposed in (Forrest et al., 1994), and later updated in (Hofmeyr & Forrest, 1999) prevailed. (Forrest et al., 1994) can be considered the first significant publication in this area. It described a system which learns the usual network traffic found on a particular computer network and then alerts an administrator when an anomaly is detected. The system was later realised and called LISYS (Lightweight Intrusion detection SYStem) (Hofmeyr & Forrest, 2000), which includes the first known use of a co-stimulation signal to positively reward detectors for a correct match.

Kim and Bentley have performed much research in negative selection for network intrusion detection with their idea first outlined in (Kim & Bentley, 1999) and then extended to use a form of clonal selection in (Kim & Bentley, 2001b). The authors then critically appraised their own work, and indeed the notion of using negative selection for intrusion detection in (Kim & Bentley, 2001a), and concluded that such a scheme shows severe scaling problems handling real traffic and so would be impractical in the real world. However, the idea of using AIS for network intrusion detection was not discarded entirely, with the authors using clonal selection replacing negative selection (Kim & Bentley, 2002c). This was one of the first papers to make explicit that the immune system is most suited to dynamic situations rather than static learning tasks. Other Intrusion Detection Systems (IDS) include (Dasgupta & Gonzalez, 2002) which also makes use of a negative selection approach.

To support the conclusions of (Kim & Bentley, 2001a) and the change in algorithm used in (Kim & Bentley, 2002c), the use of negative selection has recently been criticised. In addition, papers such as (Stibor et al., 2004; Stibor et al., 2005a; Stibor et al., 2005b) provide some formal proof that the popular "r-chunk" affinity measure as used in many negative selection based anomaly detection algorithms will suffer scaling problems in real-world scenarios.

With these criticisms in mind, intrusion detection systems have developed over time. CARDINAL (Kim et al., 2005), for example, is a computer worm detection system which takes inspiration for its algorithm from the numerous states of a T-cell during maturation.

3.5.2 Data Mining with AIS

While negative selection has tended to be used for anomaly detection, algorithms based on the immune network theory and clonal selection have been used extensively for data mining. These algorithms commonly contain a generalised pattern matching process as well as natural adaptation and learning mechanisms. The combination of these features conspire to produce some notable data mining systems.

3.5.2.1 Immune Network Solutions

The first work in this area was performed by Hunt and Cooke (Hunt & Cooke, 1996). Citing the advantages of AIS as an adaptive non-linear network whose control is decentralised and is capable of forgetting little used information. The proposed algorithm contains a number of characteristics found in those used today, such as bone marrow algorithms are used to create artificial B-cells. In this algorithm, B-cells are stimulated or suppressed using Farmer's immune network modelling equation. Given N types of antibody with concentrations $\{c_1 \ldots c_n\}$ and M types of antigen with concentrations $\{y_1 \ldots y_n\}$ the rate of change in antibody concentration with time, Farmer's immune network model is given by Equation 3.8, where the first term represents the stimulation of antibody i by antibody j, the second is the suppression of antibody i by antibody j. The third term models the stimulatory effects of antigen and the fourth term is a constant cell death rate. K_1, K_2 and K_3 are all constant terms:

$$\frac{dc_i}{dt} = k_1 \left[\sum_{j=1}^{N} m_{j,i} c_i c_j - k_2 \sum_{j=1}^{N} m_{i,j} c_i c_j + \sum_{j=1}^{M} m_{j,i} c_i y_j \right] - k_3 c_i \qquad \textbf{3.8}$$

This AIS was evaluated first on a simple pattern recognition problem and then used for the classification of DNA sequences with results similar to or better than approaches such as nearest neighbour and neural networks. In order to examine whether this type of technique could be used for more traditional data mining problems, (Hunt et al., 1998) and (Neal et al., 1998) applied an immune network inspired algorithm, called JISYS, to the task of fraud detection. This work was a direct extension to that in (Hunt & Cooke, 1996). The results in (Neal et al., 1998) showed JISYS could identify all fraudulent patterns in a set of loan and mortgage application data.

In (Timmis et al., 2000) the issue of the use of immune networks for unsupervised learning is addressed. A population of B-cells was used, the stimulation of each was determined based both on the matching with other cells in the network and on an antigenic stimulus. The algorithm was tested on the public domain "Fisher Iris" dataset (Blake & Merz, 1998). Although the classification results were generally good, it was found that the size of the population grew uncontrollably and only identical cells may become connected in the network. This work was furthered in (Timmis & Neal, 2001), in which these issues were addressed and the resultant algorithm named AINE. Rather than using B-cells, this algorithm uses a population of what are termed "Artificial Recognition Balls" (ARBs), the stimulation of which determines the survival of the B-cell. Significant improvements in this algorithm include B-cell removal becoming dependent on cell performance, the use of ARBs to represent multiple B-cells thus increasing data reduction and the use of a tool called aiVIS to visualise the resultant network (Timmis, 2001). In (Knight & Timmis, 2001), AINE was further tested on benchmark datasets. AINE also inspired a multilayered AIS for clustering (Knight & Timmis, 2002).

Contrasting with this work is that of de Castro and Von Zuben, who authored a system called aiNet (de Castro & Von Zuben, 2002), an unsupervised data clustering tool based on immune network principles. aiNet is "*a disconnected weighted graph composed of nodes called antibodies and sets of node pairs called edges with an assigned number called weight or connection strength*" (de Castro & Von Zuben, 2002). aiNet relies on a set of cells interconnected by links with associated connection strengths. The higher this strength then the closer and more similar the antibodies are. In common with many other algorithms of this type, aiNet uses clonal selection while hierarchical clustering techniques are used to define network structure. A complexity analysis of aiNet was performed and aiNet was shown to work in time of order $O(m^2)$, where m is the final number of memory cells. Performance was tested on three

benchmark problems and, while the results appeared good, no comparison with another algorithm was performed. It was noted also that aiNet contains many user definable parameters. For this reason a sensitivity analysis of these parameters was performed and it was concluded that aiNet should be augmented to account for adaptive parameter adjustment. aiNet has recently been shown to mimic observed behaviours of the natural immune system (de Castro, 2003).

Further examples of immune networks and unsupervised learning include the formation of stable clusters in immune network clustering algorithms (Wierzchon & Kuzelewska, 2002) and an investigation of adaptive radius techniques for clustering (Bezerra et al., 2005). Related to classification is the field of information filtering. (Chao & Forrest, 2002) introduces the notion of an information immune system; a negative selection based immune algorithm for using collaborative group judgements to filter information.

3.5.2.2 Clonal Selection Solutions

Clonal selection algorithms have shown themselves to be competitive engineering solutions to a wide variety of problems such as optimisation (Cruz-Cortés et al., 2005) and the travelling salesman problem (TSP) (de Castro & Von Zuben, 2000). Due to their evolutionary nature, and therefore the similarities with evolutionary algorithms, many clonal selection algorithms have been used for optimisation (Kelsy et al., 2003; Timmis et al., 2004; Clark et al., 2005). Clonal selection has also proved fruitful in the field of data mining, especially classification. A noteworthy clonal selection based classification algorithm is Artificial Immune Recognition System (AIRS) (Watkins, 2001; Marwah & Watkins, 2002; Watkins & Boggess, 2002a; Watkins & Boggess, 2002b; Watkins & Timmis, 2002; Watkins & Timmis, 2004), as revised in (Watkins et al., 2004). As this algorithm is the closest example of work similar to the classification task encountered in part of this thesis, it will be described in its own section (section 3.5.3). Before that, some other works pertinent to this thesis are reviewed.

A clonal selection algorithm for data mining to draw inspiration from AINE (discussed in previously) is that of (Knight & Timmis, 2002; Knight & Timmis, 2003). This algorithm is a clonal selection based algorithm for clustering. It was noted by the authors that AINE has problems regarding the stability of the network produced. This algorithm, called MARIA, is based on clonal selection and designed to overcome the stability problems of AINE. While the immune system is inherently multi layered, MARIA has three layers of cells as follows:

1. **A free antibody layer.** This layer takes a single pattern as input based on the immune analogy that once entering the immune system an antigenic pattern is only presented to part of it. Any free antibodies with high enough affinity with this pattern are placed in the B-cell layer.
2. **A B-cell layer.** This performs more specific search of the antigenic pattern and its role is to learn and generalise the antigenic pattern. The cells in the B-cell layer undergo the usual processes of affinity maturation if the stimulus is recognised, otherwise the antigenic pattern is added in a manner the authors describe as being akin to a primary immune response.
3. **A memory cell layer.** This layer contains generalised patterns representative of both the free antibody and B-cell layers.

Cells present in all layers will age over time and once they have not received stimulation for a given number of iterations they are removed. This population control mechanism discourages an increasing population and also increases quality as only useful cells tend to survive. Some testing was performed and the algorithm was shown to achieve good data compression ratios without the loss of information as seen in the AINE system.

The paper (Alves et al., 2004) uses a different method to those cited above to produce a set of fuzzy classification rules. Called IFRAIS, the algorithm uses clonal selection to optimise a population of candidate fuzzy classification rules. Each candidate rule is conceptually an antibody and training examples to be classified are antigens. The overall affinity of the population with training data will increase as the population evolves the fuzzy classification rules over many generations to maximise affinity with the antigenic patterns (training examples). The results were shown to compare favourably with C4.5, a well known classification rule discovery algorithm, when tested on a number of public data sets.

3.5.3 AIRS: An Artificial Immune System for Classification

AIRS is an immune inspired classifier for multi-class problems. The inspiration for AIRS came from unsupervised data mining algorithms such as AINE and aiNet. The influence of AINE in AIRS can be seen most strikingly in the use of ARBs. Recall ARBs represent a number of identical antibodies, thus reducing duplication in the population. Rather than antibodies, ARBs undergo the process of clonal expansion and mutation. Also drawing on prior work of (Timmis & Neal, 2001), AIRS employs a

resource limitation mechanism. Much like the network model, antibodies in ARBs will compete for survival based on their stimulation level. This stimulation level is decided not only based on the affinity with the training antigen but also on a comparison with the antigen's classification. The higher the simulation level of an ARB, the more likely it is to be given a share of the global resources. ARBs that are not successful in this competition stage are removed. It is noted in (Watkins et al., 2004) that this competition for resources applies strong selective pressure on the population, as any ARBs surviving this stage will go on to produce offspring. The most suitable ARBs will be turned into long-lived memory cells and at the end of the training stage only these memory cells are left, thus increasing generality and reducing data redundancy. Memory cell identification, retention and replacement are based primarily on the classification ability of a memory cell. A stopping criterion during training is employed based on stimulation level and class to ensure only high quality memory cells are left.

After the algorithm has been trained and the final population of memory cells has been returned, the classification of unseen data is performed using a *k*-nearest neighbour algorithm (see Chapter 2) using the remaining memory cells. The pseudocode for AIRS is reproduced in Pseudocode 3.4 for comparison with the algorithms proposed in this thesis, to be described in subsequent chapters.

1. **Initialise**: Randomly create memory pool (M) and ARB pool (P)
2. **Antigenic presentation**: For each antigenic pattern (Ag)
   - 2.1. **Clonal Expansion**: Determine affinity between Ag and all elements of M from the same class. Select highest affinity element (mc), clone with rate proportional to affinity and place clones in P
   - 2.2. **Affinity Maturation**: Mutate all new elements of P.
   - 2.3. **Metadynamics**: Assess addition or reduction of resources of each element of P based on affinity with presented antigen. Remove elements where necessary. Calculate average stimulation and check for termination condition
   - 2.4. **Clonal expansion and affinity maturation**: Clone and mutate a subset of P probabilistically selected based on affinity
   - 2.5. **Cycle**: While average stimulation value from 2.3 is less than a given value, repeat from 2.3
   - 2.6. **Metadynamics of memory cells**: Select highest affinity ARB of same class as the antigen (mc-candidate). If affinity of mc-candidate is greater than that of mc then add mc-candidate to M.
3. **Cycle**: repeat until all antigenic patterns presented.

Pseudocode 3.4. AIRS: an artificial immune system for classification

AIRS was compared in (Watkins, 2001) against four standard datasets: iris, ionosphere, diabetes and sonar datasets from the UCI repository (Blake & Merz, 1998). It was noted that AIRS was very competitive with other algorithms, achieving a higher

classification accuracy than both C4.5 and a Bayesian approach over the diabetes dataset. A sensitivity analysis was also performed and the authors noted it satisfying that the algorithm was not particularly susceptible to deviations in the k value of the final k-nearest neighbour algorithm. AIRS was updated to AIRS2 in (Watkins et al., 2004) with five small changes being made, including slight updates to the mutation and cloning routines. AIRS2 was updated further to work as an efficient parallel system in (Watkins & Timmis, 2004; Watkins, 2005). AIRS has also been used for document classification, the subject of this thesis, as described in section 3.5.6.

3.5.4 Dynamic Clonal Selection

Until now only static learning has been considered, but this thesis addresses the issue of continuous learning. Continuous learning does not involve a training and a testing stage. Rather, the algorithm is initialised and both training and testing happen together.

The Dynamic Clonal Selection Algorithm (DynamiCS) (Kim & Bentley, 2002c; Kim & Bentley, 2002a; Kim & Bentley, 2002b) addresses just such an issue and is of interest as continuous learning is one of the main issues addressed in this thesis. The dynamic clonal selection algorithm was designed for use in a computer security scenario, where the threat to computers on a network will continuously change. In (Kim & Bentley, 2002c) a process of negative selection is still used to censor immature detectors. In this way the system learns normal behaviour by observing only a small set of self-antigens at any one time. Detector cells will be replaced whenever normal behaviours change. However, the system was found to be slow to react to changes, and a sharp change in self behaviour resulted in a high false positive rate. This outcome was due to memory detectors not being exposed to the entire set of self-patterns during tollerisation, a situation also present in the natural system. (Kim & Bentley, 2002a) introduced extended DynamiCS which had the added mechanism of removing memory detectors when they showed a poor degree of self-tolerance. This was shown to reduce the high false positive rate, but was at the expense of requiring a larger amount of co-stimulation (user intervention) to achieve this.

This work was further augmented in (Kim & Bentley, 2002b) by the addition of a hypermutation operator to produce the effect of gene library evolution. Rather than new detectors being generated randomly, new detectors were produced by mutating deleted detectors. Thus, a "virtual gene library" made from mutations of deleted memory detectors was maintained. The test results showed this scheme produced immature detectors that were better suited to cover existing non-self antigens.

One other work in the literature that addresses clonal selection for continuous learning is PICS (Pittsburgh Immune Classifier System) which is evaluated on both static and dynamic Time Dependent Learning (TDL) (Gaspar & Hirsbrunner, 2002; Gaspar & Hirsbrunner, 2004).

3.5.5 Other AIS-based Continuous Learning Systems

While not clonal selection based or used for classification, there exist some other AIS systems that exploit the immune system's ability to continually learn and adapt and as such are worthy of note.

In (Hart & Ross, 2002) the authors argue that there is a high computational overhead to clustering large amounts of data and as such, if the data is constantly changing, re-applying a clustering algorithm numerous times can be wasteful. Thus a new model for clustering non-static datasets is presented based on a combination of a type of associative memory and immune metaphors. Called SOSDM the system was shown to cluster non-stationary data well. This was updated in (Hart & Ross, 2003) to combine the SOSDM algorithm with ideas gleaned from Danger Theory (Matzinger, 2002a). Called dSOSDM, this updated algorithm was shown to cluster dynamically changing data better than the original SOSDM algorithm.

Immune networks are again used in (Neal, 2002) involving the continual analysis of changing data. The algorithm, SSAIS (Self-Stabilising Artificial Immune System), furthered the previous work of Timmis (Timmis & Neal, 2001) which was itself a step towards a continuously learning/self stabilising data analysis tool. The influence of this previous work can be seen in the use of ARBs, although this time no global resource allocation mechanism was used. Standard datasets were used for testing and a network was created that appeared stable over numerous updates. However, using these standard datasets to test a continuous learning algorithm by just presenting the same data multiple times seems a little artificial. Other immune network solutions to clustering moving or evolving data include the work of Nasraoui (Nasraoui et al., 2003), but classification and the use of clonal selection is notable by its absence in the literature when dealing with dynamic data.

While a rather different subject area to data mining, continuous adaptation has been exploited in the area of robotic control. A robot may be present in a constantly changing environment resulting in a need for it to continually update its behaviour in response. The paper (Hart et al., 2003) gives some motivation as to why a robot should naturally "grow up" and argues that the immune network model is a plausible approach. In

robotics, the common procedure of global path planning is often done offline and the robot knows the shape, location and movement of all obstacles in the environment. Local reactive navigation is an online and reactive strategy where the robot navigates without any prior knowledge and can therefore be the more useful of the two. This situation is conceptually close to the continuous learning scenario described in this thesis as the robot typically receives minimal training and then reaches its goal by relying on a certain amount of feedback from the environment (analogous to the feedback from the user as implemented in the algorithms described in the following chapters). The paper (Luh & Liu, 2004) describes one such strategy in which a continuous immune model is exploited to allow a robot to gradually learn its environment and navigate through an environment to reach a goal. It argues that the strategy as implemented is not highly optimised for particular environments; instead it is a robust technique suitable for more general environmental changes. The current state of the environment is presented as an antigen which will bind to a B-cell. These B-cells encode behaviours, such as steering angle, while the concentrations of the B-cells are manipulated by a differential equation.

One further, and particularly interesting use of AIS applied to a dynamic environment is demonstrated in (Singh & Thayer, 2001). In this case a group of robots are required to work together to complete a task; in this case, mine clearing. This is not the situation where the environment can be learnt offline and so true continuous learning was demonstrated. The group of robots exhibit self organisation without any global control mechanism and because of this the group of robots was seen to be tolerant of failures to robots within that group.

Inspiration for the strategy implemented by Luh and Lui can be seen to have come from (Wantanabe et al., 1999) in which continuous immune networks were used for robotic control. This time, rather than navigation, the robot has a specific task to do in that it must retrieve rubbish in its environment and deposit the rubbish in a bin whilst ensuring its batteries remain charged by using a charging point when necessary. Each immune cell encodes an action and there is competition between all cells in the network to decide which action takes place next. The ranking of a cell is decided by its concentration in the population with that being determined by a differential equation. Previous works by the author of interest are (Ishiguro et al., 1997b) and (Ishiguro et al., 1997a).

Ayara has shown AIS to be a reasonable technique for predicting faults in automatic teller machines (Ayara et al., 2005; Ayara, 2006). In this situation, a modified AISEC

algorithm (Secker et al., 2003a) continually learns the situations that may cause a machine to fail and allows an engineer to be alerted before the machine fails, whilst adapting to changing conditions to ensure a low false alarm rate.

It can be seen that there is a small but growing amount of work in the literature regarding the application of AIS to continuous learning situations. However, it is interesting to note that this evidence is empirical only. While abstracting from the biology many authors have stated it reasonable to assume that AIS should be predisposed to learning continuously no formal proof can be found for this in the literature. This empirical evidence is enough to motivate these investigations and it is thought that there is also enough to justify the use of AIS as a reasonable choice.

3.5.6 Web and Text Mining with Artificial Immune Systems

A review of the literature shows few projects in the field of AIS that have been used for either text mining or web mining. The first work to mention the use of AIS for a text mining task was (Aickelin & Cayzer, 2002). This conceptual paper briefly mentions the use of an AIS, stimulated by user behaviour, which will match certain document features and flag the document as interesting to the user. This again is the first known use of the term "interestingness" as applied to mining documents with AIS. However, the example given in the paper did not deal with interestingness as defined in this thesis; it appeared to refer to document relevance. The description of such an algorithm is at a high level of abstraction, no algorithmic details are attempted, but none the less, that high level description has provided some motivation for this thesis.

A second publication, co-authored again by Cayzer, was the first work to use an AIS for document classification (Twycross & Cayzer, 2002); and later (Twycross, 2002b; Twycross, 2002a; Twycross & Cayzer, 2003) describe a clonal-selection based method for document classification. It is the system's task to classify a document into two classes: those which were on a given topic or not on that topic. Each individual cell encodes the presence or absence of a specific feature (in this case, a word not on a given list of stopwords) using a binary representation. A "don't care" attribute is also used. A feature extraction algorithm was used to convert a raw HTML document into a Boolean feature vector. The cell also includes a matching threshold, again encoded in binary. This was a number in the range 0-1 which had the effect of weighting the cell in the population. Cells that would predict a class with high confidence were rewarded with lower thresholds. Thus, fewer of their bits had to match a presented antigenic pattern for a positive classification to take place.

The algorithm was tested by classifying pages taken from the Syskill and Webert Web Page Ratings from the UCI data repository, a public collection of standard datasets that data miners may use to test their algorithms (Blake & Merz, 1998). This dataset consists of HTML pages, each on one of four different topics. The task was for this immune inspired system to predict if an unseen page was on a given topic or not when the system had been trained using a number of example pages. Hence, in order to predict the four classes (topics) the algorithm had to be run four times, each run performing a binary classification indicating whether or not a document belonged to a given topic. The predictive accuracy compared favourably against a naïve Bayesian algorithm, and achieved a higher predictive accuracy in three out of the four topics. The results showed that the system was relatively insensitive to the size of the training set. This was in contrast to the Bayesian system, although the AIS did favour large amounts of training data. It should also be noted that this ran on static data sets unlike the dynamic data considered in this thesis. Indeed, in (Twycross, 2002a), the use of this algorithm in a dynamic domain was acknowledged as a possible further direction.

In a typical case of algorithm modification and reuse, (Greensmith & Cayzer, 2003) use the immune classifier AIRS (section 3.5.2.2) but extend it for the classification of documents into a taxonomy that may be personalised for a particular user. The authors acknowledge that a suitable encoding scheme must be identified, for which they suggest repeating the Boolean approach taken by Twycross or using TFIDF. This change in representation may also require a change in affinity evaluation function such as a cosine measure. Although the authors acknowledge that the AIRS default of Euclidean distance is suitable for a baseline.

In (Greensmith, 2003) some preliminary results are presented for both two-class and multi-class document classification. The results showed no statistically significant variation in accuracy between a two-class and a four-class problem. A measure of information gain was used to extract the n (where n was varied between 100 and 200) most informative features (words) and the feature string of each cell consisted of a binary representation of the presence or absence of these words.

One AIS more concerned with general document filtering rather than specifically e-mail filtering is described in (Nanas et al., 2004). It uses the immune network theory (Jerne, 1974) to represent a user's interests. The main offering of this paper is to acknowledge the need for an adaptive system that will track changes in user interests. The system may represent multiple, interacting topics and has the advantage of being swift to react to fast changes in user preference or *concept shift*. The network paradigm

seems to show a fluid system as links between concepts continually grow weaker or stronger based on user interaction, but no experimental results are provided.

Other than classification there has been some work in the field regarding the use of AIS for document clustering. (Hang & Dai, 2004) is one example and, in a similar manner to Greensmith's approach, (Tang & Vemuri, 2005) describe their attempt to use aiNet for document clustering. A binary representation was used with approximately 15% of features being chosen based on word frequency variance. However no details about the affinity evaluation metric was given. Empirical analysis was undertaken using 20 Usenet newsgroups datasets and it was shown that aiNet produces compact clusters especially over large or noisy data.

The works of Twycross and Greensmith are both web content mining systems to some extent. Both attempt to classify web data but the emphasis was rather on the classification of documents, where the documents happen to come from the web, rather than being specifically aimed at web content mining. The same is true of the clustering system of Tang and Vemuri, although Usenet data does not strictly come from the web (Tang & Vemuri, 2005), all can be seem to have some commonality with the website recommendation system found in (Morrison & Aickelin, 2002), the inspiration for which can be seen in (Cayzer & Aickelin, 2002b; Cayzer & Aickelin, 2002a). Although immune network based, this algorithm performs a binary classification. Rather than a strict class being assigned to an unknown class example, a recommender system makes a suggestion much like that of an "information immune system" (Chao & Forrest, 2002). This system is unique amongst the current literature in that it is a collaborative system whose dynamics are influenced by many users. Nootropia (Nanas et al., 2004) performs a similar function, again using an immune network.

Other than the web content mining and document mining papers above, one single paper in the literature tackles web usage mining. (Nasraoui et al., 2002) uses an algorithm based on AINE to detect user access patterns for websites. Since usage mining is not the subject of this thesis, the reader is directed to the literature for further details about that work.

3.5.7 AIS for E-Mail Classification

Recently a very small number of papers have been published regarding the use of an Artificial Immune System (AIS) for the classification of e-mail, as considered in the next chapter of this thesis. It is believed that, at the time of writing, this review covers all papers concerning the classification of e-mail with AIS.

To begin with the system of Oda and White, in their first paper (Oda & White, 2003a) the authors describe an AIS based spam detection system based on matching regular expressions with e-mail. A set of 300 regular expressions were used as a gene library. These genes were randomly concatenated together to form longer regular expressions. It should be noted here that each regular expression in the gene library was derived from the common free spam filter "SpamAssassin"(SpamAssassin, 2006). Each of these new regular expressions become an antibody in the AIS and around 1000 of these antibodies were generated. The system was then exposed to a training set of around 1600 spam e-mails and 1000 non-spam e-mails. The AIS adjusted a weighting of each individual antibody depending on that antibody's classification performance. A selection process occurred over all antibodies, leaving only the highly weighted – and therefore the most useful antibodies. Approximately 100 antibody cells were left after this stage.

A mechanism was implemented such that unused antibodies would expire if unused for classification for a certain amount of time. A new cell would be generated to take its place. During the testing phase the remaining antibodies were exposed to 1200 e-mails of undetermined class. To perform a classification, an e-mail was presented to every cell in the system and a record is made of the weight of each cell containing a regular expression matching the e-mail. If the sum of these weights is above a threshold, the e-mail is considered spam.

The results presented in this paper were that 1% of the messages were false positives and the overall classification accuracy was approximately 90%. It was not reported that this figure was achieved over multiple runs of the system and no confusion matrix or standard deviation figures were given.

In Oda and White's second paper (Oda & White, 2003b), the authors make some changes to the weighting system as "*there was the potential for highly-weighted vectors to overwhelm the effect of any others*" in the original implementation. During the training stage two different weighting strategies were tried: straight sum and weighted average. Training this time took place on 3000 messages with a 50%/50% class split and 127 detectors were produced. These were candidate detectors as they had not matched any legitimate messages during the training phase. This time 501 non-spam and 401 spam messages were used for testing (and therefore it is possible a different test set has been used and as such the results of the two papers are not comparable). The straight sum weighting scheme recorded an overall predictive accuracy of 86% over the

test set, which was slightly less than the 90% reported for the weighted average. This latter scheme also performed better with regard to reducing false positive classifications.

Finally, in 2005 a third paper was published (Oda & White, 2005) in which the algorithm was again changed slightly and this time the SpamAssassin heuristics were augmented with an English dictionary and a set of tokens drawn from the training set of e-mails. Only 201 heuristics were used, it is unclear why fewer were used in this investigation. In addition to the weighted average and straight sum methods of scoring, this investigation also used a Bayes score in an attempt to increase accuracy. Tests were performed on the SpamAssassin public corpus of e-mail. In the results, however, the most accurate scoring method turned out to be the weighted average as described in the previous paper, which scored 93.6% accuracy with 1.1% false positives.

Although the first published attempt at an AIS for spam filtering there are certain problems with the Oda and White technique. The main criticism would be that, in all fairness, the immune system as implemented is only acting as an optimiser for the SpamAssassin rule set. It can be argued that for an AIS to stand out in the realm of spam filtering, these classification rules, or any other classification model, should be generated by the AIS itself. SpamAssassin, as a rule based spam classifier, has many hundreds of rules, each honed over time to increase accuracy and reduce false positives. Each rule in spam assassin is weighted already, and the SpamAssassin program even contains a module to allow a genetic algorithm to update individual rule weights based on a user's previously seen e-mail. By using a subset of these rules, the algorithm of Oda and White is already one step ahead of other classifiers as the rules used have already been constructed and tested for quality. During the system lifecycle, Oda and White's system concatenates rules, thus the antibodies do not use exactly the same rules as SpamAssassin, but it is quite possible that the system will construct antibodies where one part is a regular expression that is generally good at detecting general spam, whilst the other part is a rather specific regular expression that may match little. This may tend to result in an expression, for all intents and purposes classifying e-mail using the most generic part of the rule, thus acting in a similar manner to SpamAssassin. To take this one stage further, no examples of the top n generated regular expressions were given. It is therefore possible that one single but general regular expression dominates a significant proportion of the resulting cells.

In the third paper (Oda & White, 2005) changes made to the algorithm appear to have increased the accuracy and reduced the false positive rate. However, the dataset used for testing is the SpamAssassin dataset. While this is a publicly accessible and

widely used dataset, SpamAssassin itself has been biased to excel when classifying this dataset. Therefore, if the Oda and White's system has used SpamAssassin heuristics, is it any wonder that it exhibits an increase in accuracy? The low false positive rate is likely to also be a symptom of this, especially as the Oda and White algorithm has no specific mechanism to reduce false positive classifications. In addition to this, a total of 700 cells were required to produce this behaviour, far in excess of the 127 used in the previous paper which resulted in similar classification accuracies.

To summarise, Oda has used SpamAssassin regular expressions and written an AIS to optimise the weights of these expressions and picked the top n weighted expressions as the antibody set. Although no direct comparison with SpamAssassin was undertaken the reported predictive accuracy figures for the AIS are comparable to but possibly lower than the typical performance of SpamAssassin itself. Indeed, the AIS results in a slightly higher rate of false positives than the expected performance of SpamAssassin. Therefore, it can be argued that Oda and White have decreased the number of rules used for classification at the expense of accuracy.

3.6 Summary

In this chapter natural and artificial immune systems have been reviewed in some depth. After presenting some basic immunology, a generic framework for engineering AIS was used to detail the multilayered approach of representation, affinity and process. The most relevant works in the literature were then reviewed with focus on classification. The few papers currently available regarding continuous learning with a clonal selection algorithm, and the use of artificial immune systems for text mining, web content mining and e-mail classification were reviewed in some depth.

In this chapter and Chapter 2, the foundation has been laid for the remainder of the thesis. In the remainder of the thesis, the immune principles from this chapter are implemented as a computer algorithm aimed at solving two web mining problems.

Chapter 4 AISEC – an Artificial Immune System for E-mail Classification

4.1 Introduction

This chapter proposes and tests an Artificial Immune System (AIS) for filtering e-mail. A more succinct description of the system covered in this chapter, with fewer experimental results can be found in (Secker et al., 2003a). It is the aim of this thesis to investigate ways AIS algorithms react in text-mining scenarios with regard to the retrieval of interesting information. Two individual case studies are described in this and the following chapter. These are the tasks of dynamically filtering e-mail and dynamically identifying interesting web pages.

The task of e-mail filtering was chosen as a scenario particularly suitable for an AIS as the domain is assumed to be dynamic, however it is also a simpler domain when compared to the web and as such an ideal area to test and refine new ideas. The task detailed in this chapter is to distinguish between two classes of e-mail: that which the user would not be interested in and that which, to the user, is important or interesting. The choice is made automatically by the AIS based on previous experience. The AIS algorithm described here is designed specifically for use in a continuous learning scenario. The concept of what the user finds interesting will change over time, so too will the content of uninteresting e-mails. This e-mail classification task is considered a kind of web mining task as covered by the definition of web mining by (Kosala & Blockeel, 2000) as the text contained in e-mail is used for the purposes of classification and e-mail is one example of a web service .

A further reason for this choice of testing scenario was that the problem of receiving uninteresting e-mail is one faced by most who use e-mail on a day-to-day basis. It is a well understood problem with a number of references in the literature which propose solutions, some of which were described in Chapter 3. Although, it should be noted that the following algorithm is not proposed as a spam filter. Such spam filters are highly

specialized pieces of software specially written for the task of removing mass-mail from a user's inbox. The rules of spam filtering are rather different to the generic text-mining task this algorithm is designed to investigate (Graham, 2003). For example, an important or otherwise interesting e-mail incorrectly classified and removed can be disastrous. Spam filters are therefore written specially to minimize the risk of such an occurrence. The penalty for misclassifying an interesting document when the final web mining system is run is not nearly as severe as misclassifying an e-mail. The task of the system described in this chapter is to sort incoming e-mail into two classes; that which is interesting to a user and that which is not interesting. This latter class may well include mass mailed e-mail (spam) but may well also include other types of e-mail specifically not interesting to that user. Spam also tends to be general and non-personal in subject, but personal e-mail may be uninteresting too.

The Artificial Immune System for E-mail Classification (AISEC) described in this chapter possesses a number of features, the combination of which distances it from spam filtering systems. The main difference is that this addresses a continuous learning scenario. This contrasts with the majority of classification systems taken from the literature and reviewed previously which are trained once and then left to run. In addition to this, concept drift is addressed, a feature implicit in the continuous learning scenario and one feature few other e-mail classifiers possess. One further advantage of an AIS is that such a classifier may require no specific feature selection mechanisms. In contrast to some systems words are not pre-selected from the training data, instead a selection is performed in a data driven manner implicitly by the evolutionary operators.

To summarise, the system proposed in this chapter is more general than a spam filter because it can cope with any definition of interesting and uninteresting e-mail.

4.2 Motivations

4.2.1 Immune Inspiration

But why use AIS to tackle this problem? Specific to the task of e-mail classification, an AIS naturally lends itself to coverage of continuously changing and/or noisy data. Noisy data in this context could include misspelled words or sections of text that may distract from the main topic. The use of AIS for domains in which the data tends to be noisy has been highlighted for some years now. Work in (de Castro & Timmis, 2002a) describes a number of desirable properties afforded by AIS Algorithms, one of which is noise tolerance. Due to the combination of an affinity function and a population of cells

covering potentially overlapping points in the solution space, an AIS has the potential to filter noisy data and uncover underlying concepts.

In addition to this, the immune system lives in an ever-changing environment. The immune system will constantly have to tackle antigenic signatures it has never seen before while those it has seen may change and adapt over time. With constant regeneration of immune cells covering new areas of shape space, combined with the clonal selection mechanism when the immune system is stimulated results in a system constantly changing and adapting. The dynamic nature of the immune system can be capitalised on in the domain of e-mail classification as the attribute values of the application domain are under constant change. The topics a user may be interested in are liable to drift over time, so too may the actual content of the e-mails on the interesting subject. The ability of an algorithm to keep track of changes in the application domain is very important in such a filter and one given "for free" by the correct implementation of ideas inspired by the natural immune system. Indeed, in the paper (Hart & Timmis, 2005), the authors cite "life-long learning" as one of the application domains which may set AIS apart from other machine learning algorithms.

4.3 An Overview of the Artificial Immune System for E-mail Classification (AISEC) Algorithm

AISEC seeks to classify unknown e-mail into one of two classes – interesting and uninteresting – based on previous experience. It does this by manipulating the populations of two sets of artificial immune cells. Each immune cell captures several features and behaviours from both B-cells and T-cells, but for simplicity these are known as B-cells throughout. These two sets consist of a set of naïve (immature) B-cells and a set of memory B-cells. The use of both mature and immature cells was inspired somewhat by (Hofmeyr & Forrest, 1999; Hofmeyr & Forrest, 2000).

Once the algorithm has been initialised, each B-cell represents an example of an uninteresting e-mail that contains words from that e-mail's subject and sender fields in its feature vector. That is, the vector of attribute values defining the position of the B-cell in solution space. Uninteresting rather than interesting e-mail is represented in order to reduce the number of undesirable misclassifications. That is, if mail is misclassified then it is likely to be uninteresting e-mail that has not been recognised as such and thus has been misclassified as interesting. This contrasts with the opposing strategy which would misclassify interesting e-mail as uninteresting which is a very undesirable situation.

New e-mails to be classified are considered to be antigens. Hence in order to classify an e-mail, it is first processed into the same format of feature vector as a B-cell and then presented to all B-cells in the population. If the affinity between the antigen and any B-cell is higher than a threshold, the B-cell is said to recognise the antigen and thus the latter is classified as uninteresting. Otherwise the antigen is assumed to be interesting and allowed to pass to the user's standard inbox. Assuming the antigen (e-mail) is classified as uninteresting it will be removed to a temporary store. The tracking of concept drift is most effective when the user has some input (Klinkenberg, 1999) and so if this antigen is later confirmed by a user to represent an uninteresting e-mail the B-cell which classified it as such is useful, and is rewarded by promotion to a long-lived memory B-cell (assuming it was not already). This confirmation is given by the user in a non-invasive way by the user deleting the e-mail from the store, an action the user will typically perform anyway. Thus AISEC fulfils the criteria that relevance feedback techniques should be avoided (Chen et al., 2001) as "*web personalisation products must be non-invasive for the user*" (Chan, 1999).

Upon receipt of a confirmation signal, the cell is selected to reproduce by clonal selection. This constant reproduction combined with appropriate cell death mechanisms are features that afford this algorithm its dynamic nature. User feedback is given asynchronously to classification but on a regular basis. As the algorithm is designed to address concept drift over long periods, reasonable pauses in this feedback should not cause an undue drop in classification accuracy.

Once an e-mail has been placed in the user's inbox, either by classification or by the user, it is no longer accessible to the algorithm. The algorithm was designed to be run as a proxy with the user client retrieving its e-mail from AISEC, which in turn connects to the remote mail server. When the user removes mail to save space it is assumed he/she will do so by removing mail from the mail client's inbox, thus having no effect on the algorithm. As the "uninteresting mail" folder kept by the algorithm is nothing more than a temporary store and so should be emptied regularly. A high level outline of this process is shown in Figure 4.1

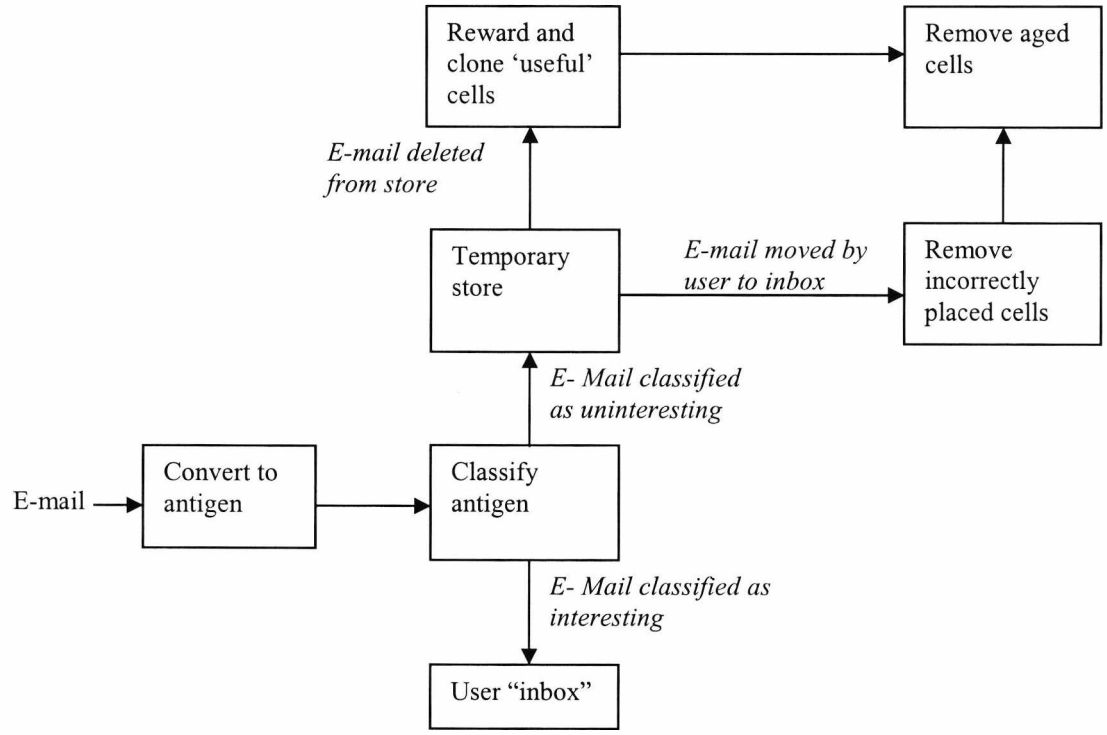

Figure 4.1. High level view of the AISEC system after initialisation

During design a number of special considerations were given to the specialist nature of the text mining problem. Some of these considerations took into account the basic principle of tailoring an AIS for a target problem in order to increase the algorithm's effectiveness (Freitas & Timmis, 2003). These design decisions are discussed below:

Gene libraries: Two libraries of words, one for subject words and one for sender words are used. These contain words known to have previously been used in uninteresting e-mail. When a mutation is performed, a word from this library replaces a word from a cell's feature vector. Mutating a word in any other way, by replacing characters for example, would result in a meaningless string in almost all cases. This strategy has recently been highlighted in (Cayzer et al., 2005).

Reproduction by cloning: A random generation of feature vectors as described in (Hofmeyr & Forrest, 1999) has been common but would be wholly inefficient in this application domain for the same reasons as above. Therefore all new cells entering the naïve cell set due to the cloning of a parent cell are mutants of the parent.

Co-stimulation: As mentioned previously, e-mail classified as uninteresting is not deleted but removed to a temporary store. Interesting e-mail is delivered to the user client in the normal way (and so no longer accessible by the algorithm). B-cells must have become stimulated to classify an e-mail as uninteresting, and therefore it is assumed the first stimulatory signal has already occurred. Feedback from a user is then interpreted to provide (or not provide) a co-stimulation signal (or signal two of the two-

signal approach to self/nonself discrimination). At a time of the user's convenience this store may be emptied. The actions of the user during this procedure drive a number of dynamic processes. If an e-mail is simply deleted from this store the algorithm has performed a correct classification as the user really was not interested in that e-mail, and so a co-stimulation signal has occurred. The cell is rewarded by being allowed to reproduce. If, on the other hand, the user does not delete the e-mail the algorithm has performed a misclassification, therefore signal two does not occur and B-cells are removed appropriately.

Two recognition regions: Around each B-cell is a recognition region within which the affinity between this cell and an antigen is above a threshold (Figure 4.2). It is within this region an antigen may stimulate the B-cell. A single region was found to be inefficient for both the activation of evolutionary processes (clonal selection) and for the recognition of a new e-mail. A smaller region, a recognition region, was introduced for classification only. During development, the introduction of this second region was shown to increase the classification accuracy significantly on the test set.

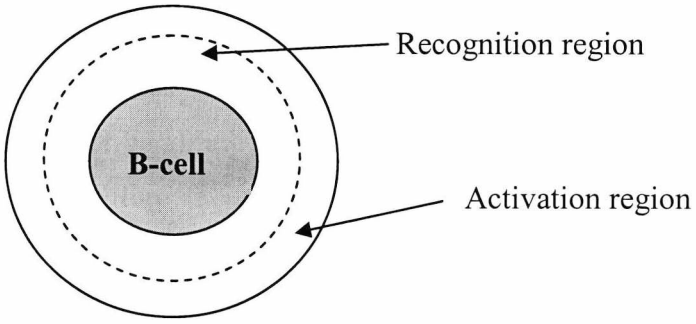

Figure 4.2. Recognition and classification regions surrounding a B-cell

Cell death processes: To both counteract the increase in population size brought about by reproduction and keep the algorithm dynamic, cell death processes are required. A naïve B-cell has not proved itself useful to the algorithm and, as such, is given a finite lifespan when created, although it may lengthen its life by continually recognizing new uninteresting e-mails. Memory B-cells may also die, but these cells have proved their worth and it can be hard for the algorithm to generate clones capable of performing well. For this reason, unlike naïve B-cells, memory cells are purged in a data driven manner. When a new memory cell is added to the memory cell set all other memory cells recognising this new cell have their stimulation level reduced. When this stimulation reaches zero they are purged from the population. This dissuades the algorithm from producing an overabundance of memory cells, each providing coverage

over a given area of the solution when a single cell is sufficient. This population control mechanism is conceptually similar to the use of Artificial Recognition Balls (ARBs) as found in many works by Timmis and derivatives (Timmis & Neal, 2000; Timmis & Neal, 2001; Watkins & Timmis, 2002) although the actual mechanism differ somewhat.

It can be seen that this strategy will allow the AIS to continually update and keep track of changing user preference. For example, if e-mail about a certain subject was frequently classed as uninteresting, but the user changed opinion and decided that topic was interesting after all, the action of moving the e-mail(s) on that subject to the inbox would ensure that topic is less likely to be classified as uninteresting in the future. Thus AISEC is very fast to react.

After consideration it was noted there is a situation in which AISEC would be slow to react to changing user preference. In this case, e-mail that is considered interesting suddenly becomes uninteresting to the user. In this case, the user may delete the e-mail from the inbox but the system has no way to react to this. The only way the system may adapt to this change is by random mutation of the B-cell set over time, however, it is acknowledged that it may be hard to get some specific words that are present in this newly uninteresting e-mail into the gene libraries in the first place as the system is not going to classify them as uninteresting and as such does not have access to the words they contain. In this instance it is acknowledged that AISEC will be slow to react. The feedback mechanism from the user's inbox was deliberately left out during design as it was thought it may be intrusive, and would limit the way in which AISEC could be implemented in a real environment (recall AISEC was designed to act as a proxy between an e-mail client and server). However, this is an acknowledged limitation and one that may be worthy of further work in the future.

4.4 The ASIEC System in Detail

In this section, AISEC algorithm is described. This section is divided into three subsections: representation, affinity measure and processes. This follows the same engineering process introduced in Chapter 3 when a generic explanation of AIS was given. The following therefore follows the AIS framework proposed by (de Castro & Timmis, 2002a).

The following notational conventions will apply throughout:

- Let `BC` refer to an initially empty set of naïve B-cells
- Let `MC` refer to an initially empty set of memory B-cells

There are seven major parameters in the AISEC algorithm:

- Let `Kt` refer to the initial number of memory cells generated during initialisation
- Let `Kl` refer to a constant which controls the rate of cloning
- Let `Km` refer to a constant which controls the rate of mutation
- Let `Kc` refer to the classification threshold
- Let `Ka` refer to the affinity threshold
- Let `Ksb` refer to the initial stimulation count for naïve B-cells
- Let `Ksm` refer to the initial stimulation count for memory B-cells

4.4.1 Representation

A B-cell receptor holds information extracted from a single e-mail, this is represented as a structured vector of two parts, Figure 4.3. One part of the vector holds words contained in the subject field of an e-mail, the second holds words contained in the sender (and return address) fields. The two sub-vectors are unordered, can contain duplicate words and are of variable length. The contents of both parts can be replaced by different words during the mutation process which occurs when the algorithm is in the running phase. The actual words are stored in the feature vector for computational efficiency and other programming reasons. This can be contrasted to the common practice of using a vector containing binary values as the receptor, each position in which represents the presence or absence of a word. As new words are continually being encountered by and removed from the population, each cell's vector would have to be updated as appropriate each time this action occurs. This could account for many updates per generation and, when multiplied by the size of the population, would create a significant computational overhead.

```
B-cell vector = <subject,sender>
```

Where

```
subject = <word 1,word 2,...,word n>
sender = <word 1,word 2,...,word m>
```

Figure 4.3. Structure of B-cell vector

Each B-cell will also contain a stimulation counter used for aging the cell. This is initialised to `Ksb` on cell generation and reset to `Ksm` if the B-cell is later promoted to a memory cell and added to `MC`. An antigen is represented in the same way as an

antibody. This allows for a straight forward affinity function to be defined to measure the similarity between two feature vectors.

4.4.2 Affinity Measure

The affinity between two cells is a measure of the proportion of one cell's feature vector also present in the other, in terms of number of words in the feature vectors. It is used throughout the algorithm and is guaranteed to return a value between 0 and 1. The matching between words in a feature vector is case insensitive but otherwise requires an exact character-wise match. The ordering of the words in the vector is irrelevant. Given `bc1` and `bc2` are the cells whose affinities are to be computed, the procedure is outlined in Pseudocode 4.1.

```
1  PROCEDURE affinity (bc1, bc2)
2    IF(bc1 has a shorter feature vector than bc2)
3      bshort ← bc1, blong ← bc2
4    ELSE
5      bshort ← bc2, blong ← bc1
6    count  ← the number of words in bshort present in blong
7    bs_len ← the length of bshort's feature vector
8  RETURN count/bs_len
```

Pseudocode 4.1. Affinity

In this pseudocode, the inputs are given the names `bc1` and `bc2` referring to B-cells however in the following pseudocodes the actual inputs may be B-cells (usually denoted `bc`), antigens (usually denoted `ag`), memory cells (usually denoted `mc`) or elements of the training set (usually denoted `te`). However, all conform to the pattern of a B-cell, that is, all have feature vectors conforming to the same template.

After some analysis it was noted that this function may have the unintended effect of driving down the average length of the feature vector. This occurs because it is easier for a shorter length vector to score highly compared to a longer one. For example, a vector of length 1 can score the maximum affinity value possible if it only matches 1 element of a longer vector. Based on this observation, am affinity function that does not exhibit this characteristic yet can still compare 2 features of different lengths would be a welcome update in the future.

4.4.3 Algorithms and Processes

AISEC is a population-based algorithm consisting of two distinct stages: a training phase followed by a running phase. This running phase is further divided into two tasks,

that of classifying new data and intercepting user feedback to drive evolution. During the initialisation phase, initial gene libraries are populated with words from the initialisation set. Actual classification of unknown examples is then performed in the running phase. User feedback is employed as a second signal confirming whether classification is correct or incorrect. The first signal is recognition of uninteresting e-mail or non-self antigen. Depending on the result of the second signal, immune cells are rewarded or suppressed. Immune cells are rewarded for correct uninteresting classification through a process of stimulation. The highest affinity cell is then selected for cloning and possible mutation. If the selected cell is a naïve B-cell, it is promoted to a memory cell. The introduction of a new memory cell suppresses other similar or nearby memory cells (as dictated by the affinity threshold, `Ka`). Cell death processes control cell population by removing non-stimulated cells or by removing cells which take part in misclassification. With regard to the mutation process, two separate gene libraries provide "word pools" for replacement words in the subject and sender sub-vectors. These contain words known to have previously been used in uninteresting e-mail.

A high level overview of the entire algorithm is shown in Pseudocode 4.2, where `TE` denotes the set of initialisation examples and `ag` denotes an antigen (processed e-mail of unknown class).

```
1   PROGRAM aisec
2     initialise(TE)
3     wait until (an e-mail arrives or a user action is intercepted)
4       ag ← convert e-mail into antigen
5       IF(ag requires classification)
6         classify(ag)
7         IF(ag classified as uninteresting)
8           move ag into user accessible storage
9         ELSE
10          allow e-mail to pass through
11      IF(user has given feedback on ag)
12        update_population(ag)
```

Pseudocode 4.2. AISEC overview

Each of these three stages is further described in turn: initialisation, classification and the updating of the population based on user feedback.

During the initialisation stage (Pseudocode 4.3) the goal is to populate the gene libraries, produce an initial set of memory cells from initialisation examples, and produce naïve B-cells based on mutated initialisation examples. As the B-cells in the AISEC algorithm represent only one class, the initialisation set (`TE`) contains only e-mails the user has explicitly selected as uninteresting. This is the most efficient way of

representing the learned knowledge in this system as in this dataset the uninteresting e-mail is the minority class. In other words, the system can focus on learning good descriptions (B-cells) of the minority class, and e-mails that do not satisfy those descriptions are assumed to have the majority class by default.

```
1   PROCEDURE initialise(TE)
2     FOREACH(te ∈ TE)
3       process e-mail into a B-cell
4       add subject words and sender words to appropriate library
5     insert Kt processed e-mails into MC, selected at random
6     FOREACH(mc ∈ MC)
7       mc's stimulation count ← Ksm
8     FOREACH(te ∈ TE)
9       te's stimulation count ← Ksb
10      FOREACH(mc ∈ MC)
11        IF(affinity(mc,te) > Ka)
12          clones ← clone_mutate(mc,te)
13          FOREACH(clo ∈ clones)
14            IF(affinity(clo,te) >= affinity(mc,te))
15              BC ← BC ∪ {clo}
```

Pseudocode 4.3. Initialisation

Throughout the rest of this chapter it is stressed that this stage is initialisation, not training in the machine learning sense. This process is used simply to generate an initial set of cells to allow the system to function. Its primary job is not to produce a set of cells that will classify with a high degree of accuracy (although this is, of course, highly desirable).

Once the system has been initialised, it is available to begin two distinct functions. These are the classification of unknown e-mails and the population update processes based on user feedback on the correctness of classification attempts. During this running phase the algorithm will wait for either a new e-mail to enter the system and so be classified or an action from the user indicating feedback. Upon receipt of either of these, the system will invoke the necessary procedure as outlined in either Pseudocode 4.4 or Pseudocode 4.5.

```
1   PROCEDURE classify(ag)
2     FOREACH(bc ∈ (BC ∪ MC))
3       IF(affinity(ag,bc) > Kc)
4         classify ag as "uninteresting"
5       ELSE
6         classify ag as "interesting"
```

Pseudocode 4.4. Classification

To classify an e-mail, an antigen, ag, is created in the same form as a B-cell, taking its feature vector elements from the words in the e-mail's subject and sender fields. ag is then assigned a class based on Pseudocode 4.4. The procedure is straight forward. If an antigen has an affinity with any naïve B-cell or any memory B-cell greater than a threshold it is assumed to be of the "uninteresting" class.

To purge the population of cells which may match interesting e-mails, the AISEC algorithm uses the two signal approach, the basis of which was made clear in Chapter 3. Recall, this strategy ensures T-cells only react to antigen when properly presented. Recognition of the antigen is signal one and stimulation by antigen presenting cell is signal two. Since signal one has occurred, that is, the antigen has already stimulated a B-cell and been classified, and therefore recognised. Signal two comes from the user in the form of interpreting the user's reaction to this e-mail. It is during this stage that useful cells are stimulated, and un-stimulated cells are removed from the system. ag is the antigen (e-mail) on which feedback has been given. Pseudocode 4.5 shows this process in detail.

```
1   PROCEDURE update_population(ag)
2     IF(classification was correct)
3       FOREACH(bc ∈ BC)
4         IF(affinity(ag,bc) > Ka)
5           increment bc's stimulation count
6       bc_best ← element of BC with highest affinity to ag
7       BC_new ← clone_mutate(bc_best,ag)
8       FOREACH(bc_new ∈ BC_new)
9         determine affinity(bc_new,ag)
10      bc_best ← element of BC with highest affinity to ag
11      mc_best ← element of MC with highest affinity to ag
12      IF(affinity(bc_Best,ag) > affinity(mc_best,ag))
13        BC ← BC \{bc_best}
14        bc_best's stimulation count ← Ksm
15        MC ← MC ∪ {bc_best}
16        FOREACH(mc ∈ MC)
17          IF(affinity(bc_best,mc) > Ka)
18            decrement mc stimulation count
19        add words from ag's feature vector to gene libraries
20    ELSE
21      FOREACH(bc ∈ (MC ∪ BC))
22        IF(affinity(bc,ag) > Ka)
23          remove all words in bc feature vector from gene libraries
24          delete bc from system
25    FOREACH(bc ∈ BC)
26      decrement bc's stimulation count
27    FOREACH(bc ∈ (MC ∪ BC))
28      IF(bc's stimulation count = 0)
29        BC ← BC \{bc}
```

Pseudocode 4.5. Update B-cell population

Assuming the e-mail was correctly classified, in the population update procedure all cells with affinities with the e-mail above a threshold are rewarded with an increased stimulation count. The cell (naïve or memory) with the highest affinity to the e-mail is then cloned. The affinities of the clones with the e-mail are then determined and if one of the naïve cells (clones or pre-existing cells) is found to have an affinity with the e-mail greater than a pre-existing memory cell, then that cell is promoted to a memory cell. If, on the other hand, an incorrect classification occurred then all cells with affinities with the misclassified e-mail over a certain threshold are identified and removed. This is acknowledged as being somewhat draconian, but it was considered necessary to implement this sort of tactic to drive down the rate of false positive classifications, although some thought to relaxing this may be prudent in the future.

No matter which type of classification was performed all naïve cells have their stimulation count reduced, thus any cells stimulated in the procedure above will have their stimulation level reduced again. This results in these cells ending the procedure with the stimulation level they started with while all others have reduced stimulation level. No cell can therefore increase its stimulation level, only maintain it at a current level or allow it to be reduced. This results in a constant replacement of cells and encourages dynamics.

```
1   PROCEDURE clone_mutate(bc,ag)
2     aff ← affinity(bc,ag)
3     clones ← ∅
4     num_clones ← ⌊aff × Kl⌋
5     num_mutate ← ⌊ (1-aff) × bc's feature vector length × Km ⌋ + 1
6     DO(num_clones)TIMES
7       bcx ← a copy of bc
8       DO(num_mutate)TIMES
9         p ← a random point in bcx's feature vector
10        w ← a random word from the appropriate gene library
11        replace word in bcx's feature vector at location p with w
12      bcx's stimulation level ← Ksb
13      clones ← clones ∪ {bcx}
14  RETURN clones
```

Pseudocode 4.6. Cloning and mutation

The process of cloning and mutation which has been used throughout this section is detailed in Pseudocode 4.6. `bc1` is the B-cell to be cloned based on its affinity with `ag`. `Kl` and `Km` are constants used to control the rate of cloning and mutation. The symbol $\lfloor x \rfloor$ denotes the "floor" of x. That is, the greatest integer smaller than or equal to the real-valued number x. This operator is necessary because `num_clones` and `num_mutates` must both be integers.

In this procedure the input `bc1` is first cloned `num_clones` times then each clone is mutated `num_mutate` times by picking a point in the clone's feature vector and replacing the word found in that point with another suitable word pulled from the required gene library.

4.5 Computational Complexity Analysis

Time complexity analysis concerns the derivation of the order of magnitude of time the algorithm will need for execution. To abstract from implementation details, "big O" notation can be used to compare algorithms in a more generic way. A complexity analysis with the results expressed in big O notation can abstract from the architecture and software engineering issues. It does not aim to give figures for the running time but rather aims to place algorithms on a scale of *complexity classes*. The complexity stated refer to the worst-case scenario for the algorithm.

In the following, the initialisation stage is considered separately to the running stage. Also, one iteration of the algorithm is considered when in the running stage. The size of a set is denoted in the usual way. i.e. |BC| represents the size of the set of naïve B-cells. Likewise |MC| is the size of the set of memory B-cells. The size of the union of these two sets is denoted by n.

Given the length of the average B-cell vector is denoted by v, it can be seen that the complexity of the affinity procedure (Pseudocode 4.1) is $O(v)$. Similarly, given the average number of clones per clone and mutate operation is denoted by a and the average number of mutations of each clone is denoted by b, it can be seen that the complexity of the clone and mutate procedure according to Pseudocode 4.6 is $O(a \times b)$. For clarity this clone and mutate procedure will be referred to as having complexity $O(c)$ from here onwards.

Considering first initialisation, as defined by Pseudocode 4.3.

Line 3: Processing an e-mail takes $O(1)$ time.

Line 4: Addition of words takes order v time, and takes place once per initialisation element, |TE| times (line 2), thus lines 2- 4 are $O(v \times |TE|)$.

Line 5: This stage also takes $O(1)$ time.

Line 6-7: A constant time operation but takes place |MC| times, giving $O(|MC|)$. Therefore lines 2-4 and lines 6-7 are $O(v \times |TE| + |MC|)$.

Now considering lines 8-15, but working backwards from 15 for clarity.

Line 15 requires $O(1)$. Line 14 requires one computation of affinity, an operation of $O(v)$ which happens c times from line 13, thus lines 13-15 are $O(c \times v)$.

Line 12 is O(c), while lines13-15 are $O(c \times v)$. Therefore the sum of these is $O(c + c \times v) = O(c \times v)$.

Line 11 is an affinity computation which has not yet been completed thus of $O(v)$, therefore lines 11-15 are $O(v) + O(c \times v) = O(c \times v)$.

Lines 11-15 are executed $O(|MC|)$ times by line 10, and therefore lines 10-15 are $O(|MC| \times c \times v)$.

Line 9 will complete in constant time and lines 9-15 are enclosed in a loop operating |TE| times from line 8, therefore lines 8-15 are evaluated with time given by $O(|TE|) \times O(|MC| \times c \times v) = O(|TE| \times |MC| \times c \times v)$.

Lines 2-7 as computed previously $O(v \times |TE| + |MC|)$ will be dominated by this and can therefore be ignored.

Therefore the running time of the initialisation routine is

$O(|TE| \times |MC| \times c \times v)$.

Considering next the running phase, the time complexity of the algorithm is based on one iteration of the AISEC program (from line 3 of Pseudocode 4.2) initiated by the arrival of a new item to classify and with regard to the numbers of cell in the system. The complexity of one iteration of AISEC is the sum of the complexities of the two procedures: classify and update_population.

Classify – In the classification procedure (Pseudocode 4.4) the affinity between every memory cell and each naïve B-cell and an antigenic pattern is measured. That is, line 3 of Pseudocode 4.4, of complexity $O(v)$ is run n times (where $n = |BC| + |MC|$). This results in classification having complexity $O(n \times v)$.

Update Population – This procedure (Pseudocode 4.5) comprises of a choice of branches of program flow of which one will ever be executed. The branch with the greatest complexity is determined. The following regards Pseudocode 4.5. Considering first the IF branch starting on line 2 of Pseudocode 4.5.

Lines 3-5: Line 5 runs in constant time, Line 4 requires an assessment of affinity and is thus $O(v)$. These two will be run |BC| times. This lines 3-5 run in $O(|BC| \times v)$.

Line 6: Assume no sorting is needed and that the best cell has been remembered, requiring an $O(1)$ operation to retrieve.

Line 7: runs in O(c) time, thus lines 3-7 run in:

$= O(|BC| \times 1) + O(c)$

$= O(|BC| \times 1 + c)$

Lines 8 and 9 : run in $O(a \times v)$, thus lines 3-9 have time complexity given by:

$= O(|BC| \times 1 + c) + O(a \times 1)$

$= O(|BC| \times 1 + c + a \times 1)$

$= O(1(|BC| + a) + c)$

Line 10: runs in $O(1)$ time, as the affinities have already been determined in line 4.

Line 11: runs in $O(|MC| \times v)$ time as every element of MC must be compared with ag.

Combining with the complexity of lines 3-9 as calculated above, lines 3-11 have complexity:

$= O(1(|BC| + a) + c) + O(|MC| \times 1)$

$= O(v(|BC| + |MC| + a) + c)$

Line 12 does not require an affinity calculation as this has already taken place thus Lines 12-15 run in $O(1)$ time.

Line 17-18: require time $O(1)$, and will be done $|MC|$ times according to line 16.

Line 19 completes in $O(v)$ time.

Thus lines 3-11, lines 17-18 and line 19 combined have a final complexity of

$= O(v(|BC| + |MC| + a) + c) + O(|MC|) + O(v)$

$= O(v(|BC| + |MC| + a) + c + |MC| + v)$

As the term $v(|BC| + |MC| + a)$ will dominate the terms $|MC|$ and v, these terms can be ignored. Thus the complexity of lines 3-19 is: $O(v(|BC| + |MC| + a) + c)$

As n has been defined as a substitute for $|BC| + |MC|$, the final complexity for the IF branch is $O(v\ (n + a) + c)$.

The ELSE branch, line 20 is much simpler. Line 22 takes $O(v)$ time and so does line 23. Line 24 takes constant time and so is ignored thus lines 22 and 23 complete in time $O(v + v) = O(2v) = O(v)$.

Both these lines are executed n times on line 21 (recall n has been defined as a substitute for $|BC| + |MC|$), thus the time for lines 20-24, the ELSE branch, is $O(v \times n)$. This complexity is much smaller than the IF branch and so the time complexity for lines 1 – 22 is taken as that of the IF branch as this represents the worst case: $O(v\ (n + a) + c)$.

Line 26 executes in constant time, |BC| times as defined in line 25 thus lines 25-26 are of $O(|BC|)$.

Line 29 is a constant time operation and is executed n times as per line 27, thus lines 27-29 are of $O(n)$.

So the sum of the time complexities of lines 1-19, lines 25-26 and lines 27-29 is:

$= O(v\ (n + a) + c) + O(|BC|) + O(n)$

$= O(v\ (n + a) + c) + |BC| + n)$

|BC| an n will always be dominated by $(v\ (\ n\ +\ a)\ +\ c)$, so they can be removed. Therefore the complexity of the update population operation is

$=O(v\ (\ n\ +\ a)\ +\ c)$

The overall time complexity of AISEC in the running stage is the sum of the complexities of classification and running:

$=O(n \times v)\ +\ O(v\ (n\ +\ a)\ +\ c)$.

Assuming a is not equal to 0, the term $O(v\ (n\ +\ a)\ +\ c)$ will dominate the function.

Therefore the complexity of AISEC in the running phase is:

$O(v\ (n\ +\ a)\ +\ c)$.

4.6 Experimental Results

This section contains an evaluation of the AISEC algorithm and some tests regarding the algorithm's characteristics.

4.6.1 Comparing the Predictive Accuracies of AISEC and Naïve Bayes

To determine the relative classification performance of AISEC, it was necessary to test it against another continuous learning algorithm. The well-known naïve Bayesian classifier was chosen as a suitable comparison algorithm, even though a fundamental assumption of the Bayesian approach, that all attributes are independent, is violated in this situation. In (Mitchell, 1997) the author states, "*probabilistic approaches such as the one described here [naïve Bayesian] are among the most effective currently known to classify text documents*" and due in part to the ability for such a statistical technique to account for the unbalanced penalties of misclassifying e-mail this technique remains popular for classification of e-mail.

4.6.1.1 Experimental Setup

A variation of the naïve Bayesian algorithm was adapted to intercept user input relating to classification accuracy in the same way as AISEC and update itself accordingly. This was implemented according to the equation taken from (Mitchell, 1997).

$$v_{NB} = \underset{v_j \in V}{\text{argmax}}\ P(v_j) \prod_i P(a_i \mid v_j) \qquad 4.1$$

Where the set of class v = {uninteresting, interesting}, $P(v_j)$ is the probability of mail belonging to class v_j and calculated based on the frequency of occurrence of class v_j in the initialisation set. The term $P(a_i|v_j)$ is the probability of the e-mail containing word a_i given the e-mail belongs to class v_j. This probability is calculated using observed word frequencies over the initialisation data. In this modified algorithm these observed word frequencies are updated based on the same user feedback mechanism as AISEC. Consideration must be given to words not yet encountered by the algorithm, yet contained in the e-mail to be classified. The probability of this unknown word occurring in either class of e-mail cannot be taken as 0, as the result would resolve to 0. Instead, it is given a probability of occurrence of $1/k$, where k is the total number of words known to the system.

Experiments were performed with 2268 e-mails of which 742 (32.7%) were manually classified as uninteresting, the remaining 1526 (67.3%) were assumed of some interest. Due to the unsuitability of the few publicly accessible e-mail datasets which are traditionally used for single shot learning, the algorithm could not be tested on a standard e-mail dataset. Some problems encountered were a lack of e-mail header information and the messages being pulled from various Usenet newsgroups and thus not containing much consistency. In addition, these sets were all classed as spam/non-spam rather than interesting/non-interesting. Hence a dataset containing e-mail received by the author was used. All e-mails used were received between October 2002 and March 2003, and importantly, their temporal ordering was preserved.

Parameter	Value	Range
`Kc` (classification threshold)	0.2	0 - 1
`Ka` (affinity threshold)	0.5	0 - 1
`Kl` (clone constant)	7.0	>= 1
`Km` (mutation constant)	0.7	0 - 1
`Ksb` (Naïve B-cell stimulation level)	125	0 – size of test set
`Ksm` (Memory cell stimulation level)	25	0 – size of test set
`Kt` (initial number of memory cells)	20	0 – size of test set

Table 4.1. Parameters used for testing AISEC

Only the words contained in the subject and sender fields of the e-mails were used, but the sender information also included the return address, as these fields may differ. The fields were tokenized using spaces and common non-alphanumeric characters as delimiters. Each token was then inserted into a separate element of the correct feature vector. Simulated user feedback was given to both algorithms after the classification of each e-mail as the real class of each e-mail is known. Throughout the algorithm a single pseudo-random number generator was used. This was an implementation of the

Mersenne Twister algorithm (Matsumoto & Nishimura, 1998) written in Java by Sean Luke (Luke, 2000). During the reported runs of the AISEC algorithm, the same values for all parameters were used. These values (shown in Table 4.1) were arrived at by empirical testing during development, and as a result tend to work well over this dataset. A legal range for each parameter is also indicated.

4.6.1.2 Computational Results for Continuous Learning

The naïve Bayesian algorithm was trained on the first 25 e-mails, when ordered by date/time received. This set will contain both classes of e-mail. In contrast, the AISEC algorithm was trained on the first 25 uninteresting e-mails only. The remainder were used as the continuous test set. Thus the continuous test set contained 2243 e-mails, all presented in the correct temporal order.

Unlike traditional "single shot" learning, where the test set has no impact on the internal state of the algorithm (the classifier is set during the initialisation then will not change), this investigation addresses continuous learning, where the algorithm is continually receiving e-mails to be classified. Each time a new e-mail is classified the algorithm can use the result of this classification (the information about whether or not the class assigned was correct) to update its internal representation. This continuous learning scenario calls for a slightly different measure of accuracy to that which is normally applied. Conceptually, as there is no fixed "test set", the algorithm keeps track of its performance over the past 100 classification attempts. As each e-mail is classified an average accuracy over these previous attempts is reported. The final classification accuracy is determined by taking the mean of these values.

A successful result in this case would be a classification accuracy for AISEC to be found to be at least the same as the classification accuracy of the naïve Bayesian algorithm. A successful result would also include a greater value for both recall and precision figures over a naïve Bayesian algorithm. If both these criteria are met then the classification characteristics for AISEC are more desirable then those of naïve Bayesian.

Algorithm	Classification Accuracy	Recall	Precision
Bayesian	88.05%	67.76%	93.93%
AISEC	88.77% (0.95)	80.06% (3.99)	82.76% (2.40)

Table 4.2. Predictive accuracy for AISEC and Naïve Bayes

As AISEC is non-deterministic the results presented in Table 4.2 are the mean values for 50 independent runs using a different random seed each time. The value in

brackets is the standard deviation. Since it is deterministic, the result for the naïve Bayesian algorithm has no associated standard deviation as only a single run was required.

Table 4.2 summarises the results over the continuous test set. Precision is the percentage of messages classified as uninteresting which really are uninteresting, and recall is the percentage of uninteresting messages classified as uninteresting as defined in Chapter 2. AISEC shows a better balance between these two measures. The naïve Bayesian classifier achieves a significantly higher precision at the expense of recall. This demonstrates the naïve Bayesian classifier blocks fewer uninteresting messages, but the ones it does block are more likely to be uninteresting. This is due to a Bayesian classifier's bias towards assigning the majority class to an example, since the class prior probabilities are used to compute the probabilities of the classes given the feature vectors.

It is possible to determine how significant the difference between the accuracy for the Bayesian and the mean accuracy of AISEC. As the observations consist of a single value and a set of observations it is possible to compare these using the normal distribution (Alder & Roessler, 1968; Creighton, 1994). A null hypothesis can be stated, that the accuracy for the Bayesian and mean accuracy of AISEC do not differ significantly.

$$z = \frac{X - \upsilon}{\sigma}$$
$$z = \frac{88.05 - 88.77}{0.95} \qquad 4.2$$
$$z = -0.7579$$

Looking this figure up on a standard normal distribution table reveals a probability of the null hypothesis holding of $P_{null} = 0.33$, this is greater than the threshold of 0.05 for statistical significance. Therefore the null hypothesis cannot be rejected and as such this higher mean accuracy is not statistically significant, although AISEC will perform with greater predictive accuracy on average most of the time. Thus, it may be reasonably concluded that AISEC performs with accuracy comparable to that of the naïve Bayesian algorithm, but with somewhat different dynamics.

The line chart, Figure 4.4, shows the changing predictive accuracy after the classification of each mail by number of e-mails classified. This uses the accuracy measure described above and therefore details the results for the test set from 100 classification attempts onwards. This chart is drawn using the mean of the same 50 runs as used to construct Table 4.2. Error bars have been omitted for clarity but the overall

error is shown by way of a standard deviation figure in Table 4.2. It can be seen that, but there are certain areas where the changing data causes them to behave very differently. Of interest are the areas 1,000 to 1,200 and 1,900 to 2,250 e-mails classified. In both situations AISEC exhibits an increase in accuracy while there is a decrease in accuracy from the naïve Bayesian algorithm.

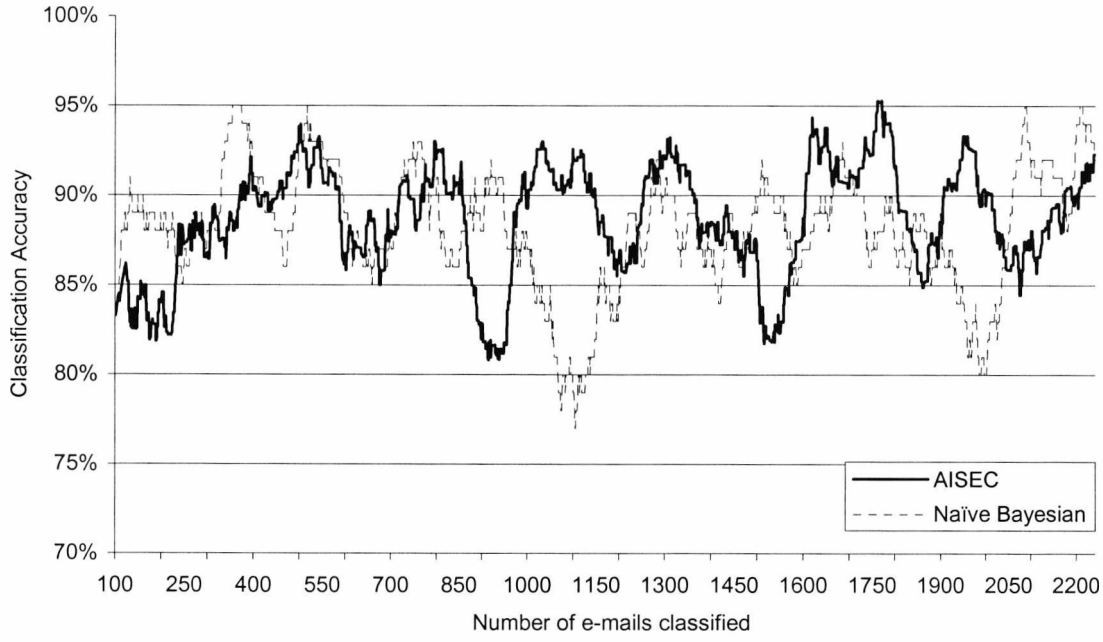

Figure 4.4. Change in classification accuracy by e-mails classified

One explanation of this could be that AISEC is faster to react to sudden changes. Consider for example a word that has been very common among uninteresting e-mail. AISEC will represent this detail as the presence of this word in a number of B-cells. The Bayesian algorithm will represent this as a high frequency of occurrence in this uninteresting class compared to the other class. Consider now this word begins to be used in interesting e-mail. The AISEC algorithm will react quickly by immediately deleting any cells containing this word that would result in a misclassification. By contrast the Bayesian algorithm will react by only incrementing the frequency count of this word in the interesting class by one. Given the word has been common in uninteresting e-mail for some time the frequency of occurrence in the uninteresting class will still be large compared with the frequency of occurrence in the interesting class, thus resulting in a negligible effect on the differences between the calculated final class probabilities. Only after this word has been used a number of times in confirmed interesting e-mail the differences in the usage frequencies may even out and the

difference in the probabilities of this word being used in either class significantly decrease.

This test has been undertaken with only one single dataset and this is an acknowledged limitation of this section. The effects of AISEC as a non-deterministic algorithm were negated as much as possible by determining the mean of a number of runs, and as such it is believed that the conclusions drawn are reasonable. However for conclusions to be drawn about the generality of these results it is recommended that further tests are run with different datasets.

4.6.1.3 Assessing Performance with Concept Drift

Experiments were also undertaken to investigate the hypothesis that AISEC would track concept drift. This was done by presenting the e-mails in the test data to each system in a random order. The ordering was changed for each of the 50 runs used. A successful outcome of this test would show that the classification accuracy is decreased by not presenting the ordered set of e-mails. Although it is acknowledged here that it is intractable to *prove* the original result contained concept drift.. The results for the naïve Bayesian algorithm contains a standard deviation this time as the data is presented in a different order for each test and therefore the output is no longer deterministic. Results are shown in Table 4.3.

Algorithm	Classification Accuracy	Recall	Precision
Bayesian	84.17% (2.33)	55.49% (7.71)	93.59% (3.00)
AISEC	82.53% (3.02)	74.52% (7.35)	73.14% (5.52)

Table 4.3. Assessment of performance with randomised data

Results showed that the mean accuracy of AISEC has indeed reduced, indeed the mean classification accuracy has reduced by a considerable amount. This tends to suggest that AISEC is indeed tracking concept drift as expected. The mean classification accuracy of the Bayesian algorithm has also reduced, although the performance of the naïve Bayesian algorithm has not suffered as much as AISEC. Also noticeable is the enormous change in the standard deviation values for both AISEC and naïve Bayesian between these results and those shown in Table 4.2. In this case, AISEC has changed from having a standard deviation of 0.95 to 3.02. The precision value for the naïve Bayesian algorithm is unexpectedly similar to that in Table 4.2. While the remaining precision and recall values for both algorithms are, in this situation, reduced. The result would seem to suggest that AISEC reacts somewhat badly to a situation

where rapid or almost random concept shift is occurring, rather than concept drift, but AISEC certainly does not fail altogether.

4.6.1.4 Change in B-cell Population Size

This test is used to ensure the population control mechanisms of AISEC are working as expected. A population based algorithm such as AISEC, will tend to have reduced in usefulness if the size of the population grows uncontrollably. As each cell must be compared with each unknown e-mail (antigen), the more cells in the system the less efficiently the system will run. It is therefore desirable to minimise the size of the sets of cells, although as the system learns new information as the run continues, it is inevitable the more cells must be produced to store this knowledge. A successful result to this test will show a small increase in population size as the algorithm runs, ideally levelling off after a time.

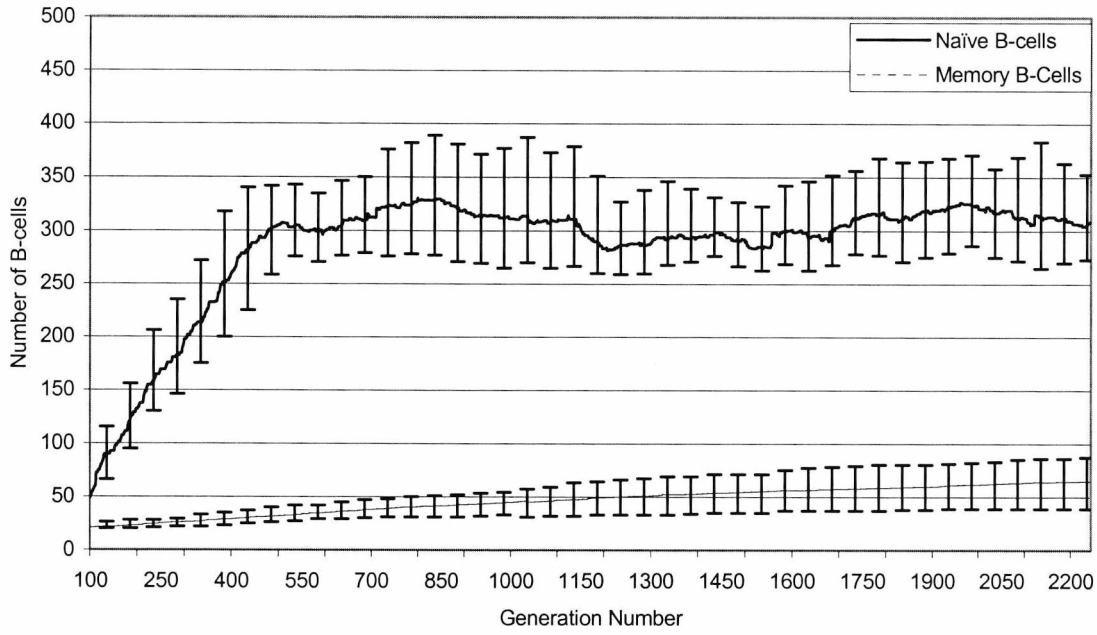

Figure 4.5. Change in B-cell population over time

Figure 4.5 shows the variation in size of the naïve and memory B-cell populations during the same 50 runs of AISEC used to produce Table 4.2. Error bars have been placed every 50 iterations. As expected there are many more naïve B-cells compared with memory cells. The number of cells in the naïve B-cell population, after an initial rapid growth period, appears fairly stable. There is an increase over the duration of the testing (348 naïve cells at 519 e-mails compared with a final value of 366 cells at 2243 e-mails), but this is small relative to the size of the population. All changes appear

steady but it is impossible to tell if the slight increase in numbers is due to the nature of the data rather than an underlying problem with the algorithm. On the basis of these results the process of naïve cell death after a given number of user signals appears an effective control mechanism.

Similarly, the memory B-cell population size is showing expected behaviour. There is no rapid change in the size of this cell set, as would be the case if many of its elements were subject to deletion at once. This would be evidence that the algorithm had failed in the placing of many memory cells. The memory cell population size is increasing over time, but at a decreasing rate. From this evidence it is impossible to tell if the algorithm will reach a state where the creation of new memory cells is exactly balanced by cell death, but since the population size appears to be levelling off as the number of classification attempts increases, this strategy again appears to be working broadly as expected.

4.6.1.5 Summary of Results

The results gained from testing AISEC were encouraging. The classification accuracy was shown to be comparable to a naïve Bayesian algorithm – one of the most popular strategies for classification of e-mail. The precision and recall values were also shown to be reasonable on the same continuous test set. The performance of AISEC on randomised data where no concept drift was able to occur was much lower than the previous tests in which e-mails were presented in a natural ordering. This suggests that AISEC does indeed track concept drift quite effectively. The quality of AISEC's cell population control mechanism was evaluated by measuring the size of the populations as the classification progressed. It was encouraging to see the growth of the naïve B-cell population was not unmanageable and while the memory cell population did grow over time, it did not grow quickly.

4.7 Sensitivity Analysis of AISEC Parameters

The dynamic behaviour of AISEC can be controlled by the algorithm's parameters, of which there are many. Therefore there is some need to examine the influence of the algorithm's parameters on the performance of the algorithm. In general, this analysis has been motivated by the increased use of AISEC and therefore pressure to allow practitioners informed choice in the parameter values selected. For example the implementations in (Kilgour, 2004) and (Ayara et al., 2005; Ayara, 2006) in which the

author describes an implementation of AISEC that, with little revision from the published algorithm, was deployed in a hardware fault detection scenario.

The investigation in this section is an extension of the M.Sc. dissertation (Jang, 2004) in which parameter adjustment was investigated. This original investigation is extended by using it as a basis of an investigation into the optimum parameter values on this dataset. The tests from (Jang, 2004) are completely re-run but with two significant differences which both effect the output and ensure the result is a) meaningful in the context of this thesis and previously published work regarding AISEC, and b) reveal a more meaningful result for a user.

Firstly, the investigation by Jang used a set of baseline parameters which were different from those used previously in this chapter and so the baseline parameter set is brought into line with those parameter values used previously. Secondly, and importantly, this investigation measures false positive rate and false negative rate in addition to accuracy. Using these metrics the charts presented in this chapter become more meaningful to a user as the misclassification costs are made explicit, something not obvious in the original investigation where true positive rate and true negative rates were used. Re-running these tests has also allowed the visualisation of error bars on the final charts, a significant oversight in the original work. As the visualised output of the tests is considerably more revealing in terms of misclassification costs and error bars are added, the conclusions are not the same between this chapter and the previous work and as such it is thought that this analysis is worthwhile both in itself and as a complement to the previous work.

It should be noted that the test data used in the experiments presented in this section are different from the data used in the previous evaluation section. A 50%/50% class split was used to negate an uneven class distribution having an effect on the result. While arbitrary, an equal class distribution is a reasonable default position. In addition, test the effect of using differing numbers of initialisation examples a larger sample of initialisation examples was required. Even though a different data set has been used, the generality of this chapter will not be affected in the sense that this investigation is not for evaluation of the algorithm itself, but for parameter analysis within the algorithm.

4.7.1 Parameters

As stated in section 4.4, there are seven major parameters in the AISEC algorithm:

- Classification threshold (`Kc`)
- Affinity threshold (`Ka`)

- Clone constant (Kl)
- Mutation constant (Km)
- Initial naive B-cell stimulation value (Ksb)
- Memory cell stimulation level (Ksm)
- Initial number of memory cells generated during initialisation (Kt)

The following is also investigated:

- Number of examples used during initialisation (Kts)

To summarise, the classification threshold influences the decisions on the class of incoming e-mails. If the antigen (e-mail) shows affinity for any immune cell higher than this threshold, the message is classified as uninteresting. The affinity threshold is responsible for population manipulation and dynamics throughout the algorithm. It influences the selection of immune cells for reward or punishment, depending on the classification results. The clone constant (Kl) determines the maximum number of clones a cell may produce. The mutation constant (Km) determines how many times a mutation will occur to a cloned cell. Naive B-cell stimulation level and memory cell stimulation level influences the potential life span of a B-cell and a memory cell respectively. Initial memory cell set size is used to determine how many memory cells will be selected from the initialisation data set in the initialisation phase. In addition to these parameters, the initialisation set size is also considered worthy of investigation. Table 4.5 details the parameters that are investigated. This is based on Table 4.1, but a value for Kts, the size of the initialisation set, is now included. During this analysis the values used in this investigation use the entire reasonable range as defined by the "start" and "end" columns.

4.7.2 Experimental Protocol

Table 4.4 details information on the data set used. E-mails used in this data set were collected by the author of (Jang, 2004) during the period from September 2003 to June 2004. It was necessary to reconstruct this e-mail set so a class split of exactly 50%/50% was used. This is so that a fair and known bias is used and so that there are enough e-mails available for testing the behaviour when the size of the initialisation set is varied. Collected e-mails were separated into two groups, one for initialisation and one for running. For the initialisation set, 360 uninteresting e-mails were used. None of these were re-used in the continuous test set. The actual initialisation set used for each

experiment was constructed by extracting the required number of uninteresting e-mails from this set of 360 according to `Kts` (initialisation set size). This allowed for experimentation with different `Kts` values without any change in the contents of the running set.

Test set	Interesting e-mails	Uninteresting e-mails	Total
Initialisation set	0	360	360
Running set	189	189	378
Total	189	549	738

Table 4.4. Number of e-mails per class in data sets

Value ranges for each parameter for each experiment are shown in Table 4.5. During each experiment, only one parameter value was varied, whilst the others are set to a default value. The parameter being varied has a lower limit or start point, an upper limit or end point, and an increment value. Each experiment was run 50 times and the results of each run were averaged. In Table 4.5, the column "Default" shows fixed default values for each parameter when that parameter is not being varied, the values in this column are the same as before. The number of different values tested for each parameter is shown in the column named "Values"; for example, when investigating `Kc`, there are 50 different values possible from 0.02 to 1.0 using an increment of 0.02.

Parameter	Start	Increment	End	Default	Values	Total runs
`Kc`	0.02	0.02	1.0	0.2	50	2500
`Ka`	0.02	0.02	1.0	0.5	50	2500
`Kl`	1	1	50	7.0	50	2500
`Km`	0.02	0.02	1.0	0.7	50	2500
`Ksb`	10	10	500	125	50	2500
`Ksm`	1	1	50	25	50	2500
`Kt`	1	1	25	20	25	1250
`Kts`	20	20	360	25	18	900

Table 4.5. AISEC parameter configuration

4.7.3 Evaluation Metrics

Once a test has been run and the confusion matrix has been computed, using the metrics of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative(FN), it is possible to calculate the conventional measure of predictive accuracy used in the classification literature, which is defined as: (TP + TN) / (TP + FP + FN + TN). This standard measure of accuracy is used throughout this section.

It should be noted, that the standard classification accuracy calculation has the disadvantage that it treats the two types of classification errors (FP and FN) in the same way. In the context of this research, it is important to perform a finer-grain analysis of the influence of the parameters in the predictive performance of the algorithm. This will contribute to an understanding of which kind of classification error tends to be increased or decreased as the value of a parameter is varied over its legal range. As a result, users will be able to make a more informed choice about the value of a parameter to be used in their application domain, depending on the relative importance that they assign to minimizing FP or minimizing FN. Therefore, in addition to the conventional classification performance metric, this section reports a separate measure of predictive accuracy for each of the two kinds of classification errors, namely the False Positive Rate – FPR (Equation 4.3) and the False Negative Rate – FNR (Equation 4.4). Of course, accuracy is to be maximized, whereas FPR and FNR are to be minimized.

$$FPR = \frac{FP}{FP + TN} \tag{4.3.}$$

$$FNR = \frac{FN}{FN + TP} \tag{4.4.}$$

4.7.4 Parameter Analysis

This section presents the results of the investigations into the influence of the parameters of AISEC on its performance. Experiments were carried out based on the protocol described previously. For each parameter a hypothesis is given regarding the expected observations, a chart is shown showing the behaviour and conclusions are drawn based on the chart that either confirm or refute the hypothesis.

For each parameter value a number of separate runs were performed (see Table 4.5) with different random seeds each time. The mean value over all the runs is used to form the charts displayed in this section. For each parameter, the change in accuracy, false positive rate (FPR) and false negative rate (FNR) is observed. Error bars are provided for the predictive accuracy figures, but for the sake of clarity the error bars for the other two series are omitted. These are available upon request.

While a number of separate runs are performed to negate the effects of random variation between them, only one dataset is used in this investigation. It is thought reasonable to do this as it is expected that the most significant parameters will remain so across datasets. It is likely the shapes of the curves on the charts will be broadly similar.

However the exact figures are liable to change between datasets and as such these more specific details should be extrapolated with caution.

4.7.5 Context

As made clear previously, this section is a continuation of the M.Sc. dissertation (Jang, 2004). In that document, the author makes a number of hypotheses regarding the way attribute values will affect the behaviour of the algorithm. It is important to repeat these explanations here to give context to the conclusions reached in the following subsections. Expectations regarding behaviour caused by each attribute can be reduced and summarised as follows:

Kc. An aspect of `Kc` is that it defines the algorithms tolerance level to identify uninteresting e-mails. The lower the value of Kc, the higher the tolerance. A low Kc may allow a low affinity antigen (representation of an e-mail with unknown class) to be classified as uninteresting (positive classification). However, if the tolerance level is set too high, there emerges the danger that the immune cells recognize negative class antigens (false positive classification).

Ka. `Ka` is involved in a number of processes, each having a complex effect on the others. `Ka` works over the three different processes: selecting B-cells to give rewards, selecting memory cells to lower their stimulation level (when a similar, presumably better, memory cell is added) and removing cells when false positive classification occurs. A low `Ka` value increases the chance of generating more B-cells in the initialisation phase, which may increase the population size. However, as more classifications occur, a low `Ka` value may allow an increased number of cells to be removed from the population, thus leading to a reduction in cell population. It is likely that this would cancel the effect of the larger cell population after the initialisation. In contrast, high `Ka` values allow only high affinity cells for the misclassified antigens to be removed from the algorithm. This may encourage the population to grow in the long run, resulting in an increase in FPR. High `Ka` may disable the feature of cell removal in false positive classification, allowing the algorithm to be slow in adapting itself to the changes such as user preference and resulting in an increased FPR.

Kl. Kl is responsible for the number of clones to be created during the cloning process. It applies only once a true positive classification has occurred and applies to the cell showing highest affinity amongst all B-cells and memory cells combined. High Kl may result in an increase in naïve B-cell population sizes as more clones may be generated. This may increase redundancy because clones are duplicates of the most competent cell, and the mutation applied in some cases may be insufficient to move the clone outside the affinity threshold.

Km. Km is responsible for the rate of mutation. High Km may result in an increase of diversity in immune cells by generating more variants from selected cells. A high Km may increase FPR by increasing diversity of B-cells. Diversity (and thus the generalization) of B-cells affords the algorithm the ability to recognise unknown e-mails. Considering that the unknown e-mails contain not only uninteresting e-mails but also interesting e-mails, there is still the possibility that FPR may increase.

Ksb. Ksb is responsible for a B-cell's initial life span and is therefore also related with cell death. High Ksb may allow those non-stimulated B-cells to survive longer, which results in an increase in the overall cell population size. But in this case, it is likely that allowing the non-stimulated cells live too long will also allow FPR to increase. This is because non-stimulation may indicate a change such as a drift in user preference. Positive classification may accelerate B-cell death in general. Some selected B-cells can be stimulated for being near to correctly classified antigen, but even in this circumstance, other cells are to be suppressed. Upon a false positive classification, all B-cells are suppressed by reducing their stimulation level. However, in addition to this, the cells within Kc of the misclassified antigen will be removed. Therefore, a high rate of positive classification and a low FNR will be the ideal combination for drastic cell death.

Ksm. Ksm is responsible for the memory cell's initial life span. The stimulation level of a memory cell is affected by the introduction of a new memory cell with a higher affinity level for the existing memory cell than Ka for all memory cells. High Ksb allows memory cells to survive longer in such a situation, not so with low Ksb, which may lead to increase in cell population. But in this case, it is possible that by allowing the memory cells to live longer, may allow FPR to

increase because more e-mails may be recognised and classified as uninteresting messages by old memory cells. As with Ksb, a high Kc reduces the chance of positive classification, which also decreases the chance of Ksm's involvement, and vice-versa. Therefore, the influence of Ksm is considered to be more prominent with a low Kc than with a high Kc.

Kt. Kt (the number of elements in TE) defines the number of initial memory cells generated by the initialisation phase. If Kt is a low value, then it is more likely that the affinity function returns a value less than Ka. In this case, no clones will be produced and the resultant B-cell set will be empty. Therefore, a high Kt may guarantee more B-cells are produced during the initialisation, compared with a low Kt value. Increasing Kt results in a decrease in the size of the actual initialisation data; Kt memory cells are selected from the initialisation set, and the remaining cells are used as initialisation examples. Considering that Kt is only involved in initialisation; the effect of Kt is expected to be significant at the beginning of the running phase and then become less pronounced as classification progresses.

Kts. Like Kt, Kts also influences the initial cell population during the initialisation phase of the algorithm. Kt initial memory cells are selected from Kts initialisation e-mails in the initialisation phase, and the remaining e-mails are used as initialisation antigens. In general, high Kts is considered to increase the chance for more B-cells to be generated by providing more initialisation examples.

4.7.6 Results

This section contains the results of the parameter tests as described by the protocol in section 4.7.2 as evaluated using the metrics in 4.7.3. For each parameter a chart is shown detailing the behaviour of the algorithm as that single parameter varied. An explanation of the chart is then attempted. Recall the purpose of this investigation is to assist users of the algorithm who may wish to utilise its generic learning capability. Therefore, for each parameter, some recommendations are made for the parameter settings for this task on this dataset but also suggestions are made as to the general attribute settings that could be used to achieve a desired classification characteristic.

4.7.6.1 Classification Threshold (Kc)

Figure 4.6 shows supporting evidence for the statement in 4.7.5. Recall, a positive classification occurs when the affinity is greater than Kc. Thus with a high Kc (around 0.5 and above) no antibodies have affinity greater then Kc with any antigen thus all antigens are classified negative. At a low Kc value such as 0.02, FPR is above 90%, which is comparatively high compared to the FNR value of less than 10%. As Kc increases, the situation becomes inversed; FPR decreases towards zero while FNR increases towards 100%. The distribution of accuracy forms a rough bell shape at low values of Kc, due to the fast increase in FNR and quickly falling FPR but as the value of Kc increases the accuracy levels off. It is clear from the chart that the optimum setting for this parameter with regard to overall accuracy is between Kc = 0.16 and Kc = 0.22. It is interesting to observe that the error bars immediately either side of this optimum are large with respect to the error at the two or three optimum points.

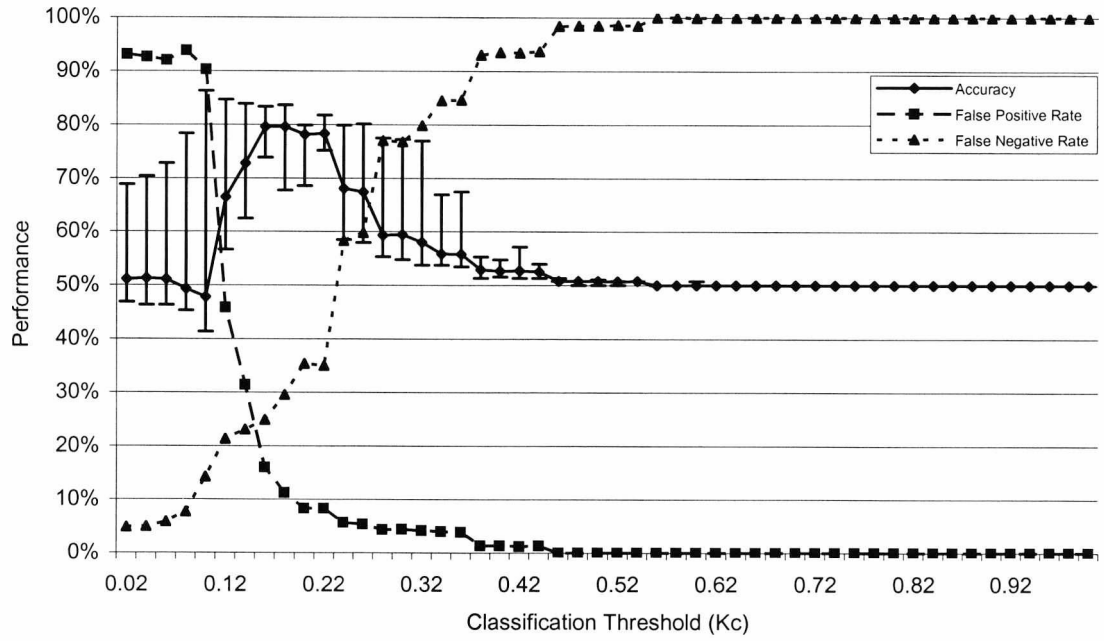

Figure 4.6. Influence of Classification Threshold (Kc)

The actual chosen value of Kc will depend to what extent the task in hand will require the minimisation/maximisation of FPR or FNR. In the case of e-mail classification a value at the higher end of the range will be appropriate to minimise FPR. It can be seen that this parameter has rather a large effect on accuracy, which varies from approximately 50% to 80%, depending on the value of Kc. This is a large variation compared with most other parameters, therefore the value of Kc should be chosen with care. It should also be noted how the error in accuracy from Kc=0.46

onwards is almost ±0%.This can be explained by the distribution of data. In the running set there is an equal distribution of the data between the two classes. At a high Kc it can be observed that the Kc value is too high for any example to be classified as positive and therefore in every run every example is classified as negative, resulting in an accuracy of exactly 50% and exactly 0 error.

4.7.6.2 Affinity Threshold (Ka)

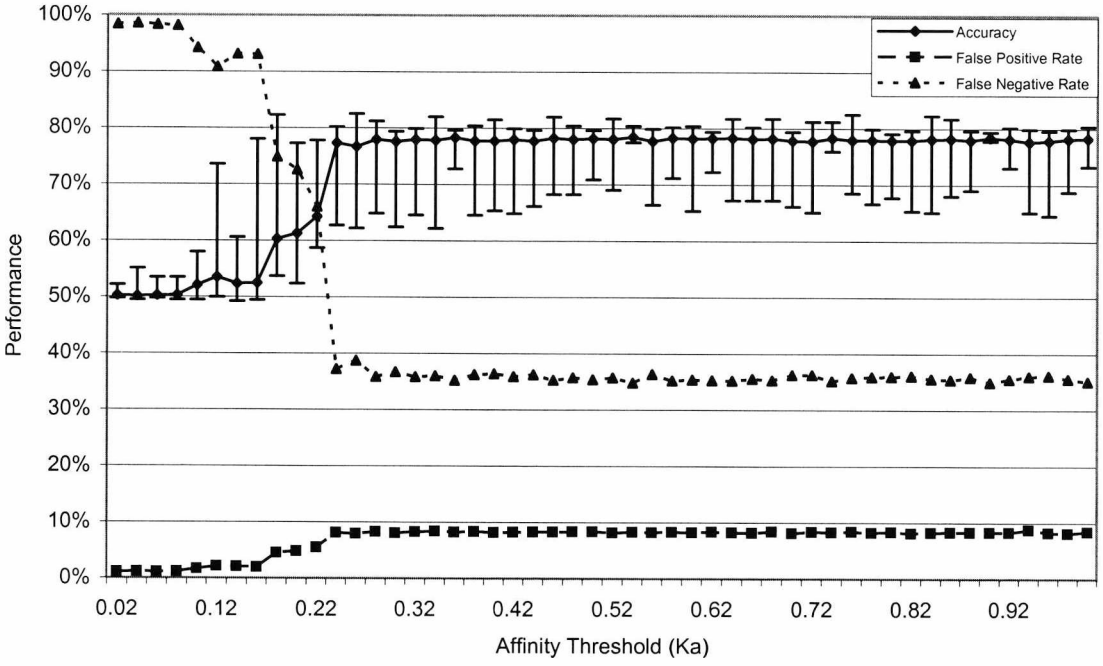

Figure 4.7. Influence of Affinity Threshold (Ka)

Figure 4.7 shows that FNR decreases from above 98% to below 40% as the Ka value increases. Inversely, FPR increases from 1% to around 10% as the Ka value increases. Unlike Kc, for Ka values higher than around 0.26, all observations appear fairly stable. It is believed that a Ka at a certain level, around 0.26 here, already disables the factors that may affect performance such as refining cells by removing them in a false positive classification, therefore values above this level make little difference. In other words, there are no more B-cells close enough to the antigens representing interesting e-mails with affinities between them greater than the Ka value. Like Kc, Ka also has a large effect on the overall accuracy of the algorithm with this value ranging again from approximately 50% to just under 80%. It can be seen that the error for a low Ka value ($Ka < 0.12$) is much smaller than values above. This behaviour is hard to explain. The lower error at low values of Kc may be explained in part by the distribution of the data. An overall accuracy of approximately 50% here suggests almost all examples are being

classified as one single class, resulting in a low error. The moderate error when Ka > 0.18 may be explained by the stabilising factors above, but the small region between Ka = 0.12 and Ka = 0.18 with large error bars can only be explained by these two effects interacting in this interval.

4.7.6.3 Clone Constant (Kl)

Figure 4.8 suggests that the influence of Kl over classification performance is small compared to that of Kc and Ka. It would appear that the clones generated placed around their parents in the search space do not bring about a drastic change in the algorithm's recognition power as the existence of their parents guarantees a certain degree of accuracy. However, what can be seen is that the increasing clone constant creates a situation in which an example is slightly more likely to be classified as positive, leading therefore to an increasing FPR. This also has the effect of reducing the FNR. However, the overriding surprise here is how little this attribute value affects the outcome.

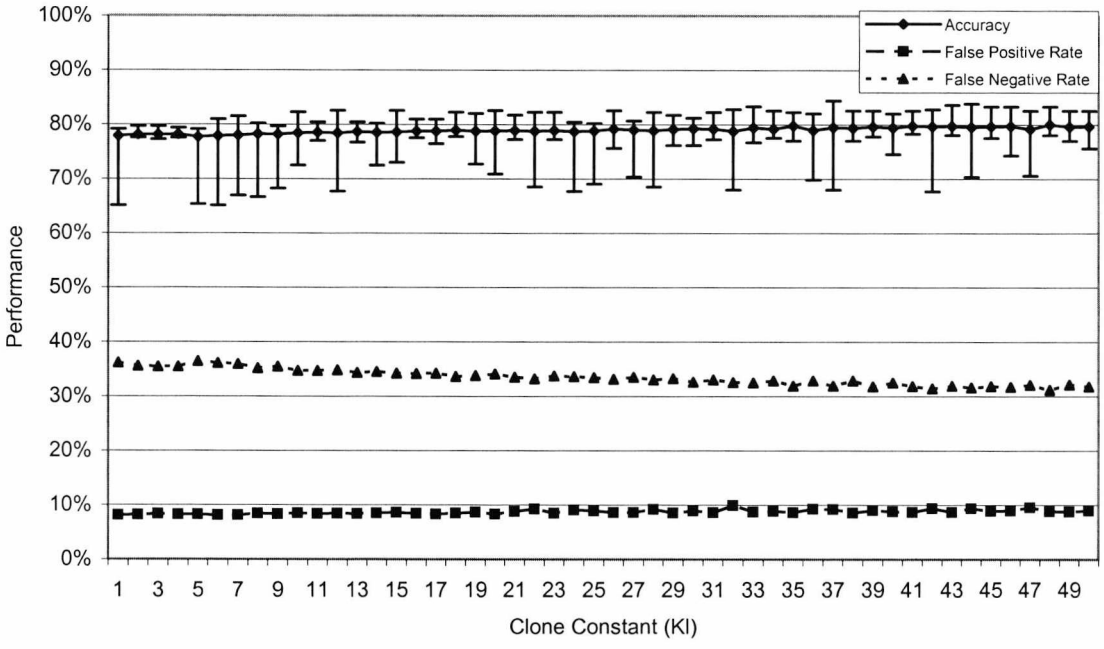

Figure 4.8. Influence of Clone Constant (Kl)

4.7.6.4 Mutation Constant (Km)

Figure 4.9 suggests that the influences of Km over the predictive performance measures is not so significant as other parameters. It appears that B-cell diversity, achieved by high rate of mutation, contributes to a slightly increased false positive classification rate, but the difference is small.

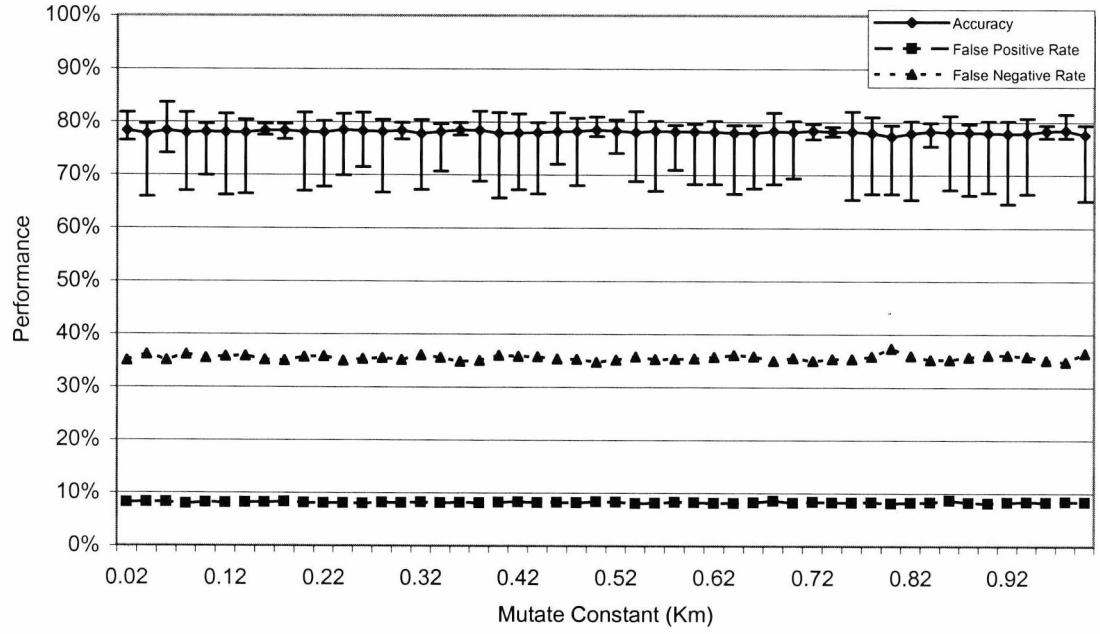

Figure 4.9. Influence of Mutate Constant (Km)

4.7.6.5 Initial Naive B-cell Stimulation Level (Ksb)

According to Figure 4.10 the influence of Ksb does not seem significant, it seems that there are no noticeable correlations between Ksb and the metrics measured.

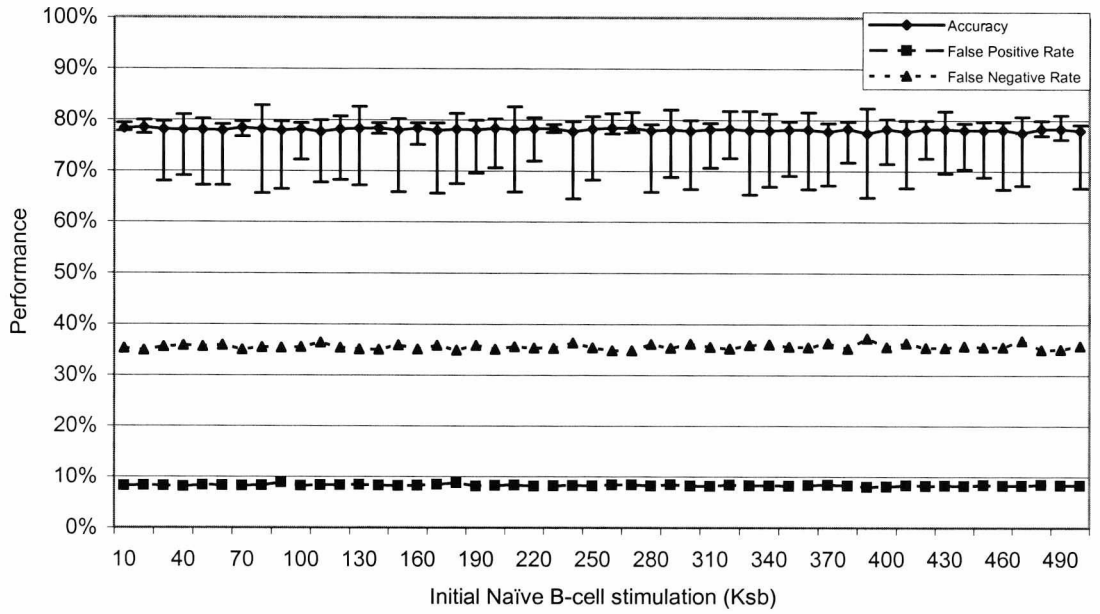

Figure 4.10. Influence of Initial Naive B-cell stimulation (Ksb)

4.7.6.6 Initial Memory Cell Stimulation Level (Ksm)

According to Figure 4.11, the influence of `Ksm` was not significant in general. The accuracy, FPR and TPR appear stable for all values of `Ksm` tested.

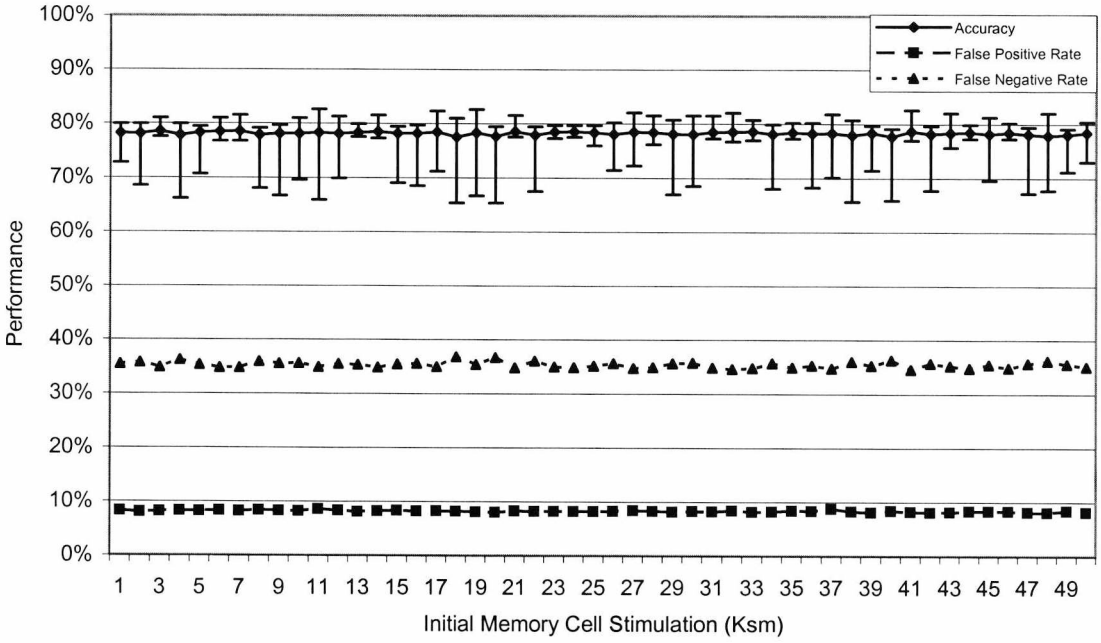

Figure 4.11. Influence of Initial Memory Cell Stimulation (`Ksm`)

4.7.6.7 Number of Initial Memory Cells Generated by Initialisation (Kt)

During the testing of this parameter, as `Kt` was increased, the number of initialisation antigens available decreased. An interesting observation from the tests is that when `Kt` is high, the resultant B-cell set is sparse. This is because, given there are a large number of memory cells, then there are a small number of cells left in the initialisation set. Therefore, there is an increased likelihood that none of the initialisation items will be of sufficient affinity, clone and populate the B-cell set. This can be seen in Figure 4.12 as an increase in the size of the error bars on the left-hand side of the chart.

Figure 4.12 suggests that influence of `Kt` was significant compared with `Kl`, `Ksm` and `Ksb`. The FPR starts low but gradually increases, suggesting the more memory cells are produced. This will create a situation where an example is more likely to be classified as positive. However, the FNR is effected by this situation as well, and given a value of `Kt` = 1 a large number of examples are misclassified. It seems that this FNR value was biased by the extreme situation and the single memory cell did not recognize any antigen during the running phase and all e-mails were classified as interesting

(negative class) messages. This provides evidence that the algorithm becomes vulnerable to be biased by the selection of initial memory cells when initial memory cell size is small. Even though, in the tests, these cells are selected randomly. This bias can be seen as a huge error in the accuracy for low values of `Kt`. When a small number of bad examples are selected the accuracy may be lower than chance (<50%) but if a good cell or few cells are selected then the accuracy may be almost good as for high values of `Kt`. In general, it appears that the greater the `Kt` value the greater the accuracy. As the FNR and FPR do vary as the `Kt` value changes, the value should once again be chosen considering whether FPR or FNR needs to be minimised or maximised for the particular task in hand.

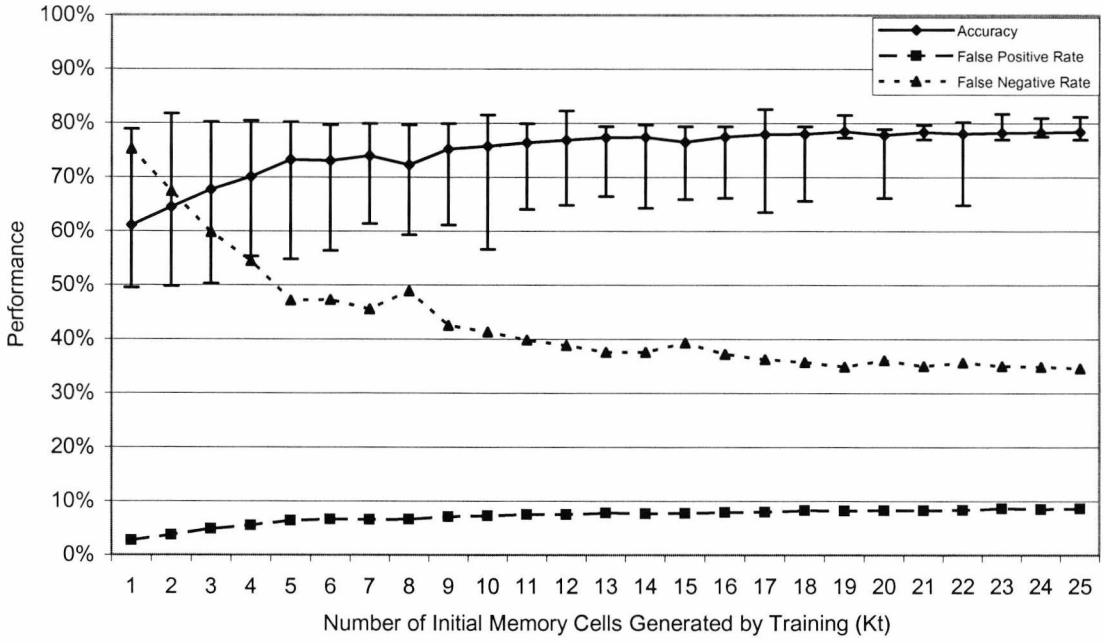

Figure 4.12. Influence of the Number of Initial Memory Cells Generated by Initialisation (`Kt`)

4.7.6.8 Number of Initialisation Examples (Kts)

Figure 4.13 suggests that `Kts` was influential especially over the FPR and variation on accuracy. The influence of `Kts` over FNR was relatively small. The increase in positive classification leads to the decrease in B-cell population. Considering that cells are suppressed only in false positive classification, it seems that the number of false positive classification increased. The number of initial B-cells produced through initialisation was not significant over the increase in the initialisation set size. The number of initial B-cells increased from 0 up to around 6 on average. A significant increase in the gene library size is observed. It appears that the main factor to increase

the false positive classification rate was the increase in gene library size. Gene libraries are involved in the mutation process increasing the diversity of B-cells. Therefore, the diversity of B-cells, built by a large initialisation set, influenced the performance of the algorithm in a negative way in this test.

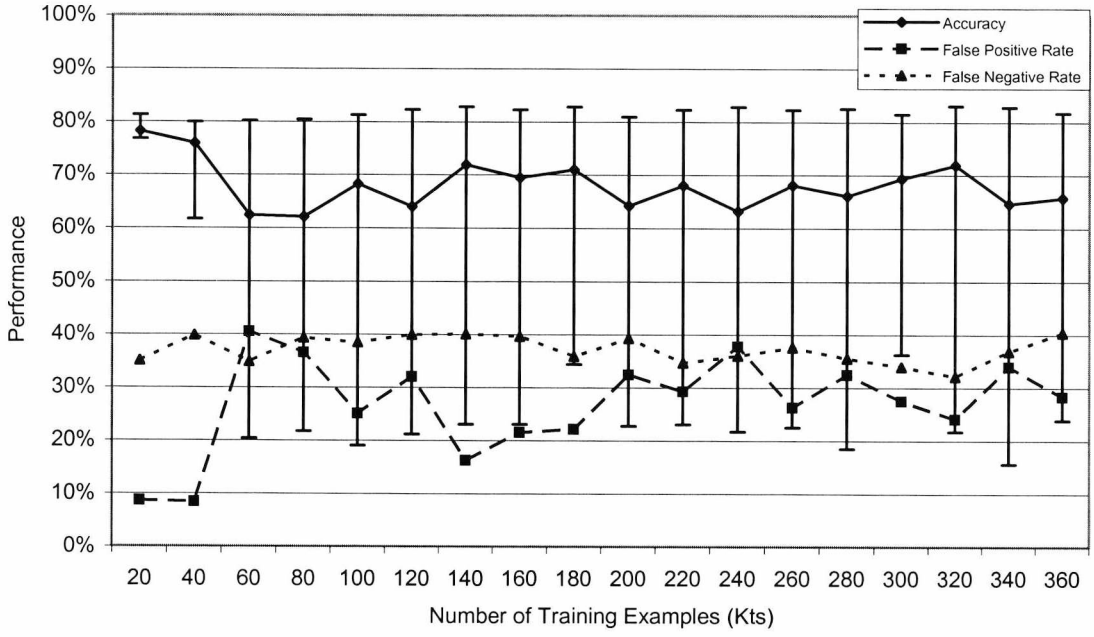

Figure 4.13. Influence of the Number of Initialisation Examples (Kts)

Combined with the results presented in section 4.7.6.7 and Figure 4.12, these suggest that the result of the algorithm can be hindered by overtraining. In Figure 4.13 it is possible to see how the positive class becomes over-represented in the population, causing dramatic variation in the false positive rate. The error in accuracy is also highly influenced by `Kts`, as it can be seen that for a value greater than 40 (or about 10% of the running set) this error becomes very large indeed, at some points having around 60% between the minimum and maximum values. It could be argued from this evidence that the initialisation stage is simplistic and requires improvement. This error can be attributed to overtraining the algorithm. However, AISEC is a continuous learning algorithm and is not meant to be trained in the traditional way. It has always been envisaged that the initialisation stage is not a training stage in a traditional sense. It therefore has no population control and no overtraining-avoidance mechanisms. The initialisation stage is used simply to generate an initial set of cells. From then on the continuous learning mechanisms can be used to prevent the algorithm from overtraining, as AISEC was designed to continuously learn. It was not thought that this undesirable performance would be an issue, but the results of this investigation show

that this is a problem that may need to be addressed. The size of the initialisation set compared to the running set may explain the increasing false positive rate as 360 e-mails are used to generate the initial population compared with 189 e-mails in the running set. This is opposed to the way the algorithm is designed for use with relatively few initial e-mails compared with the number of examples in the running set and this can be seen in Figure 4.13, as when an initialisation set of just 10% of the size of the running set is used the error in the accuracy is kept fairly low.

4.8 Using the Parameter Analysis

In the previous section an analysis of AISEC's parameters was shown. Now this has been done, it is possible to perform one final investigation to determine if using the information gained during that investigation can increase the accuracy of AISEC compared with the default parameter values in the sensitivity analysis as defined in Table 4.5. Using the results in the previous section the value of each parameter can be chosen as follows:

- The value of `Kc` is chosen to be 0.16 as according to the chart this exhibits high accuracy and gives a good trade-off with the false positive and negative rates.
- `Ka` is chosen as 0.24. This gives the best trade-off between accuracy, which fairly constant from 0.24 to 1, and FPR. With e-mail classification it is preferable for a lower false positive rate, therefore a lower `Ka` is preferable as it reduces the false positive rate at the expense of the false negative rate – an acceptable situation. This value is very different to the values of Ka in the previous tests.
- The accuracy stays roughly constant for all values of `Kl`, so a value of `Kl` = 25 is chosen, as it is the centre of the legal range. Based on the evidence, the value of this parameter is not critical.
- Similarly, `Km` is chosen as 0.5. The value of `Km` does not appear to affect the mean accuracy, FPR or FNR. Therefore a value near the centre of the valid range is a reasonable choice.
- The value of `Ksb` appears to have little impact and so the default value of 125 is kept.
- Similarly, the value of `Ksm` is kept at 25.

- An increasing `Kt` value increases the accuracy, but at the expense of the FPR. Therefore `Kt` = 15 is chosen to maximise the benefit of this trade-off, although any vale in the range 10 to 20 would be sensible based on the chart.
- Finally, with regard to `Kts`, the chart clearly shows that a value of `Kts` = 20 exhibits the highest accuracy with almost the lowest FPR. Most notable is the small deviation about the mean for this accuracy value. Higher values increase both the variation around the mean accuracy and the false positive rate dramatically, therefore this value is the only reasonable choice.

4.8.1 Assessment of Parameter Optimisation

With regard to Table 4.6, Parameter set A is that described in Table 4.5 and thus formed the basis of the investigation. Parameter set B is as described in the section above and therefore has been arrived at by informed choice and so, intuitively, should perform better than set A. The actual values of these parameters are repeated in Table 4.6. The mean results of running AISEC 50 times using each parameter set are presented in Table 4.7, where the figures in brackets represent the standard deviation for the value.

Parameter	Set A	Set B
Kc	0.2	0.16
Ka	0.5	0.24
Kl	7.0	25.0
Km	0.7	0.5
Ksb	125	125
Ksm	25	25
Kt	20	15
Kts	25	20

Table 4.6. Optimised parameter set

Parameter Set	Classification Accuracy	Recall	Precision
A	77.83% (2.27)	63.84% (4.84)	88.61% (0.96)
B	79.73% (2.66)	75.78% (3.67)	82.84% (4.09)

Table 4.7. Result of tests using optimised parameters

The optimisations can be seen to have increased the mean predictive accuracy over this test set. This is a positive result as it is entirely possible that attribute interaction could have affected the result in a negative way. There is no reason why setting all attributes to their optimum value individually would necessarily have an overall positive effect, especially when each attribute was optimised but all others were set to reasonable (but still arbitrary) figures.

The increase in accuracy from parameter set A to parameter set B can be tested for statistical significance. The null hypothesis, that the difference between the means is not statistically significant, can be stated. As more than 30 observations are used, the most appropriate test is that using the two-tailed large sample test based on the normal distribution (Appendix C). The working is shown in Equation 4.5.

$$z = \frac{79.73 - 77.83}{\sqrt{\frac{2.66^2}{50} + \frac{2.27^2}{50}}} \qquad 4.5$$
$$z = 3.03$$

A lookup table shows that the probability of the null hypothesis (both means are drawn from the same distribution) is less than 0.01. Thus the increase in accuracy can be seen to be highly significant.

However, while the accuracy has increased, the expected attribute interaction manifests itself in the recall and precision figures for parameter set B when compared with the other parameter set. The precision over parameter set B has decreased compared to set A but the recall has increased significantly. Thus, AISEC based on set B retrieves more positive elements from the running set, but those that it does retrieve it is more likely to classify incorrectly. This result is contrary to one of the aims of the optimisation as it is imperative that correct classifications occur. This is a situation therefore that couldn't entirely be predicted by the informed choice of parameters and therefore evidence that attribute interaction will still play a part.

Finally, a cautionary note. Extrapolating this result to conclude that this parameter set would work well on any other unseen data would be unwise. In this scenario, all of the running data has been used to effectively tune these parameters in this subsection. It is usual, in data mining, to use a training set, then a validation set for the tuning of parameters. Only after this validation set has been used can the algorithm run on a test set to ascertain the algorithm's accuracy. That is, the test set is only seen once by the algorithm. It is possible therefore that the parameter set developed here suffers from overfitting (a lack of generality) to the dataset used.

4.9 Investigating Effect of Class Distribution

The proportions of interesting to non-interesting e-mail in the running set were varied to determine how the relative class distributions in the unknown-class data affect the performance. This was only possible using the dataset of (Jang, 2004) as it contained an

even split between the two classes in the running set. With the number of uninteresting messages set at the standard 188, the number of interesting messages in the running set was varied from 10 to 188. Then, with the number of interesting messages set at 188, the number of uninteresting messages was varied from 188 to 10. Tests were run 50 times for each value and the mean accuracy taken. All parameters were set to those defined in Table 4.6 set A comparable with the results shown previously. The resultant chart is shown in Figure 4.14.

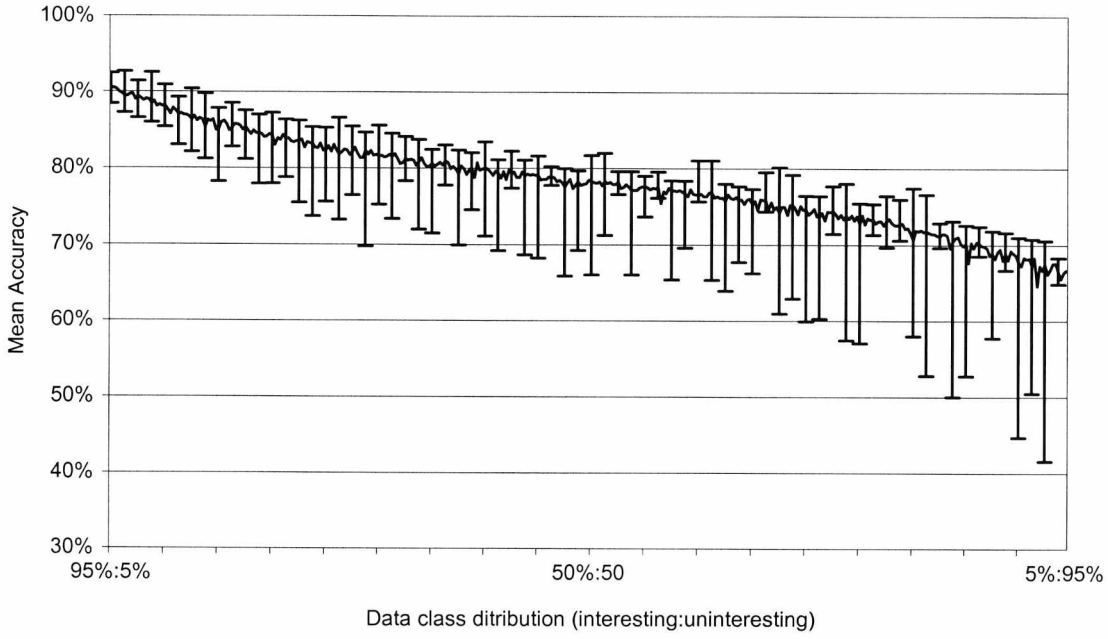

Figure 4.14. Change in predictive accuracy with changing class distribution

It can be seen that, as the amount of uninteresting e-mail increases with respect to interesting e-mail, the mean predictive accuracy decreases. In addition to this, the variation around the mean also becomes quite large. It is thought that this is a side effect of cells only representing one data class (uninteresting). Compounding this, it is possible that with increased uninteresting e-mail comes increased diversity and it may be harder for the algorithm to represent this. As such, this type of behaviour is inherent in AISEC. This investigation therefore may lend weight to the suggestion that both data classes should be represented in the cell population and as such further investigation into this is recommended for the future as it is possible that the accuracy of AISEC may be helped greatly by this change. In the meantime, it might be possible to exploit this bias. If it is found that a user is getting more uninteresting e-mail than interesting e-mail, it could be possible for AISEC to use B-cells to store the interesting e-mail rather than the uninteresting, therefore exploiting the effect seen.

4.9.1 Summary

The properties of dynamics and diversity by which e-mail classification is characterised have posed great challenges for effective filtering e-mail, promoting research on various approaches. Inspired by clonal selection, the dynamics inherent in AIS algorithms such as AISEC, are believed to be powerful enough to make AIS a successful solution to various problem domains that require a highly adaptive system. It has been shown that an immune inspired algorithm written with text mining as its primary goal may yield a classification accuracy comparable to a Bayesian approach in this continuous learning scenario. This chapter presented a time complexity analysis of AISEC and described an investigation into the dynamics of AISEC with hypotheses and experimental evaluations on parameters and their influences on performance. According to the results of the investigation, the most influential parameters are the classification threshold (`Kc`) and the affinity threshold (`Ka`). The effects of class distribution were also investigated and the class distribution of the data was found to affect the result.

It should be stressed that throughout this section only one dataset has been used for each test. While each test was run a number of times to negate the effects of random variation, as only one data set was used it is acknowledged that the generality of these results is bound to be impacted. All conclusions were drawn with this in mind but further investigations using more datasets are highly recommended as this will allow for the generality of the AISEC technique to be assessed.

Thought has been given to ways in which AISEC may be improved and it is acknowledged that for AISEC to develop as a classification tool, certain additions could be made for example use of body text from the e-mail to improve accuracy. Improved text processing may be attempted, recognising the difference between "breast" and "breast cancer" or recognising the characters M-O-N-E-Y as "money". However, it is believed that AISEC in its present form demonstrated that an AIS can be used for data mining in a continuous scenario, and so from this point of view, and therefore the point of view of this thesis, this has been successful.

The following chapter continues to further this subject area by creating an artificial immune system for web mining. It will be shown that a number of paradigms used are based on those developed for AISEC thus allowing informed choices to be made in the future, and as such the development of AISEC has been invaluable to this thesis.

Chapter 5 AISIID – an Artificial Immune System for Interesting Information Discovery

5.1 Introduction

In the previous chapter the algorithm AISEC was described, tested and evaluated. The results of the tests showed that a clonal selection based AIS algorithm could perform classification of e-mail with an accuracy similar to that of a naïve Bayesian algorithm. In addition, AISEC was shown to work well at a continuous classification task. Based on these positive steps, this chapter is concerned with a study into turning AIS towards web content mining and details a system to achieve this called AISIID (Artificial Immune System for Interesting Information Discovery). The task of AISIID is to discover interesting information on the web: its task is to discover web pages that are relevant while being novel, surprising or unexpected (as defined in Chapter 2). As such it is very much a problem orientated system, an approach to system building advocated in (Freitas & Timmis, 2003).

As the number of web pages and other information on the web has grown, so has the study of techniques for mining and manipulating this information. The literature describes a vast array of systems for web content mining but one of particular interest is (Liu et al., 2001), as it has inspired the development of AISIID to mine interesting information from the web. As the scale of the web grows, increasingly adaptive systems must be realized to keep pace with the accelerating change in web-based information. With such an overwhelming quantity of data available, users may suffer from information overload. Filters and search tools are a must for almost any web user and in the future it is likely that these tools will endeavour to become more intelligent. At present a simple keyword search on a search engine may yield far more pages than a user could ever cope with. In addition to this, it is hard to search for unexpected information as, by its very nature, it is unexpected and therefore simple search

keywords, as employed by most widespread search engines, cannot be specified by the user. A more intelligent, user-driven approach is needed such that pages a user would consider unexpected, novel, or surprising are returned from a search.

For this task to be achieved, a particular webpage must contain information which is unknown or unexpected to the user. AISIID should be able to rank discovered pages in order of interestingness before presenting them to the user. AISIID's ability to identify interesting information is based on the following hypothesis:

Documents containing a high frequency of words semantically related (synonym, antonym, hyponym or hypernym) to the typical words contained in a set of user specified documents will be of greater interest to the user than a set of documents ranked using relevance alone.

To the author's knowledge this is the first AIS to tackle such a web mining task. It will also be only the second system produced to address the problem of identification of "interesting" web pages in the sense above (the first being that of (Liu et al., 2001) as examined in Chapter 2), and the first to do this in an adaptable manner.

5.2 Motivation

Traditional keyword-based search techniques, both on the internet and otherwise, contain an inherent problem: the results obtained are likely to supply the user with mainly information that the user already knows. A keyword-based search only searches for pages that are relevant based on the keywords specified. While this is perfectly acceptable for many web searches, a lack of mechanisms exist that provide an interested user with a complement operation, that is, finding unexpected information: information that user was *not* specifically looking for.

As the goal of AISIID is to search for information which would *not* typically be returned when using traditional information retrieval methods, the use of web search engines to provide the raw data is ineffective. While search engines such as Google do indeed index a significant proportion of the web, the actual retrieval of these pages is still constrained by the input and search restraints of these search engines. That is, only pages containing *all* of the submitted keywords will be retrieved. This situation is inflexible and as such acted as a motivation for the creation of an adaptive webpage spidering technique. Given that the AISIID system will know what topics the user is searching for, strategies can be employed to only follow hyperlinks, and therefore retrieve webpages, the AISIID system estimates will be relevant to the user-specified topic. To summarise, AISIID aims at discovering interesting web pages that are

unexpected, contain novel information and/or surprising, whilst still relevant to the user's search using a sophisticated and adaptive AIS. Whereas, conventional search engines such as Google retrieve only relevant web pages using just a simple keyword-based mechanism.

5.2.1 Why Use Immune Inspiration for Web Mining?

In the previous chapter a number of reasons why an AIS is particularly suitable for e-mail classification were given. To expand on these the immune system is particularly suitable inspiration for a web content mining algorithm because of certain properties inherent in many immune inspired algorithms. Work in (de Castro & Timmis, 2002a) describes these properties which are in turn based on the work in (Dasgupta, 1999). Of those cited, many parallel the desirable features of a web mining algorithm such as AISIID. Examples of these, with explanations, include:

1. **Pattern recognition**: The ability to recognize patterns of data similar to training examples is a common characteristic found in classification tools and of use in the web mining domain. This is an important feature in such a web mining scenario where it is the task of the system to learn patterns of user interest or knowledge.
2. **Diversity**: Like the immune system, the web is diverse. It carries many different information formats, from plain text to fully animated web pages. The immune system too contains a huge number of different cells each with its own specialised function and is capable of recognising a very large number of different types of antigen. These metaphors could be extracted to produce a system in which different types of cell support different types of data and therefore the ability to identify information contained in these diverse media is a great advantage.
3. **Distributivity**: The advantages of distributing a system over many systems afford not just fault tolerance but also for the possibility of parallel processing and thus reduced processing time. In a web mining system where processing power and storage are under great demand this is of great advantage. Few large-scale web mining systems are not distributed so the precedent has been set to aim for straightforward distributivity of such a web mining system. Distribution of this algorithm is briefly discussed in Chapter 7.
4. **Self-organization**: A well designed AIS may automatically change to suit changing underlying data. In a web-mining system this is of paramount

importance as the web is extremely dynamic. The content of pages is constantly changing and so too are the links between them. Thus, to pre-program set behaviours for the system would be a time consuming task thus the system must self-organise to keep track of this changing domain.

5. **Noise tolerance** – The ease with which anyone may publish to the web can raise questions regarding its quality. Errors and omissions are common. The immune system however is noise tolerant, such that absolute matching is not required to trigger a response. Due to the non-specific affinity function at the heart of many AIS algorithms combined with a population of cells, a number of which may match different aspects of a single example, an AIS has the potential to filter noisy data and uncover an underlying concept. Such noise tolerance is essential to an algorithm mining low quality data and the learning characteristics of the immune system are invaluable in this case. At a higher level, the topology and content of the web is always changing. The ability to adapt to these changes can be an important feature of a web mining system. New computers and data can be added or removed from the internet easily, likewise immune cells are constantly undergoing cell death and reproduction. The ability for both to cope with this dynamic situation is important. AIS have shown to be adaptive, resilient and robust and so are suited to this domain.

Of these, the central characteristics for this investigation are adaptability, noise tolerance and, in future work, distributivity. As has been stated before, noise tolerance is a characteristic of AIS algorithms and is an important concept in the task performed by AISIID as web pages contain great amounts of noise. Given an AIS, when numerous cells recognise a page, the confidence that the patterns contained on that page are indeed correctly recognised is increased as the population is diverse. Therefore, the algorithm does not recognise a page based on single attributes but rather the combination of multiple attributes. From this follows the usefulness of diversity in such a situation. The use of a diverse set of cells as is common in an AIS not only confers noise tolerance to the algorithm, but also encourages adaptability.

Distributivity is an important aspect of such a web mining algorithm. The size of the web mining data source (the web), is vast and anything other than the smallest web mining systems require storage and processing capacity far in excess of that of even a high-powered single machine. This is the case especially for the task performed by AISIID where the processing time for each page is expected to be many times higher

than keyword based algorithms. It is therefore quite normal to span web mining systems over numerous separate hosts. Most web mining algorithms are specifically designed with this in mind, but due to its population based nature an AIS lends itself naturally to distribution. While distribution of such a system will always be challenging, by their nature AIS do lend themselves to distribution (Watkins & Timmis, 2004). Distribution of a web mining system will counter the problems of scaling that systems will often encounter when confronted with datasets the size of the web.

5.3 Overview of AISIID

AISIID is an Artificial Immune System for the collection and ranking of web pages judged to be interesting to the user. AISIID uses a population of immune cells to both search for and rank web pages. Both these actions are intertwined and almost inseparable. The AISIID process is summarised as follows.

AISIID uses a population of immune cells and processes inspired by clonal selection to discover interesting web pages. The user specifies a small collection of web pages that summarise his or her knowledge on the search subject. Starting on one of the user specified pages, each cell is given a position on the web and is free to move, following hyperlinks that may lead it to other interesting web pages. Each web page it encounters is regarded as an antigen and is therefore available for an affinity evaluation. Interesting webpages have high affinity with the cell and the cell is stimulated based on this affinity. Stimulation above a threshold will cause the cell to clone and mutate. While cells with low stimulation levels will be removed from the population, the clones and parent cell can then move to another webpage by following a hyperlink from the current page. When a stopping criterion is met, the pages that have been found during that run are ranked according to the mean affinity each had with all immune cells that found it during the run of the algorithm.

Many aspects of AISIID are uncommon among web mining systems, with several innovative aspects including:

- Use of semantic word transformation to allow a search for web pages containing not only relevant but also previously unknown keywords.
    - Evolved adaptation of these word transformations to increase search diversity through the use of WordNet (Fellbaum, 1998).
- The use of an automated "user feedback" mechanism (akin to the generation of a co-stimulation signal in AISEC, Chapter 4, and the immunology as outlined in Chapter 3).

- The use of the search engine Google to determine weights of features (words) where the weight is the Term Frequency Inverse Document Frequency (TFIDF) of that feature. Thus the entire web, as indexed by Google, constitutes the collection from which the feature is drawn.

Some other interesting aspects include:

- Use of the text surrounding hyperlinks to guide spidering.
- Local level population control is used to stabilise the global population. That is, cell death rates are mediated by cells immediately surrounding that cell.
- Returning of interesting words to user to aid comprehension.

When judging the affinity (quality or interestingness) of a webpage, a cell must make a calculation based on two metrics. That is:

- The relevance of a page.
- The novelty, unexpectedness or surprisingness of a page.

Both of these factors working together is important as, while a page may well contain vast amounts of information unknown (surprising or unexpected, etc.) to the user, for that page to be interesting the information must be on a topic *relevant* to the search.

Recall from Chapter 2 how one desirable property of a data mining algorithm is an explanation of the decisions made. AISIID is not a "black box". AISIID allows the user to query why a page was determined to be interesting by examining the "interesting words" used to find that page. Therefore, a user may be presented with a set of the most interesting words regarding the search, further contributing to that user's knowledge and understanding. Furthermore these words may be used in a traditional keyword web search environment to enable the user to find more information and further contribute to the user's understanding.

5.3.1 AISIID Compared With Traditional Web Search

AISIID is concerned with a different problem to that solved by traditional search engines such as Google (Google, 2006b) and Yahoo (Yahoo! 2006) etc. These take a very small amount of information specified by a user, typically just a few keywords, and aim to retrieve a set of relevant results (documents, web pages or other types of media) in the shortest time possible, with the ordering of the results typically biased by

the estimated authority of the site on which the item is located. This results in the retrieval of a large number of documents, typically thousands or more, where the actual interest (where interest is novelty, surprisingness etc.) to the user may be low. However, users receive the results quickly and the results retrieved have been extracted from a very large proportion of the web.

AISIID, on the other hand, takes a very different approach. AISIID will search a small proportion of the web (although this will typically still contain thousands of web pages) and aim to present the user with a small number of "highly interesting" web pages. The drawback is that the user must sacrifice speed for overall quality of search result. AISIID uses much more initial information, typically a number of webpages. These contain thousands of words, rather than a few keywords. To take advantage of this increased information the resulting algorithm processes pages in much more detail, but the time taken is increased greatly in comparison with systems such as conventional search engines. Specifying search criteria in this way also allows a solution to the problem that users cannot specify unexpected search terms, as enough information is present to allow the automated estimation of unexpected concepts.

AISIID is not an interactive search mechanism. The timescale of an AISIID search is not short enough to allow a user to perform a typical cycle of search adaptation and resubmission. It should be noted therefore that in a real world scenario AISIID may not be suitable for a user requiring an immediate answer to a question with limited bandwidth. Rather it is more suited to a situation where the user is able to leave a system running for many hours and would greatly benefit the user if it were to discover information that cannot be revealed using traditional techniques.

5.3.2 Some Notes on AISIID, AISEC and Biological Metaphors

AISIID has a very different concept of environment compared with AISEC, and even though AISIID is based on some processes or strategies found in AISEC the differences should be made explicit. In the AISEC system a single e-mail was processed into an antigen and then presented to all artificial cells in the system. The closest biological analogy to this could be that the system represents a single lymph node and during a particular point in time an APC presents a single antigen to all cells within that lymph node. In contrast to this the cells in AISIID have a notion of location. They come into being at one particular location (on a particular webpage) and are then allowed to move through the web following hyperlinks from page to page. If they are stimulated in some way by the contents of that page then they will react. This is more akin to a B-cell

moving through the body, the antibodies on its surface ready to bind with an antigenic pattern at any time. AISEC creates a scenario where every B-cell is forced to evaluate each new piece of data. AISIID on the other hand relies on cells seeking out new antigenic patterns then assessing their affinity with the pattern when they find it. AISIID is therefore a rare paradigm in the AIS literature. In the domain of AIS it is common for each and every artificial cell in the system to be forcibly compared with an unknown antigenic pattern. It is clear this is only a metaphor with the natural immune system. Also, the immune system of a host will be receiving numerous stimulations at any point in time. It is obvious that a single natural cell at one single location is not compared with all stimuli from all locations in the host. Rather, in the natural world a cell is located at one point in the host and therefore by its nature must act at a local level. This is mirrored in AISIID in that a cell is located at a point in space and at a single time step cannot interact with the environment other than its immediate place within that environment.

While AISIID is to some extent an extension to AISEC in that the knowledge gained during development of AISEC has greatly helped the development of AISIID, there are additional significant differences between AISEC and AISDIID that should be made explicit. Making these open here can prevent confusion and may help understand the motivations for building AISIID:

- Unlike AISEC, which works as a *passive* filter, AISIID is an *active* discoverer of information.
- Unlike AISEC, AISIID uses only a set of naïve cells. No memory cells are generated during the running stage of the algorithm.
- While in AISEC the relevant words adapt over time, in AISIID the relevant words are both shared by all cells and fixed during the lifetime of the algorithm. Instead, the transformations performed on these words change during the algorithm lifetime.

Of note is that fact that, when searching for web pages as AISIID does, the notion of strict classification should be relaxed. Whereas AISIID made a binary decision that an e-mail is either interesting or not, this is to be relaxed in the following two chapters and a ranking scheme is introduced. This is a sensible tactic and is mirrored in the way results are presented by all major search engines. That is for, all pages found to contain the keywords are returned but ranked in order of relevance for the user. Thus a certain

amount of classification has taken place, all members of what could be seen to be the positive class contain all words in the query.

5.4 Algorithm Description

This section describes the algorithm in detail. The chapter is structured to draw attention to the highlights of AISIID and give a broad overview of its working. The more formal pseudocode then follows allowing a lower level description of the algorithm to aid implementation. This chapter again follows the layered AIS framework of (de Castro & Timmis, 2002a) i.e. encoding, affinity measure and algorithms and processes, as encountered in Chapter 3.

5.4.1 General Notes

5.4.1.1 Two Signal Approach

Work in Chapter 4 demonstrated that the use of a two-signal approach when applied to continuous classification algorithms appeared beneficial, therefore this has been incorporated into AISIID. However, unlike the mechanism of AISEC where it was reasonable for a user to give feedback, it would be unreasonable for a user to give feedback in a similar manner when performing a search. A user would not be able to provide feedback quick enough for a search to progress. Such feedback would be obtrusive, a practice generally discouraged (Chan, 1999; Chen et al., 2001).

AISIID uses a confirmation signal mechanism similar to that in AISEC, but as the user is not in the loop this confirmation signal must be given automatically. When a cell moves from one page to another it has made an implicit judgement based on relevance about where to go. This judgement is expressed by an estimated value of the degree of interestingness of the page where the cell will move to. A high estimated value of interestingness can be considered analogous to signal one. The cell can then measure the actual interestingness of (and affinity with) the new page where it moved to, considering the entire text of that new page. If the estimate and the actual value are numerically close (section 5.4.6.4) then the cell has made a correct decision, signal two (a confirmation signal) occurs and the cell will be rewarded. The mechanism by which these judgements are made is described later. If the estimate and actual value differ greatly then the cell must be penalised. This processes is referred to as "automatic feedback" rather than AISEC's "user feedback".

5.4.1.2 Robust Spidering

As mentioned previously, part of AISIID's behaviour is that of a web spider. However AISIID attempts to minimise wasted computational time and network bandwidth by only retrieving information on the user-specified topic, and is therefore an instance of a guided spider. Whilst cells in AISIID are punished for retrieving information that is not interesting, it is quite possible that there may be a situation where a cell must make its way through a number of non-interesting pages before it reaches interesting ones again. Recall from Chapter 2, with regard to guided spidering (Wu & Hsu, 2005) state that "*some sets of off-topic documents often lead to highly relevant documents*" thus "*an optimally focused crawler should sacrifice visiting several off topic pages in order to reach the highly relevant pages among the hyperlinks*". Cells should not therefore be punished immediately for finding this uninteresting information but should be allowed to continue for a certain amount of time before being removed. Therefore only consistent uselessness results in cells being removed from the AISIID algorithm. Good cells will be highly stimulated and as such these good cells will be able to move through more uninteresting pages compared with bad cells, which will tend to have lower stimulation, before being removed due to low stimulation level. The cells in AISIID are allowed to make a small number of incorrect estimates, unlike AISEC where the cell is removed from the population immediately. It was noted at the time that the removal strategy was rather draconian and as such has been changed for this situation.

This promotes robustness in the spider, one of the advantages of using an AIS type algorithm. One characteristic of the web, as it stands today, is that content is often separated from navigation. That is, pages that contain large amounts of text and therefore potentially large amounts of interesting information may contain few outgoing links, whereas pages containing high numbers of links are likely to be navigation pages, that is, pages devoid of content whose sole purpose is to provide a clear and easy means to navigate to other part of a web site or other pages on the web. This is an effect of the web being produced for people rather than automated information retrieval. AISIID takes this into account as it allows a cell, stimulated highly by a content page, to make a number of moves to pages that are not interesting, which could include navigation pages (generally considered uninteresting as they lack content), before the cell's stimulation will drop too low for it to continue. This therefore deals with the situation outlined above and situations similar to it reducing sensitivity to noise in the spidering stage.

5.4.1.3 Webpage Pre-processing

Each webpage will be processed in the same way. A webpage is pre-processed by first stripping all HTML tags from the webpage, but marking the position and target of all hyperlinks. These hyperlinks are then separated from the text and stored with a reference to their original position leaving plain text only. All punctuation is removed and replaced with space characters. The remaining text is tokenised, with each token delimited by a space character. Any tokens containing only numbers are removed. Tokens containing both numbers and letters are left as this will preserve strings such as "1st". All remaining tokens are transformed to lower case and finally stopword removal is performed over the set of tokens. The list of stopwords used is shown in Appendix A. In summary, each webpage is represented as an ordered list of lower-case words, not including stopwords with the positions and targets of hyperlinks marked.

5.4.2 Flowchart of AISIID Algorithm

A diagrammatic depiction of the algorithm flow is shown in Figure 5.1. The following explanation can be quite complex and it is hoped that this flowchart is useful as a reference to show where each section of the following explanation fits in the overall algorithm.

5.4.3 Representation

Each artificial immune cell will encode:

1. A summary of the user's interest
2. An estimate of what the user will find interesting
3. A location on the web (a URL)
4. A count relating to stimulation

In (1) the user's knowledge must be summarised. The cell must encode this so as to be able to determine the *relevance* of any antigens (webpages). It is assumed a user's prior knowledge and interest will not change during a single run of the system, a reasonable assumption. The summary of the user's knowledge comprises of a vector of words. This vector carries a set of words relevant to the user's search and is therefore referred to as the *Relevant Words Vector* (RWV).

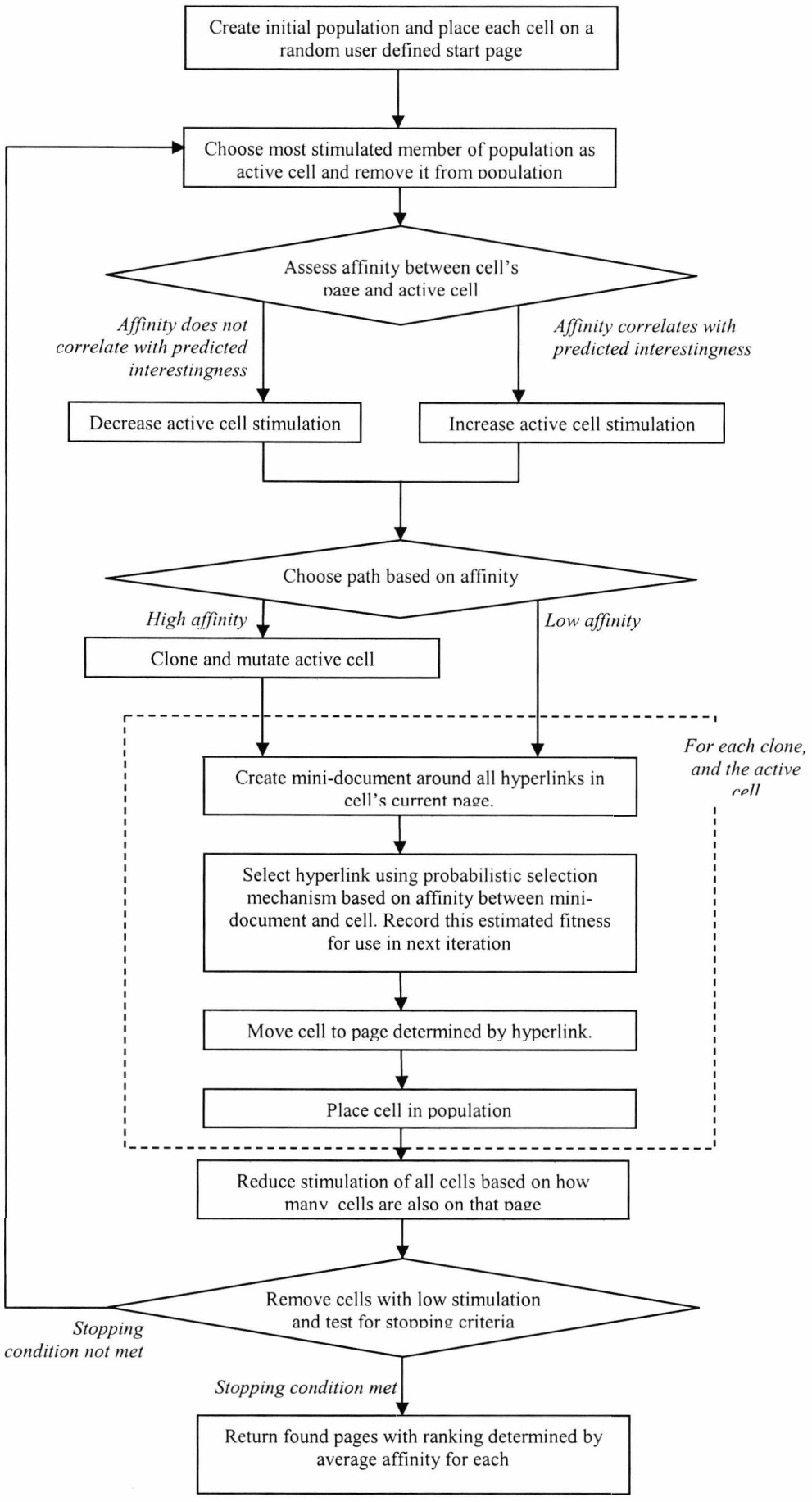

Figure 5.1. AISIID system flowchart

This vector is not variable in size and will carry the *n* most important words as ranked out of all words found in the training documents (where *n* is a user defined parameter). The mechanism by which words are ranked is described in its own section, 5.4.5.

The set of attributes describing a cell (2) does not contain a list of interesting words, but rather a list of transformations that may be used by WordNet to create a set of interesting words (novel, surprising etc.) from the RWV. This vector is therefore referred to as the *Interesting Transformation Vector* (ITV). This vector is the same length as the RWV, with each position containing exactly one transformation that is legal to apply to the corresponding element of the RWV. These transformations form the adaptable part of the immune cell and so, in contrast to the RWV, will change. For simplicity, this vector is implemented as a vector of numbers, where each represents a legal transformation on the word in the corresponding position in the RWV, see Figure 5.2.

```
RWV = <fish, chips, peas, sauce>
ITV = <1,3,0,0>
```

Figure 5.2. Example RWV and corresponding ITV

The goal of the AIS is to change the elements of the ITV to find the most interesting pages for the user, this is guided by the evaluation function to be described later. Concerning cell attribute (3), unlike AISEC, AISIID's cells do not exist simply within the feature space on the running algorithm but also occupy a position on the web. This current position is simply stored as a URL.

Finally each cell carries a real number representing a level of stimulation for that cell (4). Cells with low stimulation are removed from the population. The details of how the variation in stimulation is calculated are made explicit in section 5.4.6.4.

5.4.3.1 Using WordNet Relationships to Generate Interesting Words

It is relatively straightforward to determine a set of relevant words from a document. The text mining and information retrieval community has been tackling just such an issue for decades, for example it is believed that the first use of selecting keywords by some weighting method was in 1976 (Lancaster, 1976). However, the aim of AISIID is to find interesting documents. It is therefore important to employ a strategy for determining interesting webpages among the large number of pages encountered rather than just relevant pages. Recall the statement from Chapter 3: *it is believed that if a*

document provides users with information that is novel, unexpected or contradictory to his or her beliefs and still relevant to the user defined topic of the search, that user will find the document interesting.

AISIID's interesting page discovery strategy is therefore based on the following hypothesis:

Given a word that is ranked highly relevant in a document, words produced by semantic transformations on that word will be interesting.

To expand this hypothesis, in Chapter 2 it was stated that interesting information is that which is not only relevant but also (1) novel or (2) unexpected or (3) surprising in the sense of being contradictory. Given an initial set of (indirectly) user-specified relevant words in the RWV it is possible to generate words that satisfy both criteria using WordNet and employing the synonym, hypernym (generalisation), hyponym (specialisation) or antonym relationships. Taking (3), generating words contradictory to user's expectation can be done using the *antonym* relationship in WordNet. After all, the RWV will almost by definition contain expected information. Generating words that mean the opposite of these may be interesting because identification of these words in a document may point to information contradictory to a user's expectation. An example of this may be, when a search topic is sports cars, the word "fast" is likely to be in the RWV. An antonym of the word "fast" is the word "slow", and if the word "slow" is discovered to be prevalent in an article about sports cars then that article may contain some surprising, and therefore potentially interesting, information.

The remaining three relationships will generate words to satisfy (1) and (2). Each of these relationships will generate words that are related to a "seed" relevant word but contain slightly different meanings and each useful to the search. Generated *hypernyms* will produce generalisations of the seed word, opening the search up to documents on more generic topics, it is hoped, leading the search and ultimately the user to related topics which to the user may be unknown. In contrast, the *hyponyms* generated may guide the search and return to the user novel information in a similar way, except this time a specialisation of the seed word is performed. This can be very useful as it may automatically fill in gaps in the user's knowledge. As an example, the phrase "sports car" is a hyponym of "car". It is possible that in a search about cars, the user may also wish to know about sports cars. The application of this relationship would therefore find this sort of information automatically.

Method

The actual process used to create a set of words from the ITV using WordNet is straightforward. The Java WordNet Library (JWNL) is used to create an interface between AISIID and WordNet. JWNL is freely available from the Sourceforge website (Didion, 2005) and is released under the BSD licence. JWNL version 1.3 (release candidate 3) is used for this work.

Given a word w at position i in the RWV, the corresponding operation identifier o at position i in the ITV is retrieved. Using JWNL the set of words that are returned when the operation o is applied to w is determined. For instance, if operation o is "antonym" the set of antonyms for word w are retrieved. This must be done for all parts of speech in which the word legally exists, although this check is easily performed in WordNet. All operations of this nature will return a set of words called a synset. It should be made explicit that WordNet maps synsets to synsets, not words to other words. This is a legitimate thing to do as a synset contains a set of words all with a similar meaning. Therefore it stands to reason that the same operation on any of those words will result in the same resultant set of words as a consequence.

For the synonym and antonym relationship, a single synset maps to another single synset. For the hierarchical operations: hypernym and hyponym, one single synonym set is not enough. Instead, the hierarchy is followed with a synonym returned from each level of the hypo/hypernym hierarchy within a given number of levels of w.

WordNet does not simply contain words but also phrases. Examples of this may be the specialisation of the word "lorry" to "big lorry" or the colour "green" to "light-green". AISIID takes account of this, and when performing matching on a phrase, will look for one word immediately followed by the other (and so on if the phrase is greater then two words in length). This is significantly more costly than identifying whether a word is present, as the latter operation can be done very efficiently by hashing. Retrieving a single word is therefore of time complexity $O(\log\ n)$ while retrieving a phrase can be $O(n \times (n-1))$ where n is the number of words in the document and l is the length of the phrase. However, in empirical tests it was observed that a significant proportion of the words generated by WordNet are, in fact, compounds of more than one word and as such they cannot be ignored and must be treated appropriately.

5.4.4 Affinity Function

The affinity of a cell with a webpage is calculated using the words found in the cell's RWV and words generated by the cell's ITV. The affinity calculation begins by

generating the set of interesting words from the cell's RWV using the transformations as defined by the sequence of steps in section 5.4.3.1 (in practice these words may be cached to increase efficiency). The webpage is processed into the form of an antigen as described in 5.4.1.3 to give a set of words. The number of words generated by WordNet using the transformations from the ITV that also appear on the webpage is counted and then normalised by the length of the number of words generated by the WordNet process. Similarly the RWV is compared against the webpage and the count of the number of words present in both the webpage and the RWV is normalised by the length of the RWV. To calculate the affinity, the mean of these two scores is taken (and therefore the value of interestingness and relevance are weighted equally in the affinity function). The result is the affinity between the antigen and the immune cell and by definition will return a real number in the range [0,1]. The method is shown in Equation 5.3.

As an aside it is important to give a reason for the choice of such a simplistic matching technique when other, possibly more accurate, strategies are available. It is possible to reason that the computation of legal TFIDF values is not practically feasible in this situation. This reasoning is as follows.

For a cosine distance calculation to be performed, a TFIDF weight for every element of the RWV vector and every feature (word) in a web page must be computed. The efficient and simplistic strategy of gathering the TF count over the RWV would be nonsense as the RWV is simply a collection of words with each occurring *only once* in the vector, no matter the relative importance of the words contained in the original user specified documents. The IDF would also be impossible to accurately compute. So it could be a reasonable strategy that this value is stored from the initialisation process (which requires the TFIDF score of a word to be computed to allow words to be ranked). Given this measure each word in the web page would also need a TFIDF score associated with it, requiring the derivation of TF and IDF for that webpage feature where that webpage is in a set of documents. To compute the TFIDF weight for a webpage element a DF value must be determined. The DF calculated over the entire web (an example of which is included in 5.4.5) is the only fair option as this is the strategy used to weight the words in the RWV. It is important to compare like with like. However, this strategy is intractable to implement. Firstly, it is incredibly time consuming as a single Google query must be dispatched for every unique word on every web page. Furthermore, in practice the Google API also places constraints on the

maximum number of queries per day. For these reasons, the strategy as implemented is believed to be the most reasonable in the circumstances.

Similarly, a distance measure between the ITV and a web page must be defined. The TFIDF weight for each unique token in a web page could be computed as before However, determining a TFIDF score for words generated from the ITV (the collection of these words is called the Interesting Words Vector or IWV) is practically impossible. The TF count over the IWV is meaningless as, whilst the same word could be generated more than once from the ITV, the frequency with which each word is generated would bear no relation to proper linguistic use and therefore could not be used as a legitimate TF. The IDF for such a word is also unavailable as there is no document set from which the individual features are drawn. A calculation of IDF is therefore unavailable for the relevant word vector and interesting transformation vector words.

In the AISIID system, the relevance and interestingness of a webpage are considered separately as they are used differently in the algorithm, but are combined for the purposes of the affinity function. Taken separately, the relevance of a page is computed as shown in Equation 5.1.

$$\texttt{relevance} = \frac{\sum_{i=1}^{|RWV|} \delta_i}{|RWV|} \text{ where } \delta_i = \begin{cases} 1 \text{ if } RWV_i \in w \\ 0 \text{ otherwise} \end{cases} \quad \textbf{5.1}$$

Likewise, the interestingness calculation is shown in Equation 5.2. It should be noted that the ITV is not used directly. The words in the relevant word vector are combined with the transformations defined in the ITV to create an interesting words vector (IWV) and it is this latter vector of words which is compared with the webpage. Thus the vector's phenotype is used in this instance. Therefore relevance is a count of features found in both w and the IWV belonging to c normalised by the length of the RWV.

$$\texttt{interestingness} = \frac{\sum_{i=1}^{|IWV|} \delta_i}{|IWV|} \text{ where } \delta_i = \begin{cases} 1 \text{ if } IWV_i \in w \\ 0 \text{ otherwise} \end{cases} \quad \textbf{5.2}$$

$$\texttt{affinity} = \frac{1}{2} \times (\texttt{relevance} + \texttt{interestingness}) \quad \textbf{5.3}$$

Finally, the affinity between cell c and webpage w is calculated as shown in Equation 5.3. IWV is a vector of words generated by WordNet using the RWV and

ITV. Therefore the interestingness is count of features found in both w and the interesting words vector belonging to c, normalised by the size of the IWV. It should be noted that both the relevance and interestingness are weighted equally. It was beyond the scope of this investigation to perform any testing regarding differing weights for these. It would have been unreasonable to simply guess a weight and therefore both are equally weighted as a default position. More thoughts on the weighting of these attributes can be found in the conclusion, Chapter 7.

5.4.5 Processes

The following sections describe the main processes that, when combined, make up the AISIID algorithm.

5.4.5.1 Initialisation

The purpose of initialisation is to create an initial set of immune cells trained to recognise relevant web pages and place each on a suitable web page. The system is initialised using a set of user specified web pages. The importance of these pages cannot be overstated as they are used to summarise the user's prior "knowledge". Recall, pages are used as they can allow for the discovery of information that a user does not already know. A user specifies what he or she knows already, and the system tries to infer whet he or she doesn't know, this contrasts with regular searches where the user cannot specify keywords for concepts he or she does not know about.

One single page is not enough for this and so a small number of pages are ideally required from the user. Web pages tend to contain a certain amount of noise, whether this is from advertisements, navigation panels or simply a general mix of topics on one page. Initialising an algorithm on a single page would, therefore, leave a system prone to discover pages based on this noise. Using a number of pages has the potential to increase the probability that a system will be initialised on the correct topic. It is believed the features pertaining to a *common topic* will be reinforced (as these appear throughout the pages) and therefore is distinguishable from the noise, as the content of this noise is likely to differ between all initialisation pages. Therefore the more diverse (yet still "on topic") the set, the better the potential for good results.

5.4.5.2 Selecting Important Words from Initialisation Documents

One important decision is how the set of relevant words for the RWV should be chosen. These words are used to summarise the concept of what the user finds relevant, and

therefore the basis of what the user will find interesting; importantly this will not change during a single run of the system. It is, therefore, of great importance to choose these words correctly. Weighting the features of the initialisation webpages using a term frequency-inverse document frequency (TFIDF) scheme has been shown over many years to generally result in an accurate weighting.

Naively it would be possible to rank the words by word frequency and indeed this would be straightforward to implement. However, some words will naturally occur more than others and so while this is a possible solution, it is certainly not optimal. TFIDF ranking in this situation, however, comes with an inherent problem; *for a feature (word) to be weighted using TFIDF it must be drawn from a collection of documents where the elements of the collection not only contain documents of one "positive" class but also those of a "negative" class.*[3]

Thus, it is impossible to gauge the real quality of a feature from the positive class when it is only compared with the average over all features of that positive class. Rather, the feature must be compared with other features in both the positive and negative classes. The result of applying this TFIDF weighting to all features is that the weight associated with those features relevant to topics found throughout all initialisation pages is high, while the weight associated with features found on relatively few initialisation pages, and therefore likely to be noise, tends to be low. Given that during the initialisation stages the algorithm has access *only* to the few pages supplied by the user then assessing TFIDF weightings over a set of documents where all documents are drawn from a positive class is possible. The result will be meaningless as there is no negative class to compare against. This strategy is therefore not useful in this situation. Given that the search space for AISIID is the entire web, the best estimate of TFIDF for a single feature would therefore have to take the entire web as the collection of documents from which each feature is drawn.

The process of weighting words found in the initialisation documents proceeds as follows. The initialisation documents are first concatenated to form one single document. It is from this single document the term frequency (TF) of a feature is calculated in the usual manner by counting the number of times a word occurs

---

[3] It is acknowledged that in traditional Information Retrieval, there is no positive or negative class unlike a traditional classification task. However, for illustrative purposes this terminology is used in the following situation where "positive class" refers to the topic or typical content of a target document (or collection of documents), whilst "negative class" refers to the typical content or subject of all other documents within a set.

throughout the document. The maximum TF is computed as the greatest TF encountered over all words in the document. Term frequency is normalised using this maximum value

The inverse document frequency of a term, is calculated using a search submitted to the search engine Google. A correctly constructed search will return the number of web pages containing that term, which in turn gives an estimate of document frequency of that keyword over the web. The inverse document frequency is computed using the total number of documents Google claims to index, as stated near the bottom on the Google homepage. This is currently (September 2005) the figure 8,058,044,651. The accuracy of this document frequency is based on the assumption:

Google indexes such a large proportion of available webpages that using Google to determine document frequency will result in a value which is a good estimate of the true document frequency (a value it is impossible to calculate exactly).

It is believed that this assumption is reasonably satisfied. Using the Google programmer's API, it is possible to submit automated queries to this search engine (Google, 2005b) and this is the method used.

The TFIDF weighting (w) of a word (i) in the initialisation document is finally computed as follows:

$$w_i = \frac{f_i}{\max f_i} \times \log_2 \frac{N}{n_i} \qquad \textbf{5.4}$$

Where f_i is the raw frequency of term i in the initialisation document. The maximum frequency is computed over all the terms that appear in the initialisation document. N is the total number of documents in the collection from where the initialisation document is found. N is therefore, ideally, the number of documents on the internet. The number of documents indexed by Google is used as an estimate of this, so N is constant at 8,058,044,651. n_i is the number of documents in the collection in which word i occurs. This figure is arrived at by submitting a query containing only word i to Google thus giving the number of pages indexed by Google in which term i occurs and therefore an estimate of the frequency of i over the entire web.

As no reference to this strategy for determining weighting has been found in the literature, it is reasonable to demonstrate first that the resultant word weights are sensible. That is, given the enormous size of the constant N in the IDF calculation it is important to demonstrate that in the TFIDF calculation neither the TF or the IDF will dominate. It is beyond the scope of this part of the thesis to demonstrate mathematically

that this is the case, but empirical tests can be performed. Over a number of documents taken from the web almost at random, the DF when calculated for all features ranged from 1 to 992,000,000 but due to the $\log_2$ suppression, the resultant IDF part of the TFIDF function ranged only between 3 and 32 – a reasonable range even for a more traditional document set where the range of DF values will tend to be a lot smaller.

5.4.5.3 Initialisation of RWV and ITV

As described above, Google is used to generate the words to populate the RWV. The length of the RWV was set at 50 words as this was considered a reasonable length, trading accuracy of the result for speed. However, it is important to note that Google does restrict the number of automated searches so it is often not possible to submit all words for all training documents for TFIDF analysis. In addition, submission of a word to Google takes around 2 second to complete for each word, but at times of heavy loading this can be considerably longer. Therefore ideally only words likely to have TFIDF weights within the top 50 words when ranked should be submitted. Term frequency (TF) is a reasonable metric to base this ranking on as it is one of the two terms in the TFIDF calculation and can be easily obtained. A very low TF value is highly unlikely to result in a high TFIDF value, and therefore the term will be left out of the RWV. The top 500 words, as ranked by their term frequency in the initialisation document, are submitted to Google for automated document frequency (DF) analysis. A full justification of this figure of 500 words is given in Appendix D. In practice it is possible to remember the TF values for individual words between runs of the algorithm. This is not unreasonable, as the figures used in the IDF calculation are so vast that the change in the DF figure, as retrieved from Google, on a short timescale is extremely unlikely to make an appreciable difference to the result. Especially seeing as the range of the IDF figure is further compressed with the $\log_2$ function. Remembering past DF values for single attributes can therefore increase the speed of the initialisation procedure. It can ensure all words that should be included in the TFIDF calculation are included but with little risk of decreasing the accuracy of the result.

Once the TFIDF values for the most frequent 500 words have been computed, the words are ranked by their TFIDF value and the top 50 are selected to form the cell's RWV. An initial set of immune cells is then created using the same 50 word RWV. The ITV of each is populated by choosing WordNet transformations at random and unlike the RWV, the ITV of each cell will therefore be different.

Each cell's stimulation level is initialised at a user defined value, and the location of each cell is set to a starting page. This is chosen at random from the small set of pages specified by a user (the initialisation set). The system is then ready to begin the running stage.

5.4.6 Running

As AISIID is single-threaded, an order must be established with which to process the members of the population. There are a number of options, the simplest being that each cell is processed in turn until all have been examined, at which point the process will begin again from the start. However it is believed that the speed with which interesting web pages can be found can be dramatically increased by using a priority based approach. The population is held in a sorted queue where the order of the queue is based on cell stimulation level. The higher the stimulation level of a cell, generally the better that cell is doing at finding interesting web pages. Therefore it was decided that the most stimulated cells, those at the head of the queue, should be tested first.

During each iteration, the cell at the head of the queue is removed. This is referred to as the "active cell", and the procedures described in sections 5.4.6.2 to 5.4.6.5 are applied.

5.4.6.1 Creation and Use of Mini-documents

A cell is required to make a choice regarding which page to move to. For this to occur, each hyperlink on a webpage must be weighted. Part of this process is the creation of *mini-documents*. To generate one mini-document (to be associated with one hyperlink), all words within a distance of K_{radius} words around that hyperlink are added to an initially empty set of words. This set is associated with that hyperlink. Figure 5.3 shows an example of this process in which K_{radius}=2. The words on the webpage are converted into a set of words forming the mini-document (md) where the hyperlink in this case is "f".

Webpage = "a b c d e f g h i j k" → md = <d, e, f, g, h>

Figure 5.3. Generation of mini-document from webpage

In the situation where there are fewer words between K_{radius} and either the beginning or the end of the webpage the mini-document is shortened. In this situation it

does not make sense to wrap around from the start to the end of the webpage or vice versa.

This hyperlink weighting strategy could bias the selection mechanism against hyperlinks within K_{radius} of the beginning or end of a page as the mini-documents will contain fewer words. As no alternative is obvious to the current strategy future work may be required to examine the impact of this and change the weighting scheme accordingly, this is expanded later in Chapter 7.

The figure used for K_{radius} is important as it should be large enough to capture the description of that hyperlink in the text surrounding the hyperlink but small enough such that only the text describing the hyperlink, and therefore presumably the content of the destination page, is associated with that hyperlink.

For each mini-document produced, the proportion of features (words) present in that mini-document also present in the cell's RWV is returned and this value is associated with the mini-document for the purposes of weighting that mini-document. The weighting of each hyperlink is computed using the following:

$$\text{weight}_{md,w} = \frac{\sum_{i=1}^{|RWVw|} \delta_i}{|RWV_w|} \text{ where } \delta_i = \begin{cases} 1 \text{ if } RWVw, i \in md \\ 0 \text{ otherwise} \end{cases} \quad \textbf{5.5}$$

In Equation 5.5 the weight of mini document *md* and a webpage *w* is calculated using a count of all words occurring in the *RWV* of *md* and *w*, which is normalised by the size of the *RWV* held by webpage *w*.

5.4.6.2 Immune Cell Movement and Choice

Each immune cell must make a choice of the page it is to move to next, this allows the search space to be explored. A mini-document (and therefore a hyperlink) is chosen by running a roulette wheel selection procedure over the hyperlinks using their weightings as calculated above. These values are used to bias the roulette wheel. The probability of a mini-document being selected is given by the ratio of the hyperlink weight over the summation of the weights of all hyperlinks found on the webpage. Roulette wheel selection is a well-known selection procedure in evolutionary algorithms. (Appendix B).

While this method of scoring the mini-document is relatively simplistic, it is necessary, as more complex metrics based on weighting words using normalised TF or TFIDF then assessing similarity to the RWV are either ineffective or otherwise not viable. It would be possible to compute normalised TF over the set of mini-documents,

but not over the RWV, as this is simply a list of words where each word only occurs once. The IDF cannot reliably be computed either, as the RWV is not drawn from a set of documents.

In addition it should be noted that *only* the RWV is used to assess the estimated interestingness of the target of the hyperlink. While the language used in proximity to the hyperlink may well be a good indicator of the target's relevance (and therefore potential interest) the interestingness, as judged by document distance between the mini document and the words produced by the ITV, is a property of the page itself and not the destination. The value associated with the chosen hyperlink is stored as this is now the estimated interestingness of the target page and is used to provide automated feedback

Procedure for Dealing with Invalid Target Pages

It should be noted that not all hyperlinks embedded on pages will point to valid web pages, or may not point anywhere valid at all. In its current version AISIID can only use HTML webpages and so links to anything else, such as pictures, sounds etc. are filtered. Any links to these known invalid file types are simply ignored during the hyperlink selection process and are identified using the filename extension. The list of known extensions relating to unreadable media types may not catch all files that are not HTML, so when a page is loaded its type is determined. This limitation is discussed in the conclusion of the thesis.

If the file type is of no use, or destination of the URL is not found (an HTTP 404 state is encountered) then the cell will first backtrack to the page from where it came. The URL of this invalid or missing hyperlink is added to a list of invalid URLs and will not be selected again. The cell is then able to re-select a hyperlink. If the current page contains either no alternative hyperlinks or all the links it does contain have been previously identified as invalid, then the cell will backtrack again. This process can continue, in an extreme situation until the original page is reached (although in practice this is unlikely to happen).

5.4.6.3 Assessing Interestingness Using Affinity

This stage of the running process requires the immune cell to assess its affinity with the page it has moved to. Therefore, this is the assessment of the interestingness of the page. The current page is processed as described in section 5.4.1.3 and WordNet based transformations are applied to each word in the RWV in the same manner as described

previously. The result of these transformations produces a set of interesting words. This set of words is used in the affinity calculation between the page and the cell as described in 5.4.4.

The affinity between the page and the cell is stored for the purposes of ranking the results to be shown to the user upon completion of the run. If the current page has not been seen by any cells before, then the affinity value is associated with the page and a record of the active cell's ITV is stored. If, however, the page has already been seen then the affinity is checked against that already associated with the page. If the affinity between the page and the active cell is greater than that already stored the current cell's ITV and affinity value will replace the stored value. This affinity value determines the number of clones produced (section 5.4.6.5).

5.4.6.4 Automated Feedback

As outlined in section 5.4.1.1, the co-stimulation model is used once again to stimulate or suppress cells based on their quality. It is reiterated that the user cannot do this in an interactive manner so an automated scheme must be implemented.

As cells move between pages they do so in a probabilistic manner, they do not always move to the "best" page out of a set of potential pages. This is to promote diversity of search. It should be stressed that due to this probabilistic hyperlink selection mechanism it is quite possible for a cell to move to a webpage irrelevant to the current search. The affinity score cannot be used to determine signal two, as the affinity with the page could be "low" while the cell is otherwise useful. This is why cell stimulation is varied based on the *difference* between the estimated and actual affinity with a web page, rather than an absolute value based on the affinity. For example, a given a cell may predict a page will have a low interestingness, but moves to it due to the probabilistic nature of the selection mechanism. If the interestingness is correspondingly low the cell has *correctly predicted the page would not be interesting* and is consequently rewarded for this correct prediction.

The mechanism for this is as follows. Once a cell has moved to a page and the affinity of the cell with the antigen (current webpage) has been assessed, it is possible to use compute a value for this signal two. For this to occur, the actual affinity value between the cell and webpage can be combined with the estimated affinity value as computed when the hyperlink to that webpage was chosen. It is important to note that the difference between the estimated affinity value and the actual affinity value is used to calculate the final signal two value.

As stated before, cells in the AISIID algorithm are allowed to move to pages of little interestingness as this promotes exploration. However moving to poor quality page too often will result in the cell's stimulation dropping below a threshold and the cell being removed. The stimulation of a cell is proportional to the quality of the estimate it made regarding the current page's interestingness as shown in Equation 5.6. Note that the stimulation of a cell will always decrease, thus bounding a cell's lifetime, again much as seen in AISEC. Without this bounding it is theoretically possible for a few cells to suffer continual up-stimulation and therefore dominate the population. This would presumably lead to a significant reduction in diversity.

Equation 5.6, below, shows the calculation used to determine cell stimulation level. The stimulation of a cell at time t+1 is calculated where aff is the affinity of the cell c with the webpage w (antigen) while aff_{est} is the estimated affinity which was computed based on the mini-document whose hyperlink target was the cell's current web page.

$$\texttt{stimulation}_{t+1} = \texttt{stimulation}_{t} - \texttt{abs}\left(10 \times \left(aff_{est} - aff_{c,w}\right)\right) \qquad \textbf{5.6}$$

5.4.6.5 Cloning and Mutation

Recall, if the affinity of the cell with the current page is above a threshold, the cell has found what is considered to be an interesting page. This is rewarded with the ability to clone and mutate. Both cloning and mutation will be performed with regard to the affinity; the number of clones being proportional to affinity while number of mutations being inversely proportional to affinity. The number of clones produced based on the affinity of a cell with a web page is defined by Equation 5.7 where K_{clo} is a constant controlling the rate of cloning and K_{ct} is a constant that controls the maximum number of clones. $aff_{bc,w}$ is the affinity of the cell to be cloned with webpage w.

$$\texttt{num\_clones} = \texttt{if} \begin{cases} \lfloor \left(K_{clo} \times aff_{bc,w}\right) - K_{ct} \rfloor > 0 & \lfloor \left(K_{clo} \times aff_{bc,w}\right) - K_{ct} \rfloor \\ \texttt{0 otherwse} \end{cases} \qquad \textbf{5.7}$$

The number of mutants produced is defined in Equation 5.8. $aff_{parent,w}$ is the affinity of the parent cell with webpage w. $\texttt{K}_{\texttt{mut}}$ is a constant controlling the rate of mutation and $parent_{ITV}$ is the parent cell's set of word transformations.

$$\texttt{num mutate} = \lfloor \left(1 - aff_{parent,w}\right) \times K_{mut} \times \left| parent_{ITV} \right| \rfloor \qquad \textbf{5.8}$$

Upon cloning, each of the new cells receives the ITV and RWV of its parent. Mutation then occurs (in the ITV, but not the RWV) and values in that vector are replaced by other legal values only. No gene libraries are used for this mutation as the implementation to change the value of a WordNet operation is trivial.

After mutation, the location of each clone is temporarily placed on the same page as its parent. The clone then moves one hyperlink away from that page in the usual manner. Once it has moved, the cell is initialised with a default stimulation level and is placed in the population.

5.4.7 Population Dynamics

Recall, in the previous step cells may be punished for finding pages that are not deemed interesting. This is achieved by reducing a stimulation counter for the cell.

Population control also manifests itself in a similar way to the control of memory cells in AISEC (Chapter 4). It is important to guard against redundant cells (those in area of the search space that is already covered by other, fitter cells) in the population. If the number of cells on a single page is above a threshold then each cell currently on that page will incur a penalty of a reduced stimulation count. This reduction in stimulation count will be in proportion to the number of other cells also residing on that page. Given a page on which a number of cells are currently placed, if the population size is low, cells tend to have a chance of moving from that page before their stimulation is reduced below the threshold at which they will be removed. However, if the population size is high a cell will have its stimulation reduced a number of times before it becomes the focus of the main procedure again and can move, thus only the very best few survive. This technique allows the population to dynamically grow as the search area (number of visited pages) grows. As this suppression only occurs when the number of cells on a single page is above a threshold it does *not* impart a global limit on the numbers of cells in a population, but imposes population restrictions on a local level which tend to result in global population control. At the end of each iteration each cell's stimulation count is checked. If it is found to be below a threshold the cell is removed from the population. Otherwise the cell is maintained in the population.

Alternatives

During development a number of strategies were applied. The simplest was to impart a global maximum population size on the system. When a specific limit was reached the least stimulated cells were removed until the population size dropped back below that

threshold. Empirical tests showed this to be a little restrictive; it was hard to pick a correct value for this attribute. Also tested was the strategy of only allowing a certain percentage of the population to reside at one node. This is a lot more flexible than the previous strategy as it allowed local population flexibility while containing the global population. However it was too flexible for the following reason. Given, during the initial stages of spidering, few pages have been found it is probable that a large proportion of cells may need to reside on a single page. Thus the threshold should be set high enough to allow to happen. However, if this value is set to the level acceptable in this situation, as spidering continues the value tends to be too high. Once page or a small group of pages dominates with a vast number of cells, to the detriment of diversity. This situation was observed to get worse as spidering continued as any increase in the number of cells not residing on that page would increase the total number of cells. As the population bounds are based on a proportion of the population, the number of cells at these dominant nodes also grew. This strategy therefore tended to result in a run-away growth of the population where cells were confined to a small number of nodes.

5.4.8 Returning Results

When a stopping criterion is met, the user is presented with a ranked list of results. This stopping criterion may be a certain number of pages have been discovered or a certain number of iterations have taken place. The results consist of a list of URLs that one or more cells visited during the run of the algorithm. For each page found during a run, the mean affinity between all cells that encountered that page and the page itself is computed. The pages are then ranked according to this mean affinity, the higher the mean affinity, the higher the ranking of that page.

The contents of the cell's ITV that resulted in this affinity may also be retrieved at this time and the words generated by it could be shown to the user. This not only imparts knowledge of which pages were the best and how good they were, but also gives the user an idea regarding *why* they were ranked highly. Revealing to the user why such a system make a particular decision, such as giving an example a particular class, is one of the examples of good quality output from a classifier as defined in section 2.2.3 where the following was stated – "*It is important that a user is able to question why a decision was made by the algorithm. This allows the user an extra level of information, other than the algorithm output*". Indeed, in the paper (Liu et al., 2001),

the authors of the paper also reveal the keywords and concepts that are thought to be interesting, and thus this process seems an important one to have in place.

As well as enlightening the user to the decision making process, from this information the user may want to continue his/her search for interesting information by submitting the interesting words to a search engine he or she is familiar with.

5.5 Pseudocode

This section presents the pseudocode for the AISIID system, as described in the preceding section. The following conventions apply and legal ranges are shown in Table 5.1.

Parameter	Legal Range
K_{stim}	> 0
K_{clo}	0 - 1
K_{mut}	0 - 1
K_{stim}	> 0
K_{size}	> 0
K_{top}	> 0
K_{radius}	> 1
$K_{supress}$	> 0
$K_{proxsup}$	0 - 1

Table 5.1. Parameters and legal ranges for AISIID

Some of the following conventions are similar to those used in AISEC and it is hoped this makes the explanation easier to follow. However, while the functions may be similar, the actual values used will not be shared between the two:

- Let `BC` refer to an initially empty set of naïve immune cells (B-cells) where `bc` is used to denote one element of `BC`, that is, one individual cell, where also:
  - bc_{RWV} is the set of relevant words related to `bc`. E.g. `<spaghetti, chips>`.
  - bc_{ITV} is the set of transformations related to `bc`. E.g. `<1,3>`
  - bc_{pos} is the current position of `bc` on the web. E.g. "`www.foo.com/index.html`"
  - bc_{stim} is a real number representing `bc`'s current stimulation level.
  - This notation above also extends to derivatives of a cell (temporary copies, new clones, etc).

In addition, the following parameters are used.

- Let K_{clo} refer to a constant which controls the rate of cloning
- Let K_{mut} refer to a constant which controls the rate of mutation
- Let K_{stim} refer to the initial stimulation level for cells
- Let K_{size} refer to the initial number of cells generated during initialisation
- Let K_{top} define the number of elements in the bc_{RWV} and therefore bc_{ITV}
- Let K_{train} refer to the set of interesting pages selected by the user
- Let K_{radius} refer to the radius of a mini document
- Let $K_{supress}$ refer to a threshold beyond which cells will suppress each other.
- Let $K_{proxsup}$ refer to a constant controlling the rate of cell suppression due to their proximity to others in terms of physical position (same page).

5.5.1 Representation

A cell is represented as follows:

```
B cell = <RWV,ITV,position,stimulation>
```

Where

```
<RWV> = <word1, word2...word_n>
<ITV> = <operation1,operation2...operation_n>
```

In the implementation, the RWV contains Strings representing the words while the ITV contains integers in the range [0,3] each identifying one of the four unique WordNet operations.

1. Antonym
2. Synonym
3. Hyponym
4. Hypernym

5.5.2 Affinity

Given a current cell, bc, and a webpage processed into the form of an antigen, ag, the affinity between bc and ag is illustrated by Pseudocode 5.1. In this pseudocode, $count_{x,y}$ is a count of features found in both x and y. IWV is a vector of interesting words generated by WordNet using the RWV and the WordNet transformations defined by the elements of ITV. Therefore $count_{INT,ag}$ is a count of features found in both ag

and the set INT. The result of this function by definition will always return a value in the range $0 \geq$ aff ≥ 1.

```
1   PROCEDURE affinity (bc, ag)
2     INT ← ∅
3     FOREACH(location i in bc_ITV)
4       w ← word in location i of bc_RWV
5       int_words ← generate set of words resulting by transforming
                    w using WordNet operation in location i of bc_ITV
6       INT ← INT ∪ {int_words}
7     aff ← 1/2 × ( count_{bc_RWV ag} / |bc_RWV| + count_{INT,ag} / |INT| )
8   RETURN aff
```

Pseudocode 5.1. Affinity between immune cell and antigen (web page)

5.5.3 AISIID Algorithm

The following section describes the main algorithmic features of AISIID using the affinity measure and cell representation defined above.

5.5.3.1 Initialisation

```
1   PROCEDURE initialise()
2     W ← ∅
3     BC ← ∅
4     SCORE ← ∅
5     FOREACH(te ∈ K_train)
6       FOREACH(word w in te)
7         W ← W ∪ {w}
8     FOREACH(w ∈ W)
9       DF = document frequency of w as recorded by Google
10      TF = term frequency of w in K_train
11      wscore = TFIDF of w as computed using Equation 5.4
12      SCORE ← SCORE ∪ {w,wscore}
13    W_top = Determine top Kw words as ranked by wScore in SCORE
14    DO K_size TIMES
15      Bc_RWV ← W_top
16      Bc_stim ← K_stim
17      FOREACH(position i in bc_ITV)
18        i ← random value in range [0,3]
19      Bc_pos ← random element of K_start
20      BC ← BC ∪ {bc}
21  RETURN BC
```

Pseudocode 5.2. Initialisation of ASISIID

This procedure produces a set of cells, the number of which is dictated by K_{size}. Lines 5-7 generate a set of all words in all training documents (K_{train}) where te is an element of K_{train}. Lines 8-12 rank these words using TFIDF where document

frequency is drawn from Google. Lines 13-20 populate the cell set with K_{size} number of initial cells.

5.5.3.2 Main Algorithm

```
1    PROGRAM aisiid
2      BC ← initialise()
3      WHILE(|BC|>1)
4        bc ← cell at head of population queue
5        wp = Load webpage at URL denoted by bc_pos
6        IF (wp is illegal)
7          bc_stim ← bc_stim - 1
8          move bc to parent page using bc history
9          loop from line 3
10       ag ← process wp into antigen
11       aff = affinity(bc,ag)
12       bc_stim ← bc_stim - 10 ×ABS(aff-bc_estimated)
13       IF(aff>K_clo)
14         NEW = clone_mutate(aff,bc)
15       NEW = NEW ∪ {bc}
16       FOREACH(cell c in NEW)
17         FOREACH(hyperlink h in ag)
18           md ← create_minidoc(h,ag)
19           MD ← MD ∪ {md}
20         FOREACH(mini-document md in MD)
21           countR ← number of elements in md present in c_RWV
22           score_md ← countR / |c_RWV|
23         h_new ← result of a roulette wheel selection over all
                   mini-documents
24         c_pos ← h_new
25         c_estimnated ← score_md of mini-document selected in line 26
26         population  ← population ∪ {c}
27       FOREACH(cell c in BC)
28         numCells ← determine how many cells at c_pos
29         IF(numCells>K_supress)
30           c_stim ← c_stim - (numCells * K_proxsup)
31         IF(c_stim < 0)
32           remove cell from population
33       re-sort population with regards to stimulation level
34       loop from line 3
```

Pseudocode 5.3. AISIID main algorithm

The main algorithm consists of 8 stages within a loop. Each of these stages is detailed in Pseudocode 5.3, but to aid clarity the stages are shown below and the associated lines of pseudocode are referenced by line using the numbers in brackets.

1. Chose next cell of population (4)
2. Check cell's current webpage is legal, if not then backtrack (5-9)
3. Compute affinity between cell and webpage (11)
4. Perform automated feedback on cell and stimulate or suppress cell based on outcome (12)

5. Clone and mutate cell based on affinity, picking a new page for each new cell and the parent cell to move to next (13-25)
6. Estimate and remember the estimate of quality for this new page (22, 25)
7. Add new clones to population (26)
8. Perform population metadynamics. That is, removal of the "worst" cells in order to avoid a significant increase in the population size. The population is also reordered by descending stimulation level. (27-32)

It should be noted that the order with which the feedback and the clone/mutate routines are executed is unimportant as automated feedback does not influence the cloning ability of the cell.

The attribute $bc_{estimated}$ appears on line 12 and at first it would appear that this value is undefined. However this attribute had been set on line 25 of the previous iteration.

5.5.3.3 Clone and Mutate Procedure

Pseudocode 5.4 shows the procedure used for cloning a cell a number of times, and mutating those clones. The number of clones is proportional to the affinity of the cell and is calculated according to the equation in line (3). The number of positions of the ITV vector to be mutated in each clone is inversely proportional to the affinity of the cell and is calculated by the equation on line 4. The symbol $\lfloor x \rfloor$ denote the floor of x, that is x rounded down to the nearest integer.

```
1   PROCEDURE clone_mutate(bc,affinity)
2     clones ← ∅
3     num_clones ← ⌊aff × K_clo⌋ - K_ct
4     num_mutate ← ⌊ (1-aff) × |bc_ITV| × K_mut ⌋
5     DO(num_clones)TIMES
6       bcx ← a copy of bc
7       DO(num_mutate)TIMES
8         p ← a random point in bcx's feature vector
9         i ← a random integer in the range [0,3]
10        replace value in bcx_ITV at location p with i
11      bcx_stim ← K_stim
12      clones ← clones ∪ {bcx}
13  RETURN clones
```

Pseudocode 5.4. Procedure for cloning and mutating a cell

5.5.3.4 Generate Mini-document Procedure

Pseudocode 5.5 shows the procedure for creation of a mini document centred around hyperlink `h`, where `h` is contained in the webpage represented by `ag` (antigen). In this pseudocode, given the antigen represents a document as an ordered list of tokens (words), the position n is the index in this ordered list at which hyperlink h occurs. E.g. $n = 5$ is the 5^{th} word in the ordered list.

```
1   PROCEDURE create_minidoc(h, ag)
2     MD ← ∅
3     n ← position of h in ag
4     MD ← MD ∪ {word found at position n in ag}
5     DO K_radius TIMES
6       IF(n = end of document)
7         GOTO line 10
8       MD ← MD ∪ {word found at position n in ag}
9       n ← n+1
10    n ← position of h in ag
11    DO K_radius TIMES
12      IF(n = beginning of document)
13        GOTO line 16
14      MD ← MD ∪ {word found at position n in ag}
15      n ← n-1
16    Return MD
```

Pseudocode 5.5. Create Mini-document procedure

5.6 Visualisation of Search Progress

While not part of the information retrieval process of the AISIID algorithm, the user interface allows the user to see how the AISIID cell population is behaving, and as such leads to a greater understanding of the AISIID page identification process. In this context it can be used as an insight into determine whether the spidering process is working broadly as expected.

The user interface consists of two windows, one visualising the number of cells on a page, the other visualising the interestingness of discovered pages. Figure 5.4 and Figure 5.5 show examples of the visualisation.

The status of AISIID is shown as a mathematical graph. Each node represents a web page with each edge representing a hyperlink between two pages that at least one cell has taken to get from one to the other. As new pages are found they become attached to the graph. When a page has been found, but has no cells at that location it stays connected to the graph but its radius is reset to a default small value. Both graphs are laid out exactly the same so comparisons may be easily drawn between them.

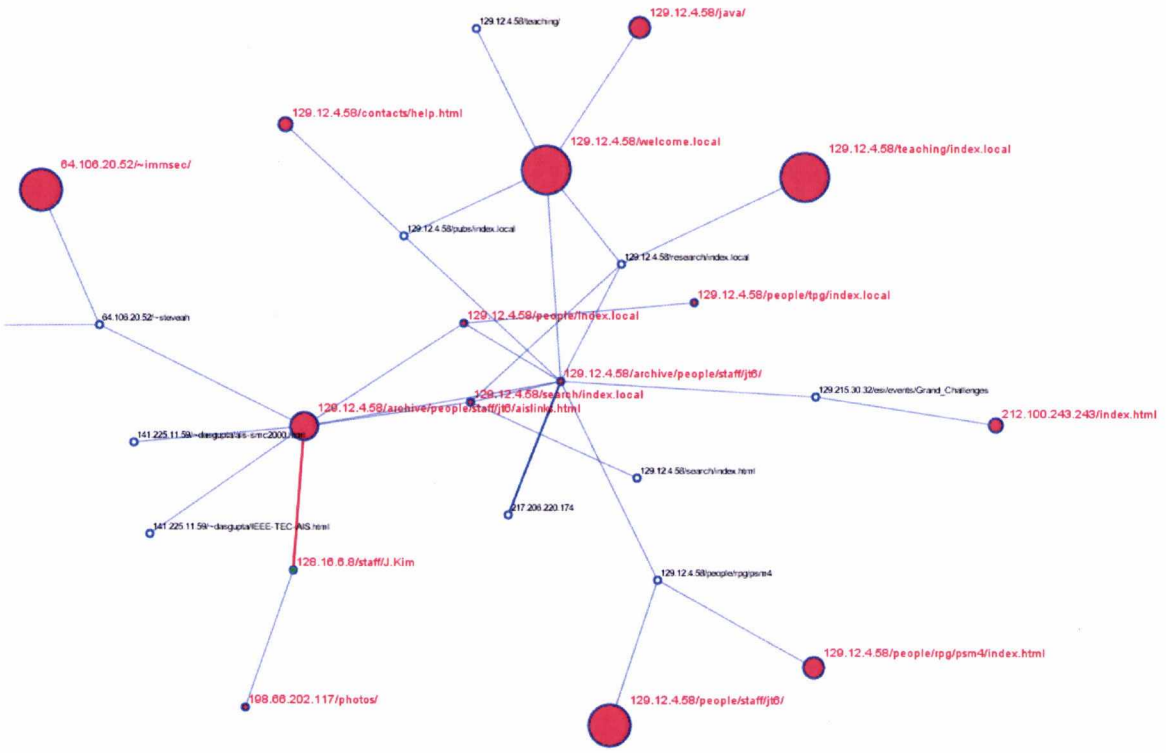

Figure 5.4. AISIID visualisation of cell distribution

Figure 5.5. AISIID visualisation of interestingness distribution.

The graph layout is performed automatically using graph drawing libraries developed in (Mutton, 2004). The width of an edge will increase in proportion to the number of cells traversing a hyperlink. The radius of the node also conveys information depending on the graph in question. With regard to the cell distribution graph (Figure 5.4), the node radius increases in proportion to the number of cells currently at that position on the web. With regard to the interestingness distribution graph (Figure 5.5) the node radius depicts the maximum interestingness score each page has received so far.

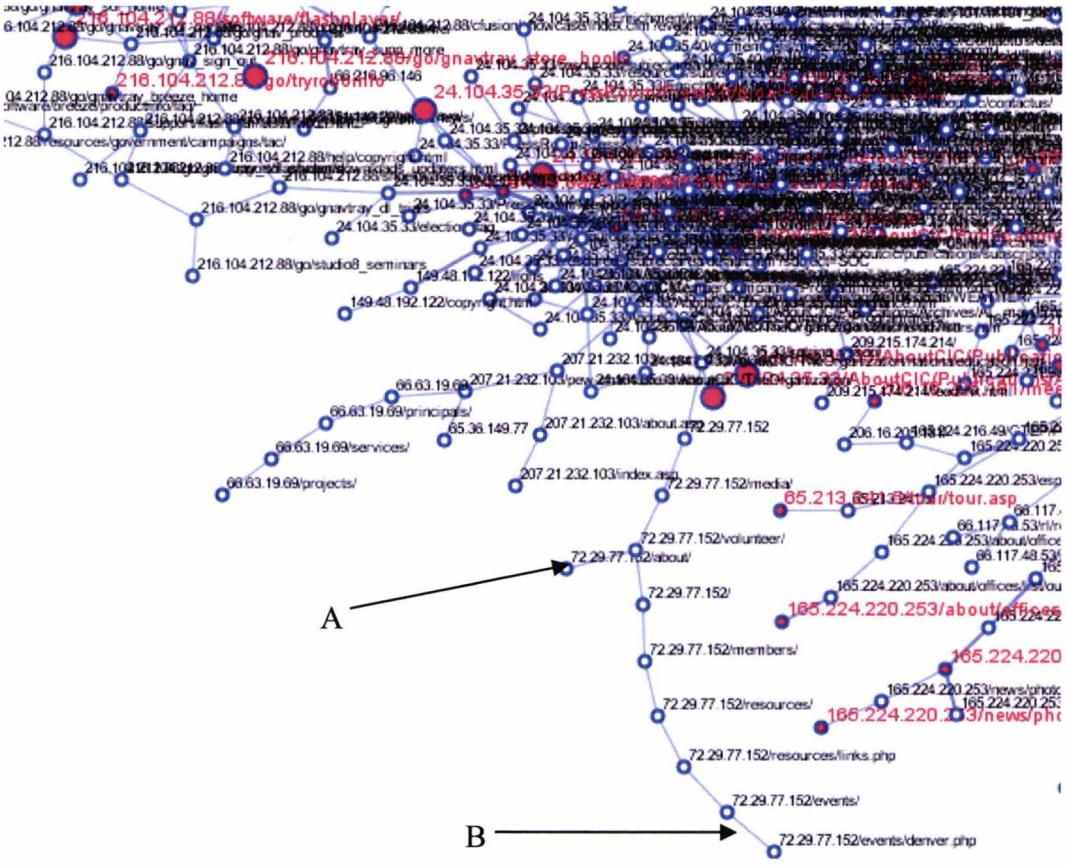

Figure 5.6. Analysing AISIID spider technique

Inspection of the graphs tends to reveal some interesting observations especially regarding the usefulness of the spidering technique. Label A on Figure 5.6 shows a point where a cell had one clone and moved to just one more page before its stimulation level dropped too low and was removed. The offspring cell was then able to continue searching for interesting information but was not able to find anything. There are numerous examples of cells such as this moving from page to page without finding a page interesting enough to stimulate it to clone. See label B on Figure 5.6. This results in a linear string of pages but shows that the algorithm is behaving as expected. That is,

the cell had a certain stimulation level and as it moved from page to page finding uninteresting pages its stimulation level gradually reduced.

In many of the graphs that are generated it can be noticed where AISIID cells have moved from areas of high interestingness, through areas of low interestingness, back to areas of high interestingness again. Other, less robust directed spidering algorithms may have stopped when they encountered the area of reduced interest as it had seemed to not be producing good results; however, if this was the case other areas of high interest may not have been discovered. This situation is quite explicit in Figure 5.7 and Figure 5.8. In these figures it can be seen that a single cell or small number of cells has traversed a sequence of around 12 pages that are not regarded as particularly stimulating (circled in Figure 5.8) to the cell but has found a new cluster of interesting pages thus demonstrating the advantage of the robust method of directed spidering.

In summary, it can be seen that AISIID attempts to comply with the assertion of Wu and Hsu: "*some sets of off-topic documents often lead to highly relevant documents*" thus "*an optimally focused crawler should sacrifice visiting several off topic pages in order to reach the highly relevant pages among the hyperlinks*" (Wu & Hsu, 2005).

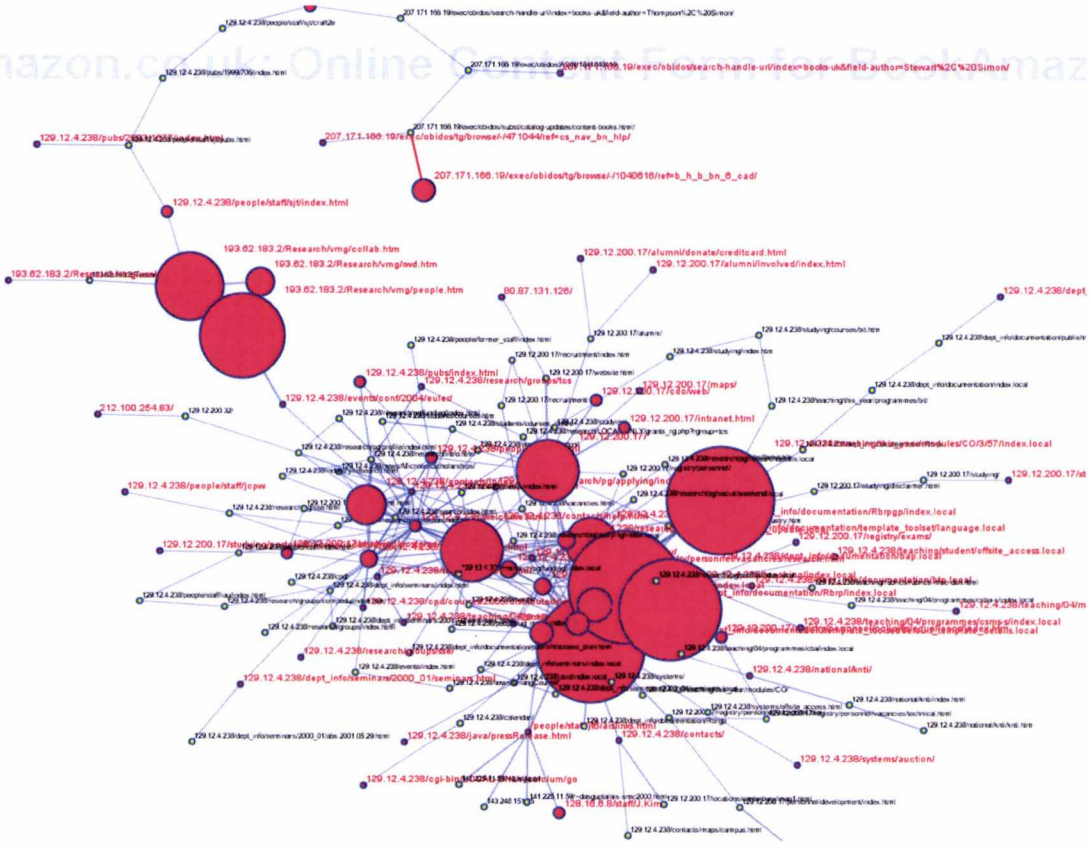

Figure 5.7. Example of good spidering technique (1). Other directed spidering techniques may decide the path at the top of the graph is not worth pursuing.

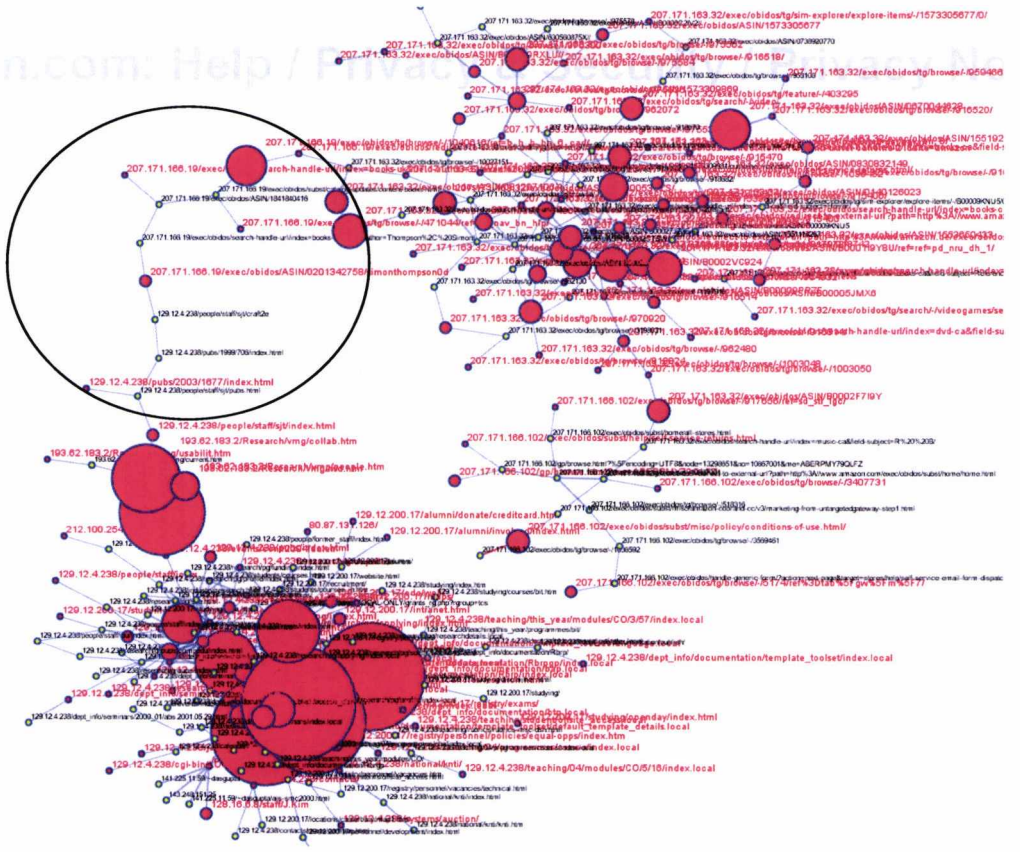

Figure 5.8. Example of good Spidering technique (2). AISIID's spider continues and finds an area of the web that is highly useful. Area of low interestingness is circled.

5.7 Summary

In this chapter an algorithm for discovery of interesting information from the web has been described. Named AISIID (Artificial Immune System for Interesting Information Discovery), the algorithm has a number of novel aspects combined to create an original algorithm. Based on the hypothesis that *documents containing a high frequency of words semantically related to the typical words contained in a set of known documents will be of greater interest to the user than a set of documents ranked using relevance alone,* an artificial immune system algorithm has been described to exploit this hypothesis. It was acknowledged that users have trouble in directly defining knowledge that would be surprising to them, thus AISIID allows a user to specify what they already know, with surprising information being inferred and identified from that. In the following chapter, AISIID is tested with the aim of observing its characteristics. In addition, AISIID's performance is subjectively evaluated using real users.

Chapter 6 Analysis of AISIID

In the previous chapter, the Artificial Immune System for Interesting Information Discovery (AISIID) was described. It was suggested there is a gap in the literature with little research occurring into the identification of interesting information in text documents. This is in contrast to the rule-based classification community where this process has been acknowledged for many years (Liu & Hsu, 1996; Liu et al., 1997; Suzuki & Zytkow, 2000; Carvalho et al., 2003). This lack of research could possibly be down to the difficulty of realising a reasonable solution to the problem and the difficulty of formalising interestingness as a concept. It was also stated that using currently available techniques, users may find it hard to search for surprising knowledge as, by definition, this can be hard for them to specify.

Objective testing of the AISIID system in terms of assessing its interesting page retrieval performance is extremely difficult. There is only one system, that of Liu et al. (Liu et al., 2001), in the literature that can address this issue. Both systems work in fundamentally different ways and as such cannot be directly compared in an objective manner. The system of Liu et al., known as WebCompare, requires a set of pre-spidered pages to be found, over which the algorithm works, whereas AISIID seeks out pages itself. The only way at present for a measure of quality to be given to the output of these algorithms is in a subjective manner involving the user and so this is the strategy this chapter employs. This was a situation also encountered by Liu et al.: "*Since the proposed system deals with subjective interestingness of information, it is difficult to have an objective measure of its performance*". It would be intractable to require a user to rank all pages AISIID could possibly retrieve. Therefore the traditional metrics of classification accuracy, precision and recall are unavailable. In any case, the notion of interestingness is specific to the user and so real users are used to give their subjective evaluation of the interestingness of the web pages returned by AISIID.

While testing the *output* of AISIID in an objective manner is intractable, it is possible to investigate the characteristics of AISIID as an algorithm in its own right.

This takes place in the following section with the aim of understanding some of the behaviours and characteristics of the AISIID algorithms. The subjective tests, those that measure the quality of the output, are reserved for section 6.3.

6.1 Objective Tests

This section describes the results of a number of tests, each with the aim of revealing AISIID's characteristics as an algorithm. These objective tests also allow the investigation of some design and implementation decisions. In Chapter 5, a number of decisions were made when designing the system, such assertions are tested in this section. For each test, a description of that test is given, including a statement regarding what is to be expected or what would constitute a successful test if appropriate. The results are then presented and interpreted. There are a large number of tests that could potentially be run on a system of this complexity. Since it is not feasible to run all of these tests, those considered to give the most insight into the basic running of the algorithm or that directly test assertions made in the previous chapter regarding design decisions are shown here.

6.1.1 Test Protocol

Throughout section 6.1, unless otherwise stated, the following experimental protocol was observed. As the algorithm is nondeterministic, AISIID was run 30 times using the same start pages and the same user attribute values but a different random seed each time. Recall that a set of pages must be chosen to provide the AISIID system with an initialisation set, K_{train}. It is from this set that the user's interest is inferred and it is on these pages that the initial AISIID immune cells will be placed. Seven start pages were used, all on one topic, although for these tests the actual topic is unimportant as no user will see the retrieved pages. The URLs of the start pages are shown in Figure 6.1.

```
http://www.andysecker.com/research
http://www.cs.kent.ac.uk/archive/people/staff/jt6/
http://en.wikipedia.org/wiki/Artificial_immune_system
http://oak.cpsc.ucalgary.ca/ICARIS-2005/
http://www.cs.kent.ac.uk/archive/people/staff/jt6/aisbook
http://www.cs.kent.ac.uk/archive/people/staff/jt6/aislinks.html
http://www.elec.york.ac.uk/ARTIST/
```

Figure 6.1. Starting URL set

Referring to the previous chapter, AISIID uses a number of parameters. These, are kept constant over all runs of AISIID while testing. To reiterate, the important parameters are shown below:

- Let K_{clo} refer to a constant which controls the rate of cloning
- Let K_{mut} refer to a constant which controls the rate of mutation
- Let K_{stim} refer to the initial stimulation level for cells
- Let K_{size} refer to the initial number of cells generated during initialisation
- Let K_{top} define the number of elements in the bc_{RWV} and therefore bc_{ITV}
- Let K_{train} refer to the set of interesting pages selected by the user
- Let K_{radius} refer to the radius of a mini document
- Let $K_{supress}$ refer to a threshold beyond which cells will suppress each other.
- Let $K_{proxsup}$ refer to a constant controlling the rate of suppression of cells due to their proximity to others.

The values of the parameters used throughout all runs of AISIID are shown in Table 6.1.

Parameter	Default value
K_{clo}	10
K_{mut}	0.2
K_{stim}	50
K_{size}	10
K_{top}	50
K_{radius}	25
$K_{supress}$	6
$K_{proxsup}$	10

Table 6.1. Parameter values for AISIID tests

In all tests, AISIID was tested on the web, not an artificial data set. However it is not necessary to pre-crawl the web and store all pages that could potentially be visited by AISIID. Indeed the sheer number of pages just within a new hyperlinks depth make this intractable. Rather than expend time and network bandwidth pre-spidering, a caching mechanism was used. When a page is encountered by a cell in AISIID that has not been retrieved before, it is stored locally on disk. This local copy will never be refreshed. During the rest of that run of AISIID and *during all subsequent runs*, that local copy is then retrieved in preference to the external version. Thus, while the external version may change after it has been downloaded, the content, as seen by subsequent instances of AISIID, remains the same. Any changes made to the document

before it was first seen by AISIID are immaterial. In the test protocol described below, it is believed that all tests were as fair as possible and any changes in the web environment would have not affected the results.

The stopping criterion was when AISIID had retrieved exactly 1000 different pages. With regard to the charts in this section, some runs of the system will reach their stopping criterion before others, and as such the maximum value for the x-axis on the graph will be the iteration on which the run taking the minimum number of iterations to complete terminates. Thus it is important to note that all points on the chart represent same number of independent tests. Allowing the x-axis to continue past this point, would cause sections of the chart to be plotted with fewer then the stated number of runs combining to form the mean.

Some charts make use of the mean affinity of the population. This is calculated such that at every time step each cell is interrogated to determine the last affinity it evaluated. New clones will determine their affinity with the parent's page when they are created, so all cells will have legal affinity values stored. The sum of these affinities is then divided by the total number of cells in the usual manner. Where applicable, the mean interestingness of the population is computed in the same way, except the individual component of the affinity function is stored and used for this calculation.

6.1.2 Examining Population Size

An efficient AIS will minimise immune cell population size, and therefore computational memory and time, whilst maximising the quality of the result. As the cells in an AIS such as AISIID are used to encode the system's memory, the more data learnt by the AIS the larger the immune cell population is pressured to become. However the larger the immune cell set the less efficient comparisons of all cells in that set become. An AIS must strike a balance between population size and memory.

AISIID allows the user to specify a parameter controlling the population at one single page ($K_{suppress}$). It was hoped that the controlling of the population at each page will have the effect of controlling the population as a whole. The chart, Figure 6.2, shows how the population may vary as this parameter is changed. A successful outcome of this test would show that this balance is reasonable, i.e. the size of the population does not increase unmanageably.

In this test it is expected that a small value for $K_{suppress}$ will result in a smaller population compared to a larger value of $K_{suppress}$, the results are shown in Figure 6.2. It should be noted that on 3 occasions the algorithm failed to terminate when $K_{suppress}$

was set to 1. The reasons for this are explained below but it should be noted that the effect of this is the number of cells in the population of these non-terminating results remain static from a point. As these 3 rogue results were found to unfairly affect the mean for that series, the decision was taken to remove them. Thus the series $K_{suppress}$ = 1 is the mean of 27 runs of the system.

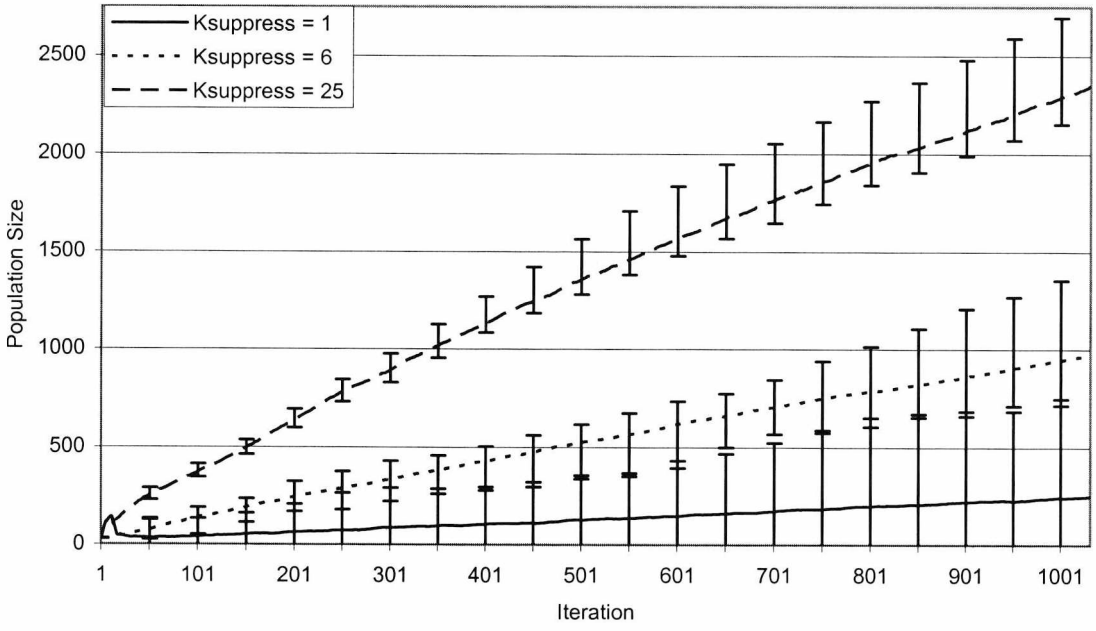

Figure 6.2. Chart showing the variation in population size for three different values of Ksuppress

The small "bump" near the left hand side of the chart present in all series is due to the lack of cell death due to the initial population all starting with a high stimulation level. As the algorithm progresses and these cells spread out, the high mean stimulation level quickly drops, reducing the population size. This has the effect of allowing the initial population to make its way from the initial start pages before reducing in stimulation and therefore being likely to be removed, thus encouraging diversity in the spidering.

It can be seen from the chart that increasing the parameter does indeed increase the overall population size. It can also be seen that, from the error bars, the number of cells present in the population when $K_{suppress}$ = 1 is highly variable suggesting a certain amount of instability associated with this value.

The observation that increasing $K_{suppress}$ results in a higher population size is somewhat intuitive, but there are some more interesting questions that may be asked regarding this situation. Firstly does increasing the population size lead to increased diversity of population and therefore to, on average, a smaller number of iterations

before 1000 pages are found? It is thought that a reasonable value for $K_{suppress}$ will maintain diversity of the population (unlike a small value), without leading to the inefficiency of an overly large population size (as with the large value). A small value will result in a more homogeneous population allowing a single or small group of cells to dominate, resulting in a lack of exploration in the search space. Likewise a large population size will result in the movement away from the initial pages being slowed as numerous cells are moved over pages that have already been seen by others. Table 6.2 and Figure 6.3 summarise the number of iterations required for 1000 different pages to be found (the standard stopping criteria) for differing numbers of $K_{suppress}$.

$K_{suppress}$	Mean	Minimum	Maximum	Standard Deviation
1	2829	1032	10000[4]	2537.
6	2192	1505	2957	424
25	2563	1401	4594	735

Table 6.2. Summary of results for experiments varying $K_{suppress}$

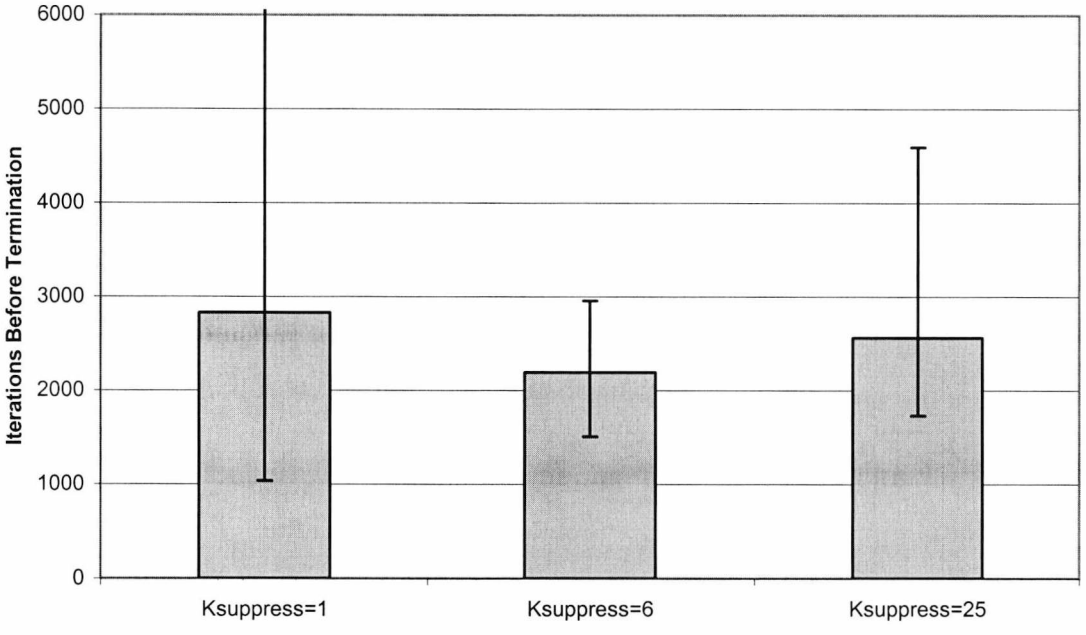

Figure 6.3. Bar chart showing the mean number of iterations required for the system to reach a terminating condition, when varying $K_{suppress}$

It was interesting to note that the figure of $K_{suppress} = 6$ resulted in the lowest mean number of iterations required before termination as, while testing, these parameters were not optimised. A value for $K_{suppress} = 6$ results in by far the smallest deviation around the mean as visualised by Figure 6.3. Student's t-test can be used to test for

[4] The value of 10,000 was substituted where the algorithm failed to terminate.

significance (Appendix C) with the null hypothesis being that the difference in the mean terminating iterations between any of the two $K_{suppress}$ values measured, do not differ. With the threshold for statistical significance set at the usual $P_{null} = 0.05$, it is found that the difference in numbers of iterations needed to terminate between $K_{suppress} = 1$ and $K_{suppress} = 6$ is *not* significant ($P_{null} = 0.18$). This result can be attributed to the enormous standard deviation in the sample for $K_{suppress} = 1$ making it hard to say where the mean for $K_{suppress} = 1$ actually lies. Conversely, the difference between the means of $K_{suppress} = 6$ and $K_{suppress} = 25$ is significant as the probability of the null hypothesis is found to be less than the threshold, in this case $P_{null} = 0.02$.

Also of interest was the fact that $K_{suppress} = 1$ resulted in the minimum and maximum number of iterations necessary for all three values. The runs of the system with $K_{suppress} = 1$ did not terminate on 3 occasions with the algorithm reaching the 10,000 iteration limit before terminating. This situation never occurred with the other two values. This predisposition to not terminate is related to the result that one run of the algorithm with $K_{suppress} = 1$ required only 1032 iterations to discover 1000 pages. When analysing the run producing this result, it was found that one cell was 193 hyperlinks away from the start pages. This compares to a distance of around 20 hyperlinks for the maximum distance any cell made for $K_{suppress} = 6$ (Figure 6.6). It is clear that this one cell, or a small number of cells dominated the population, moving from page to page constantly. While this resulted in a rapid discovery of 1,000 pages (as the cell will always move away from the start pages) it should be presumed that the results retrieved will be extremely poor when concerned with the result shown to the user as the population lacks diversity. This domination is a symptom of the priority population ordering and in extreme situations like this it is detrimental to the quality of the output. This may therefore need some research in the future.

6.1.3 Population Ordering

In Chapter 5 the decision was made to order the population by stimulation level, thus ensuring that the most stimulated cell in the population was examined and moved in preference to all others at every iteration. This strategy was a departure from the more usual two methods of examining cells in sequence so that the cell that has not been examined for the longest length of time is given priority, or the strategy adopted in many AIS and other population algorithms where the affinity of all cells is computed during each iteration. As this strategy is uncommon it was thought that an investigation into its performance would be of interest. Recall that this strategy was adopted as it was

hoped that moving the most stimulated cell at each time would find the most interesting information first. This is because it can be assumed that the most stimulated cell in the population is the one finding the "best" information at each time. Therefore it is possible that the same cell could be moved many times in a row before it begins to find information that is less good, and its stimulation drops below that of one or more other cells in the population. The strategy in which the cell with the greatest stimulation is chosen to be moved will be known as "stimulation priority".

In contrast, the situation which will be known as "age priority" uses a queue structure for storing the population. Once a cell has been moved it is added to the tail of the queue, then the head of the queue is removed and used as the cell to move next. A cell's offspring are always placed in the tail of the queue.

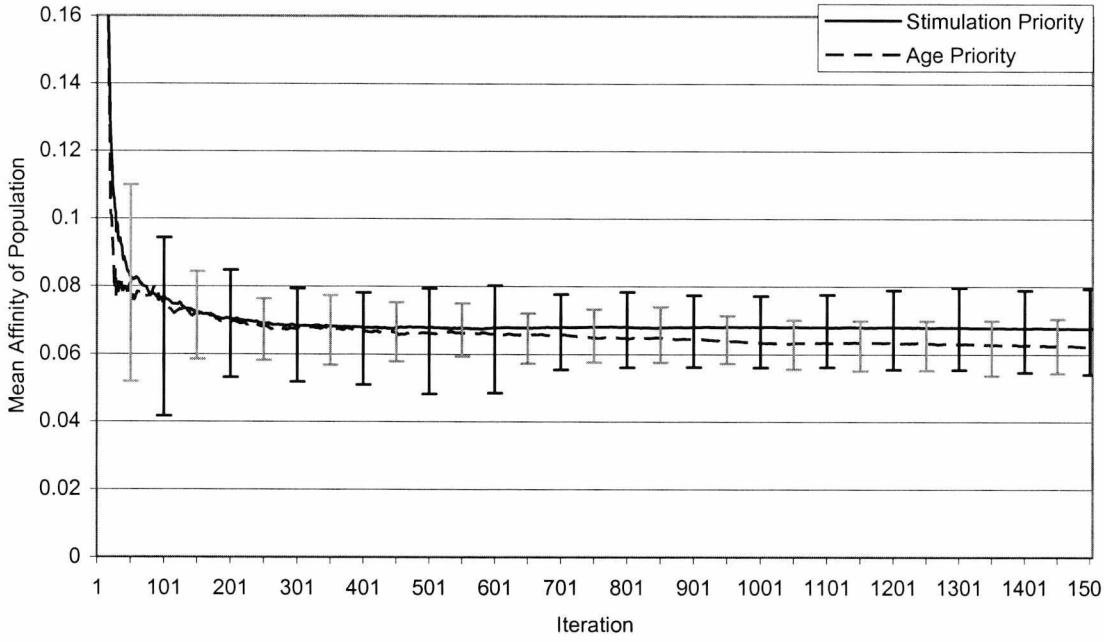

Figure 6.4. Chart showing variation in mean population affinity for two different population sorting methods.

The chart in Figure 6.4 shows the results of this investigation. This chart was constructed using the mean affinity of the population, the process by which this is determined was discussed in section 6.1.1. Error bars alternate between series with those for age priority being shown in gray and for the sake of clarity they have been placed only every 50 iterations. A successful outcome of this test would show that the chosen strategy of population ordering results in a greater mean affinity of the population compared with age priority.

It can be seen that the stimulation priority ordering of the population does indeed result in a slightly higher average affinity of the population in general. Likewise, it can

be seen in Figure 6.5 that the mean interestingness of the discovered web pages is also slightly higher throughout the run of the system. For both series the error bars show the difference around the mean for the age priority technique is smaller than the stimulation priority technique.

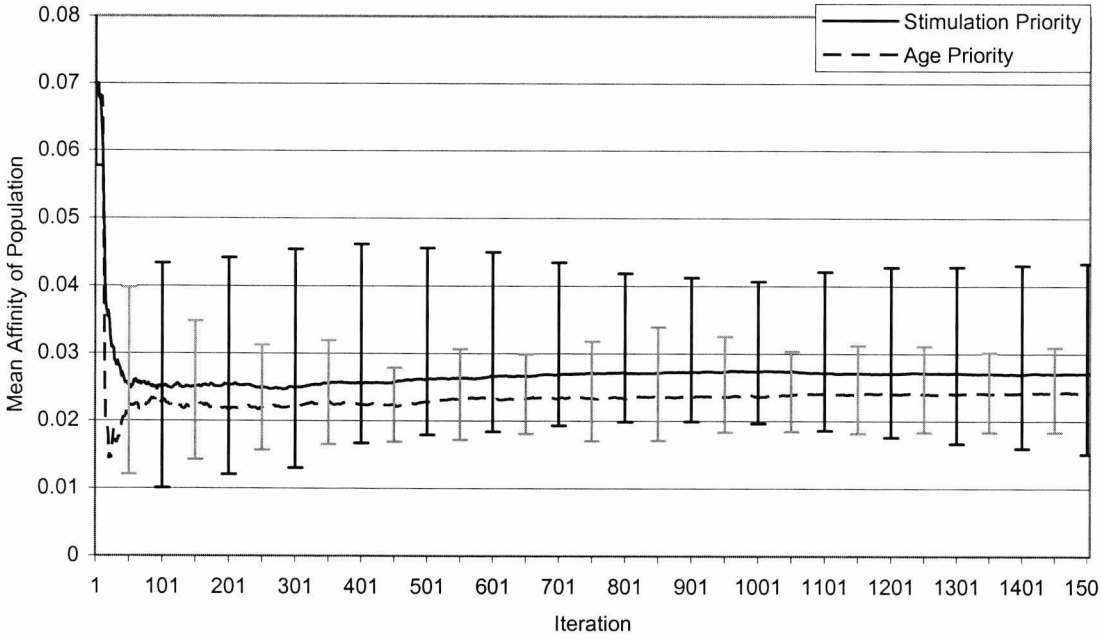

Figure 6.5. Chart showing variation in mean population interestingness for two different population sorting methods.

The two series can be tested to determine if the greater observed mean affinity and interestingness of the stimulation priority technique is significantly higher then that of the age priority technique or if the null hypothesis, that the two strategies result in the same mean, cannot be rejected. This can be done in two ways. Either the entire series can be used, or the *final* value of each series can be used. In the case of the final value, the mean of 30 independent observations is used. Using the Student's t-test as the number of observations was small (Appendix C), it was found that the difference in affinity between the two ordering techniques was statistically highly significant (P_{null} = 0.002) while the difference in interestingness was statistically significant (P_{null} = 0.03). Using the entire series and the large sample test using the normal distribution as the number of observations is large, the difference for both the affinity and interestingness is found to be highly significant ($P_{null} < 0.01$ for both).

The sharp dip in mean interestingness for the population when prioritised by age around iteration 25 cannot be easily explained. It is at this point that the last of the initial population will be moving from the initial pages. As the population is ordered by age, all cells will be making this move one after another. This is significant as it relates

to a point where every member of the population is 1 or possibly 2 links away from the initial start pages. It follows that the pages at this position may therefore be particularly uninteresting. Close to the start pages they may indeed not contain much novel or unexpected content. This dip does not show up with the stimulation ordering as there is much more variation in the position of the cells relative to the start pages. However it should be said this explanation is only really a conjecture.

6.1.4 Observations During Running

This section examines the characteristics of the algorithm regarding the way the cells spread out from the initial start pages. These sections use the notion of graph distance. Recall that the web can be seen as an undirected graph[5] with nodes representing pages and edges representing hyperlinks between those pages, the reader is referred back to section 5.6 for a visualisation of this. If nodes (pages) are a distance of 1 away from each other then only 1 edge (hyperlink) must be traversed to go from one to the other. When multiple routes are available between two particular nodes then the distance between them is the shortest distance available. All measurements are made from the set of initial start pages, on which the initial set of cells begins its search. Therefore, these are by definition distance 0 from the centre of the graph.

Investigations were undertaken into the characteristics with which the cells spread from their initial starting pages. Two strategies were compared; these were the age priority and stimulation priority population orderings as described in section 6.1.3. It was thought that when cells are ordered by stimulation level, the radius of the graph will grow quickly in comparison to the age priority approach. This is because the most stimulated cell will be selected to move, then it, and possibly its offspring, are likely to be selected to move again as it is likely that that they have high levels of stimulation.

The age priority method of population ordering was used for the purposes of a comparison. For this method it was expected that the graph radius would not grow so fast as each cell is moved in turn. Thus only once each cell has been moved 1 link away from the start pages then cells start to move 2 pages away and so on. It is not quite this straight forward because offspring cells are always moved one link away from their parents, and some cells can find paths that bring them closer to the start pages again, but generally cells move away from the stating pages. The expected result is that the cells

---

[5] Technically the web is a directed graph as hyperlinks point in one direction only, but this has been ignored for simplicity.

for the stimulation priority method are moving away from the central initial pages much faster than exhibited by the age priority process.

The results from this test can be seen in Figure 6.6. The chart shown is the mean of 30 independent runs as before. This chart shows the maximum graph distance at each generation. Error bars are again plotted alternating between series every 50 iterations again for the sake of clarity. The error bars relating to the age priority series are shown in grey.

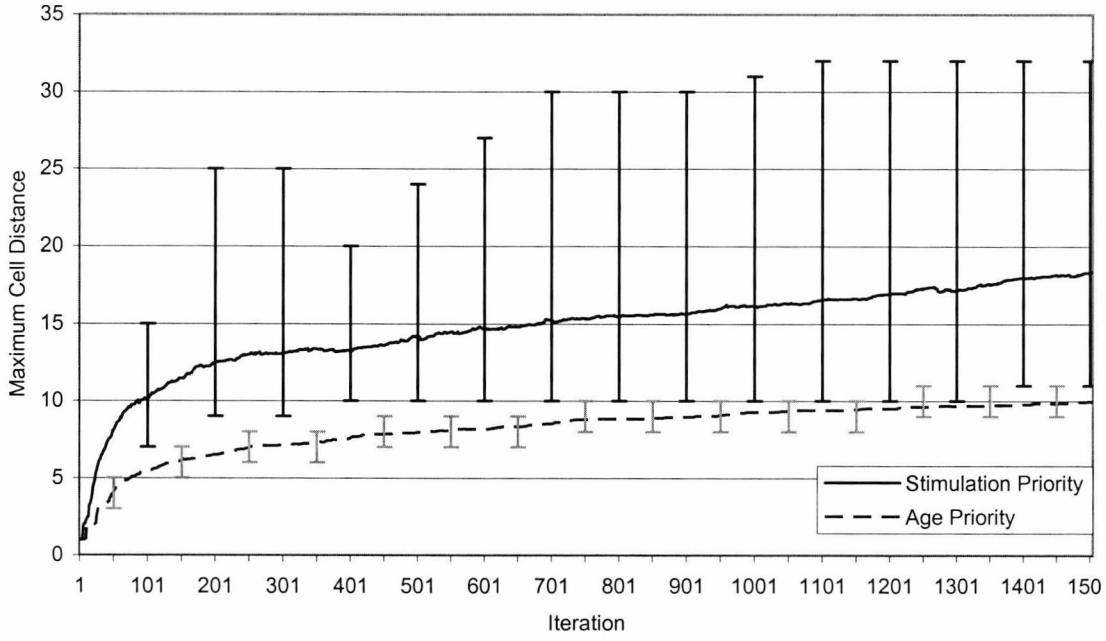

Figure 6.6. Chart showing the maximum graph distance with increasing iteration number for two population ordering strategies

It can be seen from the chart Figure 6.6 that the behaviour observed is what was expected. The stimulation ordering technique does indeed result in the maximum graph radius increasing quickly as expected. It is interesting to note the stepping effect between iterations 0 and 50 for the age priority series. This is when all cells are being moved 1 hyperlink away before they can begin to be moved 2 links away and so on, as predicted above. However, from mean graph radius = 3 and above, this stepping effect is less pronounced as the population becomes more diverse. The error bars reveal that the maximum radius of the graph is very variable for the stimulation priority method. This stands to reason as there will be occurrences when the most stimulated cell is far from the start pages and it happens to find an interesting area of the web. This cell will continually be stimulated continually increasing the size of the graph. The other extreme is possible and no cell will dominate the search. This results in more cells being moved one after the other rather then a single cell being moved in successive iterations. In this

case the maximum radius will increase much more slowly and this can be seen in the chart as the minimum bars for the stimulation priority series are very close to the age priority mean for the right hand third of the chart.

A second chart was drawn with the mean node (page) distance plotted. This was computed by determining the sum of the distances between each page found up to and including the current iteration, and dividing by the total number of pages found. This was plotted in to complement the chart above (Figure 6.6) as this only shows the maximum radius and so single or small number of cells could have been making their way increasingly far from the start pages. This would be a bad situation in terms of search, as it is better to have numerous cells looking for information rather than a small number. A chart showing the average distance should look similar to that of the maximum distance if the cells truly are spreading as expected. If the bad situation described above is happening, the line of mean node radius when plotted on a chart would look significantly flatter than the line for the maximum distance, and it would also be significantly lower than the chart plotted for a mean. (i.e. a the value at a given point on Figure 6.6 would be significantly higher than the value at the same point on Figure 6.7). The resultant chart is shown in Figure 6.7 and was constructed using the same 30 runs of AISIID used to plot Figure 6.6.

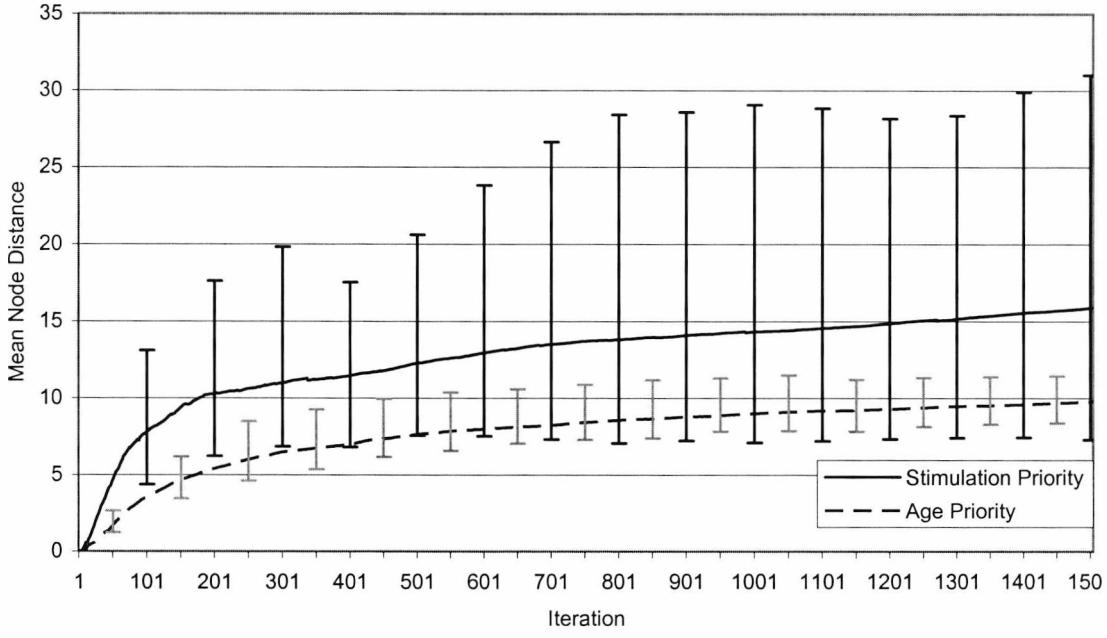

Figure 6.7. Chart showing the mean node distance with increasing iteration number for two population ordering strategies

The results in Figure 6.7 mirror Figure 6.6 in that there is a rapid increase in mean node distance for small iterations numbers, with gradually increasing mean node

distance as the number of iterations increase. This shows that no single or small group of cells are dominating the exploration effort of the cells and generally the cell population is spreading out from the central starting pages. The age priority ordering system was again seen to result in a population that spread less far from the initial starting pages compared with the stimulation ordering. Again the error bars show the deviation about the mean for the age priority situation was small in relation to the stimulation priority.

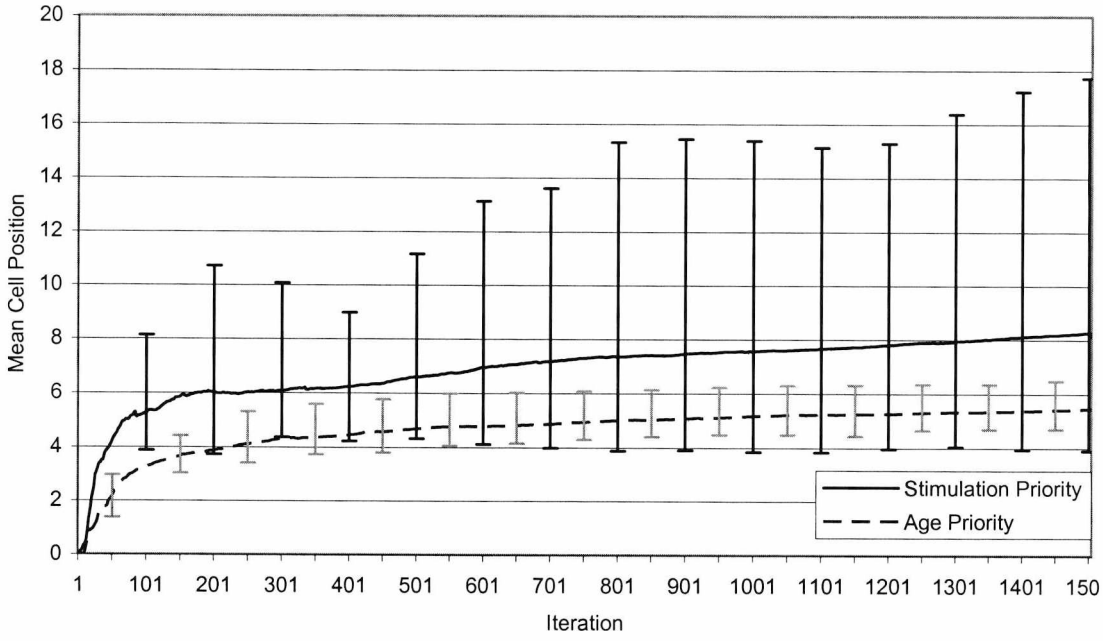

Figure 6.8. Chart showing the mean cell position with increasing iteration number for two population ordering strategies

To complement this, one further test was undertaken. This used the same data as the previous tests but analyses the mean cell position, as opposed to the node centric measurements in Figure 6.6 and Figure 6.7. In this case at every iteration it is possible to query every cell in the population and determine how far it is from the starting pages. From the sum of these distances it is possible to determine how far away the average cell is. Thus, if the average cell distance is constantly low, it would indicate that very few cells are spreading out from the centre point, the cells are crowding around the start pages. On the other hand, a constantly high mean cell position would indicate that the cells are moving quickly away from the start pages, with the cells generally being found far away from the central pages, on the edges of the search, with few cells close to the central pages. The results for this are shown in Figure 6.8.

The overall shape of the stimulation priority series is again similar to the shapes of the series in Figure 6.6 and Figure 6.7. The shape suggests that the cells move quickly

away from the starting pages, but this movement becomes more noticeable as the system continues. It is possible to see a small decrease in the mean node position at around iteration 75 when using the stimulation ordering strategy. It is expected that a parent had a number of offspring close to an uninteresting part of the web but at a distance from the start pages on the web greater than the mean cell position. As they all have a similar stimulation level (as they are all new) they would have been examined and moved by the algorithm in quick succession. As the area they were in was not interesting each would have been removed in succession. Therefore, at iteration ≈75 the mean node position decreases quite quickly.

It can also be seen that the mean cell position for most of the graph, especially for higher iteration numbers, is approximately half that of the maximum graph distance Figure 6.6. This can be viewed as a positive result as it shows the cells are not congregating around the starting pages, nor are they generally at the edges of the known graph. The result suggests that the cells are more likely to be spread throughout the full range of the graph.

6.2 Observations Upon Termination

Upon termination it is possible to analyse the results and determine the relationship between mean affinity or interestingness of a web page with respect to its distance from the start pages. At the end of each run, for each page found by the system, the distance from the starting page and the average affinity with each cell that found was recorded.

To begin, a histogram may be drawn showing, for each distance from the start pages, the number of pages found at that distance. This is shown in Figure 6.9. The source data was again generated by 30 independent runs of AISIID, the stopping criterion again being a total of 1,000 pages found. Therefore 30,000 observations are used in the generation of the histogram in Figure 6.9.

Figure 6.9 shows a peak at a distance of 5, therefore most pages are found at a distance of 5 hyperlinks from the start pages. While it could be assumed that the peak in the histogram should be at a distance value less than this, the result can be explained thus. When the initial population of cells are released they will produce offspring. As these offspring are always placed on a page further away from the start pages than the parent cell then there is a constant drive away from the start pages. This takes the mean position of the population away from the start pages to a certain extent but the behaviour results from interaction with this and another characteristic. That is, the stimulation priority of the population ordering mechanism. This tends to result in a

small group of cells searching a small area of the search space until they begin to examine to uninteresting pages, at which point their stimulation will drop below a critical level, and other cells begin to search a different area. These cells situated around 5 pages from the initial pages are those that are in waiting for the time when the stimulation level of the cells involved in each niche search begin to reduce significantly. At this point they will take over. These cells cover a very wide area of the search space at this point as they are likely to have spent many iterations on a page without being moved. As there can only be $K_{suppress}$ cells on a single page before their stimulation level begins to be reduced and there are so many cells around a distance of 5 nodes away from the starting pages, they must be well spread around this area. This is a useful place in the search space for such behaviour to take place as it is believed to be close enough to the starting pages such that the original concepts are still strong, while being far enough away to maintain variation in subjects on which the cells are sited.

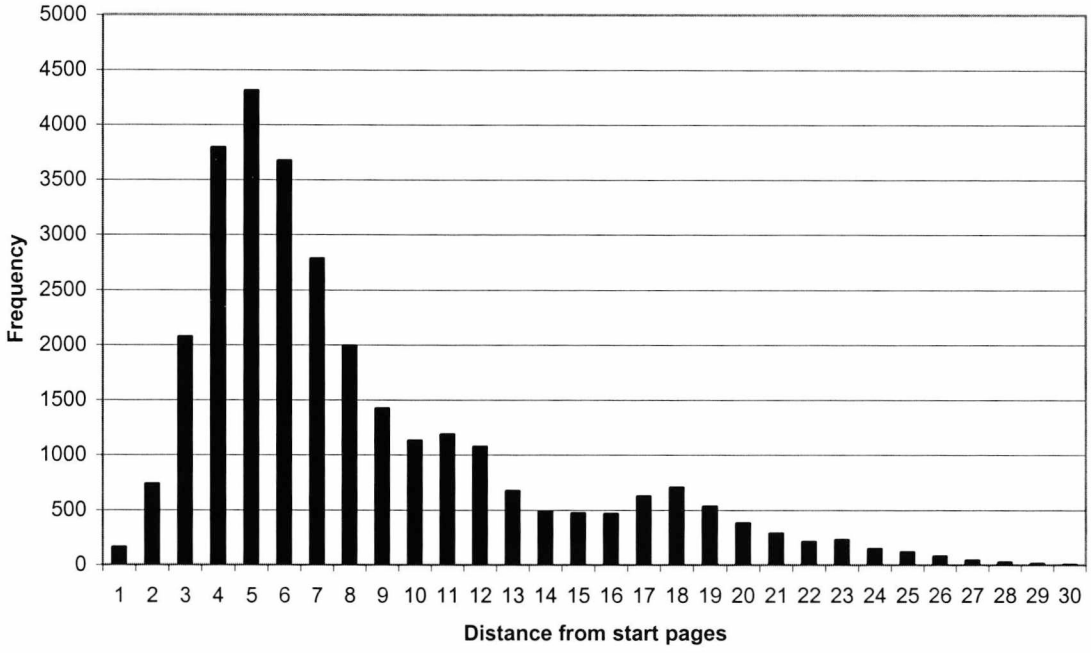

Figure 6.9. Histogram showing relationship between distance of page from start pages and number of cells finding pages at that distance.

It is possible to plot a chart to visualise the relationship between graph distance from the start pages and the mean affinity at that distance. When AISIID finishes a single run, the URL of each page found is printed along with the mean affinity over all cells that found that page and that page's distance from the starting page. It would be reasonable to assume that the average affinity measured will decrease as the distance from the starting pages increases. It is also reasonable to assume that while the concept may be strong within a few hyperlinks of the start pages, as more and more hyperlinks are

followed the concentration of the concept on individual pages will become increasingly weaker. The results of this test are shown in Figure 6.10. In this figure both the mean affinity and the mean interestingness have been plotted.

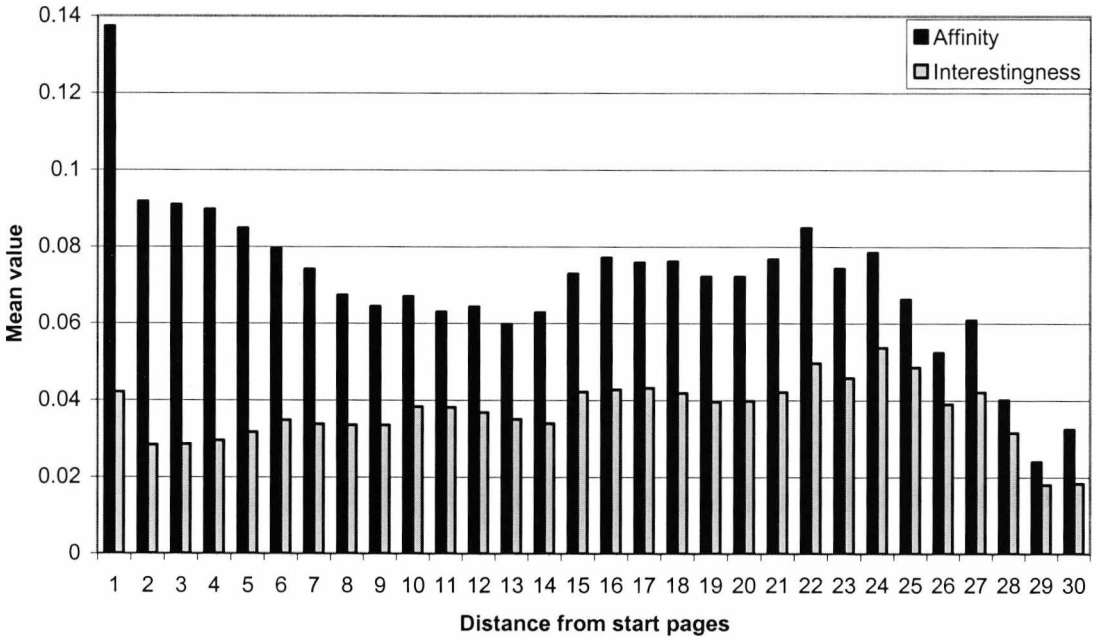

Figure 6.10. Changing mean affinity and mean interestingness with distance from start pages.

It can be seen that pages one hyperlink away from the start pages have exceptionally high affinities with cells compared with those that are 2 or more hyperlinks away. This stands to reason as the hyperlinks on the start pages are highly likely to point to pages which are highly relevant. Yet in contrast, it can be seen that that the extremely high mean affinity at a distance of 1 does not necessarily mean that the pages just one hyperlink away are interesting. It may be reasoned that pages this far away are not likely to be novel or surprising to the user as they are so closely related to the starting pages.

While the affinity of pages at distance of 1 tend to be high, the bar representing the interestingness component appears to be close to the average. Indeed, one of the interesting insights this chart allows is that, between distance = 22 and distance = 25, there are 4 distance values with mean interestingness figures greater than the value for distance = 1, whilst there are none for between distance = 2 and distance = 21. So in this experiment, the pages with the highest component of interestingness are located over 20 hyperlinks from the start page.

Contrary to what may be suggested by Figure 6.10, when the results are sorted in descending order of average affinity, the standard procedure for output from AISIID, the result is that only 3 of the top 20 pages are of distance 1 away from the start pages.

The question may be asked that, given pages at distance 1 have on average a high affinity then why doesn't the user see just pages separated from his or her own start pages by just one hyperlink? It can be explained that this ranking is made over individual page affinity while the chart above concerns the average affinity at a distance, and is thus misleading regarding the way the pages are actually used. A high affinity on average at a particular distance does not guarantee that the best page at that distance is any better than the average at that distance. This effect can be seen when error bars are plotted on an amended version of Figure 6.10. In Figure 6.11 the minimum and maximum affinities found at each distance have been shown.

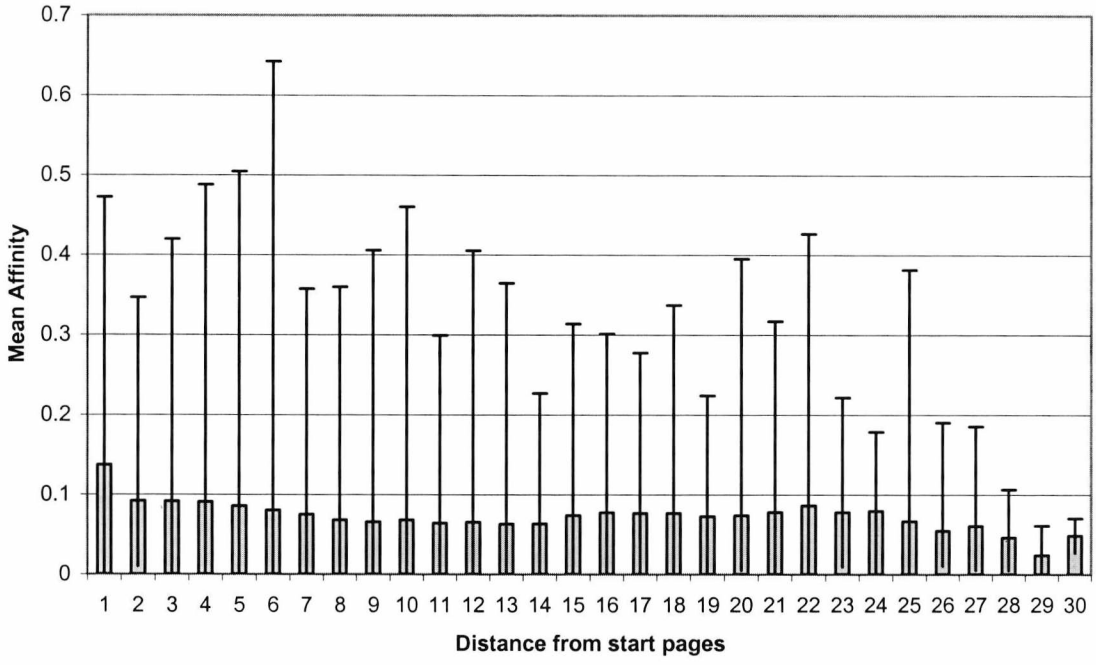

Figure 6.11. Mean affinity of pages at different distances from the start pages (bars) with maximum affinity of these pages shown as lines.

What can be seen from Figure 6.11 is that a high average affinity does not necessarily mean a high maximum affinity for a given distance. By inspection it can be seen that at least three distances have pages with a maximum affinity greater than the maximum affinity of a page at distance 1. This is a positive result as information on pages just 1 hyperlink away from the pages supplied is more likely to be known than that on pages 20 hyperlinks removed. Therefore the pages 20 hyperlinks away may be more interesting for the user as they contain novel information, they may also contain increased novel or surprising information as measured by the algorithm, increasing the interestingness component of the affinity function and therefore the affinity between that page and a cell. It is therefore possible to reason why a high mean affinity for pages

at a given distance does not necessarily result in an overabundance of pages at that distance being returned to the user.

6.2.1 Summary of Objective Tests

To summarise this section, a number of tests were performed to determine whether AISIID was performing as expected. The population control measures were seen to be performing much as expected and the decision to order the population by stimulation was also seen to produce results as expected. When running the cells were found to spread quickly away from the user's start pages and, upon termination it was found that interesting pages were identified many links away from the population's initial starting point. However, these tests do not give a value as to the actual quality of the results, as decided by a user, it is these subjective user opinions that this chapter is concerned with next.

For consistency only one set of start pages was used throughout these tests, and it is acknowledged that this may impact the extent to which these results will generalise. However, the tests were each run a reasonable number of times and where appropriate Student's t-test (Appendix C) was used for statistical tests where necessary, so it is believed that the conclusions drawn will also hold when run on other data. Future work would involve testing the generality of the technique by running the system on other starting pages sets.

6.3 Subjective Tests

The previous section of this chapter dealt with the quality and behaviour of the algorithm. However, as stated in the introduction to this chapter, an objective investigation into the perceived interestingness of retrieved pages is not feasible but AISIID can be tested against a comparison system in a subjective manner. The most natural way to test the quality of AISIID's output is to compare it with the system of Liu et al., called WebCompare, a situation that was not available when web compare was developed: "*There is also no existing system that is able to perform our task. Thus, we could not do a comparison*" (Liu et al., 2001).

The output of AISIID is, by its very nature, subjective. The most revealing way to test the output directly therefore is with a user study. Generating search results for a user using both AISIID and WebCompare then asking a user to assign scores to the results is the only way to test the quality of the output in terms of perceived interestingness. In addition to generating results with WebCompare, the search engine Google is used to

also retrieve results to give a baseline for comparison of the two other systems. Google is chosen as a baseline as Google does not explicitly try to maximise interestingness of a page and it was thought that comparing AISIID against a well established search engine may reveal interesting results.

It should be noted that the aim if this test is to determine the relative scoring between AISIID and the comparison system WebCompare, not make comment on the absolute quality of the pages retrieved. The absolute scores obtained by each system from one run to the next are too highly dependent on external factors such as the quality of the initial pages supplied by the users, the subject of the initial pages, the mood and expectations of the user and so on. It is too hard to keep these factors consistent from one run to the next to be meaningful in this context, and so the relative qualities of each system are scrutinised to allow the external factors can be negated as much as possible.

In the user study that follows, AISIID is compared against WebCompare and the well known search engine, Google. The following two sections describe how comparison pages may be generated with WebCompare and with Google respectively. These are brought together in section 6.3.3 in which information is given as to how AISIID generates results for individual users and how the results from the three systems are shown to a user.

6.3.1 Generating Comparison Pages with WebCompare

When assessing the performance of a new system, it is of course, vital to be able to compare the performance of that system with another designed to perform the same (or as similar as possible) task. In this case the only logical comparison system is that of Liu et. al. as described in the paper (Liu et al., 2001). In the paper, Liu et al. evaluate their system by recruiting just three users from three different organisations. Each user was asked to browse a competitor's website then WebCompare was run over that website. The users were shown the results and asked for comments regarding the quality of the results obtained. However, only three users were used to assess the output compared to many more than that in this investigation. It should be noted that, unlike the report in this chapter, the single test used here - "finding unexpected pages" - was not used in isolation. Indeed, users were shown unexpected pages, unexpected keywords/concepts and unexpected outgoing links. The reported comments were positive towards the WebCompare system, with some useful observations being reported. These included opinions such as the system allowed a user to browse more deeply into the site rather than becoming impatient and stopping browsing on high-level

pages, or similarly allowing the summarization of long pages with keywords. Thus, users who would otherwise grow impatient with a long page were prompted to read it in detail as they then had the motivation to spend time. Other than the user comments, no more in-depth analysis of the results was performed. No user was asked to rank or give a score to a page, keyword or outgoing link and thus no correlations were reported between the rankings given by WebCompare and the user opinion.

While the paper by Liu et. al. does describe five separate metrics, only one is relevant to this investigation, that of "*Finding unexpected pages in C* [unknown or competitor website] *with respect to U* [user website]". For further information on the WebCompare algorithm, the reader is referred back to Chapter 2 and (Liu et al., 2001), which contains an overview of the full set of WebCompare techniques. Three equations from that chapter are repeated below for ease of reading.

This particular technique assigns a score to each word related to its unexpectedness. It then determines the mean unexpectedness score over all words in a document (webpage). The pages can then be ranked according to the score given with the highest scoring page being considered the most interesting. To begin, all competitor pages are combined into to a single document, *C*, so too are all the user pages, *U*. The weights of all words in the set $C \cap U$ are determined and the mean word weight for all pages in *C* is computed. The mean score for each page is computed using Equation 6.3 below. In the following equations the unexpectedness score of each word is computed using Equation 6.2, where $tf_{r,i}$ is the normalised term frequency of the *r*th word or feature in document *i*. All calculations involving term frequency use a normalised term frequency computed as shown in Equation 6.1. This is a reasonable step, as the *C* and *U* documents may be of greatly varying sizes rendering calculations based on absolute frequency subject to inaccuracy.

$$tf_{i,j} = \frac{f_{i,j}}{\max f_{i,j}} \qquad \textbf{6.1}$$

$$\text{unexpT}_{r,i,t} = \begin{cases} 1 - \dfrac{tf_{r,j}}{tf_{r,i}} & \text{if} \quad \dfrac{tf_{r,j}}{tf_{r,i}} \leq 1 \\ 0 \ \text{Otherwise} \end{cases} \qquad \textbf{6.2}$$

The sum of unexpectedness scores for every term appearing on a page is then computed and normalised by the number of words in the document to give the final score, as shown in Equation 6.3.

$$\mathrm{unexpP}_i = \frac{\sum_{r=1}^{m} \mathrm{unexpT}_{r,c,u}}{m} \qquad 6.3$$

This particular metric, "*Finding unexpected pages in C* [unknown or competitor website] *with respect to U* [user website]" was re-implemented as per the procedure stated in (Liu et al., 2001) in Java along with the required auxiliary function "finding unexpected terms in a *C* page with respect to a *U* page". The input to this implementation was integrated into the output procedures from AISIID in the following manner.

The WebCompare algorithm requires a pre-spidered set of pages to be specified as the "competitor" pages. Once again this runs into the issue of just how many pages to retrieve for ranking. Since the goal of the experiments reported in this section is to perform a controlled comparison between AISIID and WebCompare, a straightforward and fair solution is available. The set of "competitor pages" (set *C*) is taken as the set of *all* web pages that have been encountered by AISIID during a run, and have therefore been available for AISIID to rank. Thus both AISIID and Page Compare will see and evaluate exactly the same set of documents. The "user pages" (set *U*) are those pages specified by the user as initialisation pages (K_{train}). The procedure for assigning a score to each competitor page proceeds as described above which allows the set of pages to be ranked in descending numerical order. Further explanation regarding how these pages are shown to the user is reserved until Section 6.3.3.

6.3.2 Generating Comparison Pages with Google

The search engine Google (Google, 2006b) is employed to provide one set of results for each user test. The interestingness of these Google pages, as scored by the user, is used then as a baseline for the quality of output from AISIID. That is, AISIID should provide users with pages that are, on average, of greater interestingness than Google for a specific search. This is because Google only attempts to retrieve the most relevant pages. This is, therefore, a good test of the ability of the AISIID system to discover interesting information.

For each user it is necessary to generate a set of results using inputs to Google closely mimicking the way input into AISIID is handled. Google does have a facility with which a user can specify a web page, rather than a set of keywords, as an input into a search. Google will then try and retrieve pages that are similar to the page given. From

the Google website: "*If you are interested in researching a particular field, Similar Pages can help you find a large number of resources very quickly, without having to worry about selecting the right keywords.*" The exact algorithm used for determining page similarity is undisclosed, but the Google website suggests that pages are considered similar if the language used on the pages is similar and the targets of hyperlinks found on those pages point to similar target pages (Google, 2006a). The "similar page" search is initiated using the syntax "`related:www.website.org/index.html`" in the standard Google search box. In this case Google will present results where each result is a page similar in content to the page "index.htm" from the site "website.org".

The way in which results are generated using Google should be as closely matched as possible to the way results are generated using AISIID and WebCompare. In particular, the entire set of pages supplied by the user must be used to generate the output. In this case, each user-specified page is used in turn to generate a set of results from Google. The total number of results required from Google can be divided up equally such that an equal number of results are taken from each search to be shown to the user. For example if the user specifies 5 pages and the top 20 results are to be included in the user ranking, then the *highest ranked* 4 search results generated from each user page are included. If the number of pages supplied is not divisible by 20 then any remainder is made up randomly from the remaining results. Likewise if a related search does not produce enough results or some results are invalid, then pages from the other related searches are used as necessary. Taking the highest ranked page that has not been included and so on. If a link to a page is invalid or a link is given by Google that is actually the page submitted to the related search (a common occurrence) then again, the next highest ranked page is used.

6.3.3 Experimental Protocol

In order to determine the relative interestingness of the web pages returned by AISIID, WebCompare and Google, users were recruited to take part in a user study. Each user was asked to supply a small number of URLs (typically around 5, but more was quite acceptable) referring to pages found on the web which they considered summarise their knowledge on a particular subject. Therefore each user in the study is associated with a completely separate set of pages from all others. These pages were used as the user defined starting pages for AISIID and the user pages for WebCompare. Pages with a

high number of hyperlinks were encouraged; this would give AISIID more chance of finding interesting pages as it increases the diversity of the search space somewhat.

It was found that, when left overnight, AISIID could be expected to retrieve 7,500 pages (although this was dependent upon the speed of access to the websites pages were retrieved from). This figure of 7,500 therefore gives this process at least some grounding in reality, in that it is the number of pages likely to be retrieved in a practical scenario where the user would leave this system running overnight. AISIID is run 3 times, each run retrieving 2,500 pages. Using multiple runs is standard practise when using non-deterministic algorithms such as this, and preliminary tests showed that 2,500 pages gave a reasonable trade off between depth of search and number of times the algorithm is run. As the results of the three runs are merged then sorted according to interestingness, this strategy is the same as the one typically employed when performing optimisation with evolutionary algorithms or even data mining using parallel AIS (Watkins, 2005). The specifics of how many pages are retrieved and how many runs are undertaken are thought to be relatively unimportant as this investigation only attempts to compare AISIID and WebCompare in relation to each other, it is not an investigation into the absolute scores allocated to webpages. As long as WebCompare and AISIID score the same sets of pages and the sets of pages are merged in the same way to keep the test fair, the manner in which the pages are discovered is of secondary importance.

When three runs had been completed, AISIID output a list of the URLs of all pages encountered at least once by at least one cell along with some other information about that page and the cells that found it. The mean affinity of each page with the cells that found it was used to rank the list of URLs found by AISIID in descending numerical order. All URLs found by AISIID over the three runs were used as inputs to WebCompare as the competitor set. The set of user pages submitted to WebCompare was the set of URLs submitted by the user to provide the starting pages for AISIID. The page unexpectedness score generated by WebCompare was used to sort the pages into descending numerical order. Note that when running, AISIID recorded to disk a representation of each page exactly as retrieved. This locally stored version was used when WebCompare was run to eliminate the possibility that the page has changed between the runs of AISIID and WebCompare ensuring fairness in the test.

6.3.3.1 User Test Setup

The 3 sets of results from AISIID are merged into one list, and the 3 sets of results from WebCompare are merged into another list. The top 20 URLs as ranked by AISIID, the

top 20 URLs as ranked by WebCompare and the top 20 URLs as ranked by Google (using the protocol described in the previous section) are the URLs to be seen by the user. Any duplicates within each list are removed as the same page could be visited on more than one of the runs. So too were any of the user defined starting pages. While these are not usually output, it certainly is possible for a cell to find a way through a chain of hyperlinks and discover this page.

In order to score the individual URLs, these 60 links (three sets of 20) were placed into a Microsoft Excel spreadsheet, along with a marker indicating the source of that URL. The URL was turned into a hyperlink in a separate column for ease of use for the user, and their order randomised and all columns other than the hyperlink were hidden so users would have no idea which URL was retrieved by which system. Beside each URL was a space for the user to score the page. Validation was set up to ensure the user could only input an integer between 0 and 10 to score the page. Upon clicking a hyperlink in the spreadsheet, a separate browser window would appear, allowing the user to view the webpage. The users could fill in their scores in any order they wished and there was no time limit.

Users were asked to rank each page from 0 – 10 where 0 represented a totally uninteresting page, while a score of 10 was reserved for a page that was exceptionally interesting. The exact wording of the instructions for users was:

Please rank each hyperlink for interestingness using a number between 0 and 10.

At no point in the instructions to the user was interestingness defined (no mention of novelty, surprisingness, etc.) and this point is found to be important later on.

6.3.4 User Test Results

A total of 15 users agreed to participate in the test. Note that this number of users was considerably larger then the number of users recruited to test WebCompare in (Liu et al., 2001) where only three users evaluated the system. Table 6.3 summarises the user's individual search topics and full list of URLs submitted by each user is published in Appendix E.

Table 6.4 shows the results of the tests for each individual user. For each system, the mean value of the subjective interestingness scores assigned by each user to the 20 pages returned by each system is shown along with the standard deviation of the associated value. The final row shows the mean of all scores. The results of this table

are visualised in Figure 6.12. A successful outcome for this test would show AISIID has a higher user score when averaged over all users compared with Google and WebCompare.

User	Subject
User1	Bioinformatics
User2	IPTV
User3	Areas of mathematical shapes
User4	Graph drawing
User5	Trans-membrane proteins
User6	Markov chains
User7	Java OpenGL
User8	Swarm intelligence
User9	Prokofiev (Russian composer)
User10	Antigravity
User11	Montessori schooling
User12	World of Warcraft computer game
User13	Neverwinter Nights computer game
User14	Extreme unicycling
User15	Star Formation

Table 6.3. Summary of test users and their subjects

	AISIID		Google		WebCompare	
User ID	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
User1	2.30	2.64	4.30	3.06	0.90	1.37
User2	2.50	3.03	4.65	2.87	0.00	0.00
User3	0.42	0.51	2.84	3.81	0.11	0.45
User4	1.70	2.52	4.25	3.78	0.00	0.00
User5	8.20	1.79	3.65	2.60	2.50	3.30
User6	0.25	0.55	0.85	1.46	0.10	0.45
User7	4.95	3.52	4.10	3.35	1.10	0.31
User8	2.00	2.22	2.71	3.05	0.11	0.31
User9	3.55	3.61	2.35	4.09	0.30	0.57
User10	1.55	2.39	2.85	3.96	0.00	0.00
User11	3.70	3.97	9.00	2.47	1.50	1.15
User12	2.65	3.36	2.68	3.70	0.00	0.00
User13	6.95	3.69	7.85	3.08	1.47	1.81
User14	2.95	3.40	3.70	2.87	0.00	0.00
User15	4.90	2.27	4.75	2.49	1.30	1.53
Mean	**3.24**	**2.23**	**4.04**	**2.07**	**0.63**	**0.79**

Table 6.4. Mean subjective interestingness scores for AISIID user tests

AISIID was found to have a mean score of 3.24 over all the tests, while Google achieved a higher mean score of 4.04, whilst WebCompare achieved a surprisingly low mean score of just 0.63. As such, it was a surprise that Google scored higher than the other two but very positive that AISIID scored higher than WebCompare. In addition, in not one single test did WebCompare score higher on average than AISIID, which is a positive result. These absolute scores were found to vary greatly, from user to user as can be seen in the table above. This can be attributed to a number of factors including mood of the user, expectations of the user, the subject selected by the user and the

quality of the initial set of pages. For this reason, any meaningful interpretation of the absolute scores is unavailable. However it was seen that some were more pleased with the results than others as will be discussed in the following section.

Figure 6.12 shows a bar chart summarising the results in Table 6.4. Each bar represents the mean score for that system. High/Low bars are included to aid understanding of the results, these bars show the minimum and maximum scores obtained by each system. It can be seen from the chart, Figure 6.12, that over the 15 tests, AISIID has scored, on average, considerably higher than WebCompare. The error bars reveal that the maximum value scored by AISIID was really quite high, while even the maximum score for WebCompare was low in comparison.

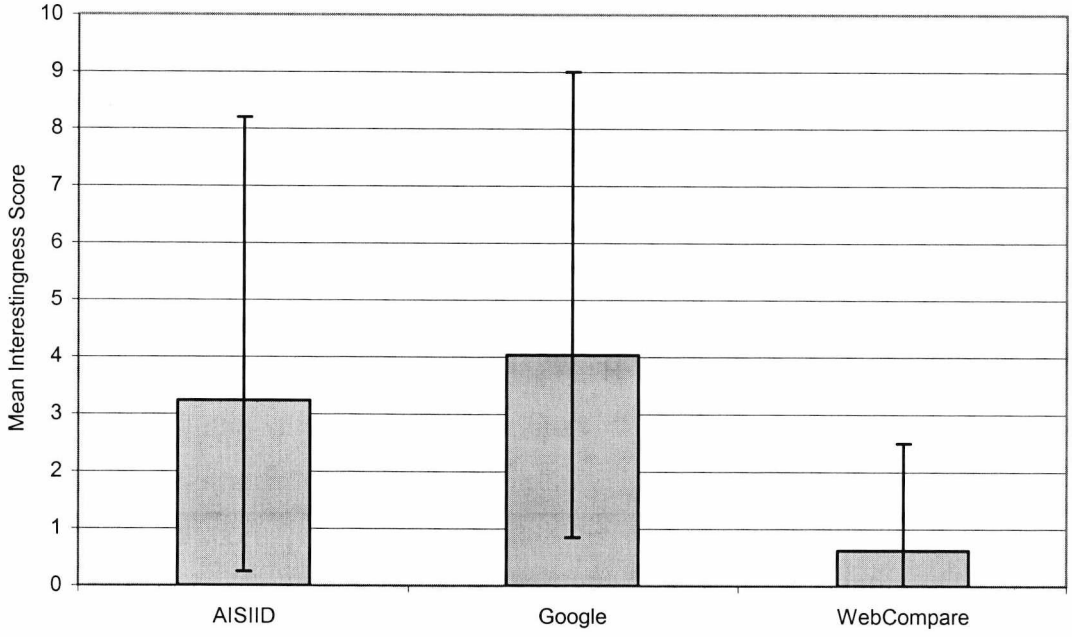

Figure 6.12. Chart showing mean scores for each system under test

It is important at this juncture to determine whether the ratings given to the pages retrieved by AISIID are statistically better than those retrieved by WebCompare. Student's t-test (Appendix C) can be used to determine whether AISIID scores significantly better or not. The t-test is especially suited for situations where only small sample sizes are available (typically fewer than 30 observations (Creighton, 1994)) and as such is particularly suited to the results of these experiments where 15 observations have been made. The number of observations made changes the shape of the t-distribution and as such this test can take account of such small samples.

A one-tailed test is in order here as the aim is to find out if the true mean of the WebCompare observations is less than or equal to the true mean of the observations for

AISIID. The threshold for significance is set at $P_{null}<0.05$ while values of $P_{null}<0.01$ are thought to be highly significant and the null hypothesis is that the means of the observed scored given to AISIID and WebCompare do not differ. The probability of the null hypothesis holding was found to be less than the threshold for significance (P_{null} = 0.0002) which indicates the mean subjective interestingness score for the pages retrieved by AISIID is significantly higher than that of WebCompare. This came as a surprise as the scores from test to test varied quite dramatically and validates the superior quality of AISIID by comparison with WebCompare.

It is clear to see that Google has obtained a higher mean score for subjective interestingness than either of the two other systems. Although in 4 tests the mean score given to AISIID is higher than the mean score given to Google and the Student's t-test reveals that the difference between AISIID's results and Google's results is not statistically significant (P_{null} = 0.16). Nevertheless, the difference in the means came as a surprise as Google was only thought to be retrieving relevant pages whereas users were asked to rate pages in terms of interestingness. This result is significant in terms of contribution to the research area, but further expansion on this is reserved for the next chapter.

6.3.4.1 Assessment of WebCompare

Some thought must be given to the surprisingly low score of WebCompare. It is thought that WebCompare does not fare well for the combination of reasons.

Firstly, it has no direct measure of relevance. The WebCompare system makes a strong assumption regarding the competitor's website. It is assumed that every page on a competitor's website will be relevant to the user's search just because it is on that competitor's website. This is unlikely to be the case as no website will ever contain 100% relevant content. So while a page on a competitor's site may contain numerous words with a high unexpectedness score, if that page is on a completely different topic to that which the user is searching for, the high surprisingness of that page is negated.

In addition to this, it is believed that WebCompare is susceptible to noise on individual web pages, as term frequency is the only characteristic of a page that is used. Inverse document frequency is not used to ascertain the relative importance of a particular word to that specific search. In the paper, the authors note that in the unexpectedness calculation, inverse document frequency numerically cancels, but no equivalent metric is introduced to ascertain the relevance if words to a search, reducing the quality of the result.

Finally, WebCompare uses the mean unexpectedness value over all terms on a page, and as a result shorter pages are found to be favoured over longer ones. Thus the users are not generally returned pages rich in content by WebCompare. Instead WebCompare seems to prize shorter pages such as navigation pages where the unexpected words are common on the page with little else in terms of content. The opposite (and advantageous) situation never seemed to be true, that is WebCompare would score highly a page with just a few highly unexpected terms, with the less unexpected terms not contributing much to the ranking. This condition is much more likely to reveal content pages to the user. This situation could be changed by the simple inclusion of a scaling factor on the unexpectedness score for each term. Using a scale such as squaring the term unexpectedness score would lead to the most unexpected terms being given disproportionately high scores compared with the other terms and it is thought that documents with few highly rated unexpected terms would be favoured. It is thought these are likely to be more interesting to the user. Thus it is recommended that further studies of the WebCompare system may consider incorporating different scaling strategies.

6.3.5 User Test Observations

During testing, the users were encouraged to give their opinions on both the process of specifying their knowledge (i.e. identifying pages) and the quality and characteristics of the results. In the case of specifying knowledge this section is concerned with AISIID's system of specifying pages. In the case of the output, the opinions here are all regarding the pages delivered by AISIID unless otherwise stated.

With regard to the former, users frequently commented that it was hard for them to specify pages that will correctly summarise their knowledge. Generally a page will tend to contain some pieces of knowledge the user does not know. Some users expressed frustration that they could not find high quality content pages on their chosen subject. Related to this, some users became frustrated that they could not submit an entire site for inclusion in their summarised prior knowledge.

In the case of a number of users, the front pages to websites were specified instead of content pages. It is possible the user was searching for general information about that subject, but these front pages contain a great deal of noise and a number of topics. These conspire to produce a final result that is not as good as it could be as some of the retrieved pages reflected this noise or confusion over a concept. It is possible that

AISIID did do a better job at ignoring this noise compared to WebCompare, leading to the scores shown in the previous section.

User11 was especially pleased with her results, stating that the pages returned by AISIID had revealed her to pages she never knew existed. This surprised her as she has done a great deal of research on that subject. She expressed that while most pages retrieved were of low importance to her, the minority that were important were very interesting indeed. The same was true of user13, generally the results were said to be relevant while a small number were very interesting indeed.

In contrast, a small number of tests were found to result in very small absolute scores but this leads to a useful observation regarding a possible shortcoming of WordNet. Most notably, the tests for user3 (areas of shapes), user4 (mathematical graph drawing) and user6 (Markov chains) produced unexpectedly bad results. These three have one thing in common; they are all about mathematical concepts. It is hard to ignore this consistency, and it is thought that two issues conspire to produce bad results. Firstly, the topics submitted, especially in the case of Markov chains were very uncommon; indeed the user admitted himself that even finding good pages through Google was a challenge. Thus the pages that could be found on the topic were exhausted very quickly.

Secondly the language of mathematics is not the sort of language that can be easily transformed by WordNet. Mathematical language tends to describe a concept quite precisely and in addition many of the more technical terms used in those web pages are likely to be too abstract or rarely used to be present in WordNet in the first place. So even if the terms were present in WordNet, due to the nature of mathematical language it would be hard to generate variations on this word. Thus both the aspects of interestingness are impeded, the pages on the topic are sparse throughout the search space resulting in problems finding relevant pages and WordNet has difficulties transforming the mathematical terms which has implications for discovery of interestingness using WordNet in isolation.

6.3.6 Interrogating the Results

This section is concerned with the output of AISIID only. A subset of users received a summary of the interesting words used during the search. It is believed that output such as this is vital for the user for reasons stated in Chapter 2. That is, a good data mining algorithm will allow the user to interrogate its results in order to support the decision making process so that the user can learn more about the patterns in the data under

scrutiny (Fayyad et al., 1996; Freitas, 2002a). In addition to this, interrogating the words produced by WordNet will allow a certain amount of validation that WordNet is producing results broadly as expected, while the procedure of showing interesting words to users is also followed by the authors of WebCompare when presenting the results to the users. In this case the top 15 keywords and concepts for each page are shown to the user after the ranking had completed. No user opinions were recorded in that paper regarding the quality of the list of words shown to the user, but the two lists published appear to be reasonable in the context of the published searches.

In the experiments reported in this thesis, the most interesting words were determined by the following short process. For each user, the URLs shown to that user previously in his or her user test are used. From each URL the cell that had the highest interestingness score with that URL is identified. The words from that cell's interesting word vector (IWV) are then added to a list. Thus 20 sets of interesting words, each relating to a single URL shown to the user are added to the list. The frequency with which each word occurs in the list is computed and these words are ranked in descending order of frequency, and thus assumed importance. As the IWV of each cell is variable in length (the number of words generated by WordNet during each transformation is variable), the length of this list of words shown to the user was also variable in length. The users were asked to use their current knowledge to select the most important words, and then use these in a search engine to discover more interesting pages.

User 9 found words such as *history*, *composer*, *serious_music*, *creative_person*, *author* and *bibliography* particularly interesting. Referring to classical music as "serious music" was a surprise to the user, and referring to the composer as a "creative person" was an unexpected, but reportedly positive description. Using these terms, especially "bibliography" in a standard search was said to be extremely useful by the user. A number of words or phrases were seen to be related to the primary search such as *symphony_orchestra* were included in the interesting words. These were thought to be neither novel nor surprising but were still of use to the user when using the words in a standard search engine.

User10, looking for pages about antigravity, attached significance to words such as *general relativity*, *force, motion*, *effort*, and *locomote*. They were said to reveal that (anti)gravity is a force, and looking for antigravity in the context of a force (rather than as an entity in itself) would not have been thought of.

User12 found fewer words of use compared with the previous two users. It is expected that WordNet had difficulty transforming words in the language of computer gaming. It is also possible that because the user was interested in a particular computer game, described by a noun that of course was not present in WordNet, the concepts that noun is related to are changed rather then the name of the computer game itself. Thus "World of Warcraft" cannot be mutated into the description of another computer game of the same genre, much like the language of mathematics was found to pose problems for WordNet in the previous section. While the proper noun describing the game could not be mutated, the words describing that genre were found in the interesting words. Therefore words such as *attack*, *fight*, *theft*, *invade* and *legion* were all said to be interesting by this user but of little use in searches.

Some general observations were made. It was noticed that, while all interesting words were available to the users, they only tended to browse the first few pages of words. While many of the words were not of use, users tended to agree that those that were of interest were very useful. The most noteworthy aspect of allowing the user to use these interesting words was the manner in which they deployed them when using a search engine. It was noticed that, almost without exception, a user would enter one of the primary search keywords, for example "Prokofiev", but then augment this with one or more interesting keywords – "Prokofiev, symphony orchestra, history". It was fascinating to observe that this mirrored the affinity function of AISIID in which both relevance and interestingness is taken into account. In this case the primary keyword used in the search guarantees the relevance of the returned result while the interesting keywords provide the interestingness. In general it was thought that listing these interesting words was illuminating in the extreme and is therefore thought to be an essential output for such a web mining system.

As noted before, some of the words included in the list shown to the user were not useful. It is thought that many of these are caused by WordNet using the wrong part of speech when generating transformations. When WordNet is used to generate variations on words, all legal parts of speech (noun, verb, adjective, etc.) in which that word exists are used to perform the transformations. Given a single word can often exist in numerous parts of speech this can often result in the set of "interesting" words becoming many times larger than necessary. By tagging every word on a page with its part of speech using a suitable algorithm such as (Brill, 1992) or (Tufis & Mason, 1998), fewer words will be irrelevant and the accuracy of AISIID has the potential to be increased.

6.3.7 Summary

The discovery of interesting information on the web poses some unique challenges. This chapter has evaluated an artificial immune system called AISIID that was specifically developed to meet these challenges. Some characteristics of the AISIID algorithm were explored in an objective manner. These included the population control mechanisms, the population ordering and the distance from the start pages interesting pages were found. It was found that ordering the population by stimulation was advantageous to both the mean affinity of the population and to the depth of search.

During the subjective user tests it was found that AISIID returned pages which were rated significantly more interesting than WebCompare. This is seen as a very positive result as AISIID can, in conclusion, be seen to work better at the task of identifying interesting web pages than the comparison system. In itself, the test was enlightening in terms of testing WebCompare as it had only been tested by 3 users previously, whereas in this investigation 15 users were asked to judge the output. It was noted that WebCompare is susceptible to noise on a page. Some thought was given as to a remedy for this and it was suspected that a non-linear scaled unexpectedness score may positively influence performance. It is thought that this chapter therefore makes a contribution to the literature regarding WebCompare as such a study had not been attempted previously and the study allowed suggestions to be made regarding improvements for this comparison system.

The 15 users were asked to look over the interesting words generated by WordNet, all those users asked found this process useful and helped them understand the data. Limitations were found in the ability for WordNet to transform words; proper nouns and the language of mathematics were found as two examples of this. The ability for users to interrogate the result is one great advantage that AISIID has over Google since, from the user's point of view, Google is a black box, i.e. it returns a set of interesting pages but not a list of interesting or useful words.

It was seen that, on average, Google returned pages that were more interesting than either AISIID or WebCompare. This result is thought to be highly significant in terms of contribution as Google only seeks to retrieve pages that are highly relevant, not interesting. The question must be asked, if AISIID finds pages that are, statistically, no better than Google, then is this a negative result? It is not believed negative. Google has the advantage that retrieving results is almost instantaneous while AISIID takes a very long time. On the other hand AISIID can give the users feedback on why certain pages were thought to be interesting which the users can use in further searches, while Google

is a "black box" to the user. However, it would not be *known* that Google would produce results of comparable interestingness without having undertaken this study, and thus a contribution has been made. Further thoughts on the performance of Google are reserved for the following chapter.

It has been acknowledged that the objective tests used one set of starting pages and this should be taken into account when considering to what extent the results may generalise, however the subjective tests used 15 different users with 15 different sets of start pages and as such it is believed that the results from this section will generalise well.

In its current incarnation it is acknowledged that AISIID's continuous learning characteristics are not as prominent as they would ideally be. Currently each search is run in a batch fashion but it is thought that the characteristics of a continual learning system may be exploited and AISIID could be taken to another level. The ultimate goal of a system like AISIID would be one that would run continuously on a user's computer, silently keeping track of what a user sees on the web every day. As interestingness is subjective, what a user sees at some point in time will impact on the interestingness of information encounters in the future. For example, does this new information contradict the original information (making it surprising), or is this new information a repeat of the original information, thus rendering it less interesting? Thus when a query is submitted to AISIID it already has the knowledge it needs to decide if something is interesting for a user without that user having to supply any additional information. This knowledge is gained from the user's actions over time with the AIS changing and adapting as appropriate. Such a system would be vastly more complex than AISIID and it is believed that AISIID is one step towards this grand goal, such a system would be of great use and would really exploit the lifelong learning characteristics (Hart & Timmis, 2005) that it is believed AIS might possess.

Chapter 7 Further Work and Conclusion

In the previous three chapters, two novel AIS systems for web mining, AISEC and AISIID, have been described and their performance evaluated on real-world data. This final chapter summarises this work, draws some conclusions about the performance of the two proposed systems and makes some recommendations regarding how each may be improved. This chapter ends with some concluding remarks regarding the contributions of the thesis and some thoughts about the future.

7.1 Assessment of AISEC

In Chapter 4, AISEC was presented. AISEC is a novel, immune inspired system built using AIS principles and components with the goal of identifying interesting e-mail in a user's e-mail box. Part of AISEC's novelty is that it works in a dynamic domain where only initialisation was required and the unseen data could change over time whilst still maintaining classification accuracy. AISEC uses an un-intrusive user signalling mechanism to allow the AIS to keep track of changing user preferences and changing topics of the uninteresting e-mails (concept drift). The dynamic nature of the algorithm is assisted, in part, by the use of two separate populations of cells – naïve B-cells and mature memory B-cells. The naïve B-cells are short lived and of unproven use whilst the memory cells are long lived and have been promoted to memory cells because they had proved to provide a correct classification in the past. The internal dynamics which allows AISEC to track concept drift and concept shift is clonal selection. As all new cells were mutated derivatives of known positive (uninteresting) class examples, this leads to an efficient mechanism to promote diversity in the population.

The naïve Bayesian algorithm was identified as suitable comparison algorithm for AISEC and this was implemented and modified for the continuous learning scenario such that it would track user preference in the same way as AISEC. When both were run on the same test set it was shown that AISEC can yield a classification accuracy comparable to a naive Bayesian approach. As the naïve Bayesian algorithm is currently

regarded as one of the most robust method with which to classify text (Mitchell, 1997), and certainly one of the most popular (Chapter 2), this is seen as a positive result. In addition to this, AISEC appeared to react much faster than the naïve Bayesian solution to changes in the data presented to it.

It was acknowledged that AISEC has a fair number of parameters, each of which can be set by the user, and so a parameter sensitivity analysis was undertaken. It was found that just 2 parameters have a great effect on the classification accuracy (`Kc` and `Ka`), another 2 have a moderate effect on the accuracy (`Kt` and `Kts`) while the remaining 4 have little effect. The change in `Kc`, the threshold at which an item was classified as positive or negative, resulted in a variation in accuracy of around 30%. `Ka`, the affinity threshold, impacted the accuracy by a similar margin. Of particular note from the remaining variables, it was seen that that the number of initialisation examples greatly impacted on the variability of the result, with too many training examples resulting in overtraining. Whilst small values of `Kts` gave consistent results, for larger values the results were very inconsistent, with predictive accuracies on some runs of less than 20% (although the mean accuracy did stay above 60%). This analysis was used to allow a more informed choice of values for each parameter which had the effect of improving the classification accuracy (with respect to the default parameter values) with statistical significance.

In the past few years AISEC has been validated by a number of other investigations. (Jang, 2004) conducted a large investigation into the behaviour of AISEC. AISEC was also turned into a functioning user program that can be interfaced with standard e-mail software (Kilgour, 2004). This validated the design characteristics, such as non-intrusive signalling, that allowed AISEC to function correctly in a user environment. Finally, Ayara used the basic structure of the AISEC algorithm to design an algorithm to detect errors in automated teller machines (Ayara et al., 2005; Ayara, 2006). This once again validated the basic AISEC principles and processes in a user scenario, and further validated the ability of AISEC to adapt to changing data as the automatic teller machine was able to learn what specific internal states are likely to lead to future failure.

7.2 Future Work: Improving AISEC

The results presented both in Chapter 4 are encouraging but there are still a number of options available to optimize a system such as AISEC. One obvious area for improvement would be by an increase in the data stored in the B-cell's feature vector, such as a measure of the relative importance of words (using a term frequency/inverse

document frequency approach), coupled with the necessary change in affinity function. It was reasonable to make the simplifying assumption that only information from the header be used for the purposes of this investigation, but an improvement in accuracy might also be made by the use of body text from the e-mail.

From the parameter analysis it was seen that the overly simplistic initialisation stage may be having a detrimental effect on the algorithm's performance. This is especially prominent when there are a relatively large number of initialisation e-mails compared with running e-mails. In light of this, an improved initialisation stage should be investigated. It was shown that parameters such as `Kc` and `Ka` have a great influence on the quality of the result. The creation of a self-optimising mechanism which would continually adapt the algorithm's parameters as the system runs (or using initialisation data before it begins) could be advantageous. A simple solution could tune only the `Kc` value as this has proven to be the most important parameter, lowering it if the user feedback indicates too many false positive classifications and raising it if the opposite is true.

Related to this would be the implementation of a dynamic classification threshold for each immune cell, based on the probability distribution of the data or that cell's performance. In the original system the classification threshold is applied to all immune cells uniformly. Dynamic application of a different classification threshold for each immune cell may increase the algorithm's precision by tailoring each immune cell's antigen recognition level to a level of confidence with which that cell is likely to make a correct judgement.

In addition, the adaptation system the algorithm currently employs works only after positive classification. However, allowing for user feedback in negative classification in addition, and implementing proper reactions to the user feedback, could be one way of making AISEC more adaptive but burdensome on the user.

During the investigations required for this thesis, some conceptual research was undertaken into the use of Danger Theory (Matzinger, 1994; Fuchs & Matzinger, 1996; Matzinger, 1998; Anderson & Matzinger, 2000; Gallucci & Matzinger, 2001; Matzinger, 2002a; Matzinger, 2002b). The details of the application of Danger Theory to AIS are not the subject of this thesis, but e-mail classification is one domain to which it is applicable. Details of the potential application of Danger Theory to AIS in the context of e-mail classification, and therefore of potential advantage to AISEC, can be found in the paper (Secker et al., 2003b; Secker et al., 2005), while the more general

application of Danger theory to AIS is examined in (Aickelin & Cayzer, 2002; Bentley et al., 2005; Twycross & Aickelin, 2005).

7.3 Assessment of AISIID

The immune inspired algorithm AISIID (Artificial Immune System for Interesting Information Discovery) was described in some detail and evaluated. This evaluation was in two parts, the characteristics of the algorithm were investigated in an objective manner, while user tests were used to evaluate the output of the algorithm in a subjective manner.

The domains in which AISIID operates are challenging indeed. The web is enormous in size and full of noise, while the concept of "interestingness" with regard to web documents is still in its infancy. Current search techniques do not allow a user to seek unexpected or surprising documents from a web search as they are, by definition, unexpected and AISIID attempts to provide a solution to this. A number of novel solutions to the challenging nature of searching for interesting documents on the web were incorporated in AISIID. AISIID uses clonal selection principles to maintain a dynamic set of cells, but has an uncommon view of that cell population. Unlike many AIS systems, AISIID is not a passive system where the affinity of an element of data is presented to and assessed by every cell in the AIS. Rather it is a cell's task to actively seek for data. It does this by following links from one page to another, assessing its affinity with each page as it goes. Each cell had a vector of relevant words, which was generated using Google as a tool to infer an approximate TFIDF score for each word on a webpage as no other suitable mechanism was available.

AISIID used an automatic feedback mechanism to control the stimulation of a cell the inspiration for which came from the work on AISEC. Another new strategy used by AISIID was the use of WordNet to semantically transform words in four preset manners in order to produce "interesting" words related to the user's query. These WordNet transformations were selected as they were thought to be the most likely candidates for producing interesting words related to a user's query. To the best of the author's knowledge, this approach to using WordNet is novel, not only in the area of AIS but also in the broader area of web mining.

The characteristics of AISIID were tested and the quality of results was investigated, the latter being performed as set of subjective tests with real users. The population control mechanism was tested and it was found that a high suppression rate of the population would result in a small population but this had the negative effect of

sometimes resulting in the system not terminating. The decision to order the population by stimulation rather than age, age priority being closer to the traditional strategy in AIS systems, was validated by the findings that the stimulation ordering resulted in a higher mean affinity of the population compared to age priority ordering. This age priority ordering was also observed to impact the depth of search with cells spreading further from the initial start pages when using this ordering scheme.

Observations were made of the resultant population when the system terminated. This showed the distribution of cells around the search space was broadly as expected, some being over 30 links away from the original pages. It was surprising to see that cells a fair distance from the starting pages were still finding interesting web pages. Indeed it was observed that the interestingness of the pages (separated from relevance and affinity) grew slightly as the cells reached a distance of around 20 links away.

Studies using real users were performed to gauge the quality of the output in a subjective manner. Compared with WebCompare, the only other system known to perform a similar task to AISIID and using Google as a baseline, it was found that AISIID would on average return pages that were rated by the user as more interesting than those returned by WebCompare. This result was found to be statistically significant. Some explanation was attempted as to why WebCompare did not appear to work well. A contribution was made to the literature regarding WebCompare because a) 15 users were used to give opinion on WebCompare's output, a significant increase on the three used in the original paper, b) this was delivered as a numerical score rather than simply a user's opinion allowing more meaningful interpretation, and c) this investigation highlighted issues with the rankings generated by WebCompare and thoughts regarding a possible remedy were given in the form of a slight change to WebCompare's word unexpectedness measure.

Users were shown the words that resulted in pages being ranked highly by the AISIID. The user could then interrogate these words and a) gain an insight into why pages were ranked highly, and b) use selected words to further their search manually. This was regarded as a very useful exercise by the users. It was interesting to note that when using these words to perform a manual search, users tended to submit one or more relevant words combined with one or more interesting words picked from the list shown to them. This mirrors the affinity function where both relevance and interestingness are taken in into account. As users found elements of the set of words returned to them interesting, credence to the decisions to use WordNet to transform relevant words to create interesting words. However, limitations were found in the ability for WordNet to

transform words; proper nouns, in this case the computer game "World of Warcraft" and the language of mathematics were found as two examples of this. User tests revealed that Google scored higher than both AISIID and WebCompare in terms of subjective interestingness to the user, although the point was made that Google is still a black box, not providing any extra information the user can use to interrogate the decision making process. Comment on the discrepancy between the scores awarded to AISIID pages and Google pages is reserved for section 7.6.

7.4 Future Work: Improving AISIID

During implementation and testing a number of technical improvements that have the potential to increase the quality of AISIID's output were identified. In the following two short sections improvements to the algorithm and to the pre-processing are briefly described acting as motivations for further study and improvement of AISIID.

7.4.1 General Improvements

To begin, further inspiration may be taken from the biology to improve the search. Recall that pages retrieved are thought to be interesting to a user if they are relevant to the user's search but significantly different from the original pages and therefore contain unknown information. This situation is conceptually close to one found in the natural immune system. That is, for T-cells to mature they must recognise the self-protein, MHC, but not bind to any other self-protein (causing an autoimmune reaction). The biological process that performs this task is called negative selection (Sompayrac, 1999). In the case of AISIID, negative selection could be used to remove mutated cells that contain a large number of interesting words that are also found on the original pages. It is acknowledged that negative selection based AIS has been criticised in the literature, especially regarding its application to the task of classification (Freitas & Timmis, 2003), for being a random search method when used in isolation without being integrated with other immune processes. Negative selection in this context would not be used in isolation rather, would be used as an enhancement to clonal selection, an enhancement that can ensure retrieved pages do contain information that is novel to the user.

The ability of AISIID to retrieve information only from hypertext (HTML) documents is an acknowledged limitation. The ability of AISIID to read the content of files such as Adobe's Portable Document Format (PDF) or Postscript would be a great improvement as it would allow users to find more results.

Attribute weighting is a problem encountered throughout the data mining literature, especially when considering instance based learning (Aha et al., 1991; Ling & Wang, 1997). Two weighting strategies could be advantageous to the running of AISIID:

1. Weighting words in mini-documents based on the proximity of the word to the hyperlink it describes. It could be hypothesised that words closer to a hyperlink are more likely to describe that hyperlink.
2. Weight interesting words generated by WordNet based on the transformation used to generate them and/or their location in the hypernym/hyponym hierarchy with respect to the base word. In language, some WordNet transformations may produce words that are fundamentally likely to be more interesting than others. Secondly, as a hierarchy is traversed it is likely that words will become less interesting.

The equal weighting between interestingness and relevance was used as a default position in the absence of any guidance in the matter. Such an investigation to determine the relative importance of these two attributes would be a significant study, but one with the potential to greatly improve the quality of the result. However, studies concerning the perception of interestingness of words generated by different semantic transformations would be required, and this is outside the scope of this thesis.

Another scheme of weighting could make use of the extra information contained in HTML. Compared with text, HTML is rich in metadata and certain parts of webpages can be inferred to be more important or authoritative than others based on their type. For example, a page's title is likely to be of supreme importance followed by 1st and then 2nd level headings. The text identified as having an elevated level of importance could be treated differently.

Furthermore, it may be an advantage for AISIID to view a cell's previous page as a potential move when choosing a link to move to. In the current AISIID algorithm, unless all links on a given page are bad in some way (missing target, pointing at a non text document, etc) then a cell can never return to a previous page. In a situation where a cell is on a page and all links on that page are scored lower than the previous page, it may be advantageous for the cell to back up onto the previous page and reselect the next page to move to.

In the paper (Liu et al., 2001) the authors mine interesting concepts from web pages. The paper describes a concept as a combination of words and states them to be "very informative". Indeed the users interviewed for that paper expressed concepts were of

great help finding interesting pages. To some extent, AISIID does use concepts as WordNet can produce phrases or concepts when transforming a word. However, the situations where these are used are confined only to matching these interesting words with a web page. Applications may include the ability to populate the relevant word vector (RWV) with these concepts to allow an enhanced matching between RWV and pages containing the subject under study.

Finally, as noted in Chapter 3, AIS algorithms show a predisposition toward distribution and AISIID is no exception. The ability to parallelise AISIID will create the potential for the system to retrieve more pages within a set timeframe and therefore potentially improving the quality of the result returned to the user. As implemented in this investigation, AISIID is not distributed as this is a significant challenge in itself, however the following attributes of AISIID allow for distribution at a later date:

1. Cells do not communicate with each other
2. Cells carry a complete copy of all data required for their existence
3. The algorithm is mostly asynchronous. There is only one step in the algorithm where all cells must be in a known state. Any number of cells may be in an undefined state between these times.

7.4.2 Improving Pre-Processing

Whilst AISIID has been designed to be robust against noise by using multiple start pages, it could be further improved if it were to disambiguate noise from content on a web page. Noise is created by advertisements, banners, navigation panels and suchlike and is a distraction from the real content on the page. There already exists one such system to do this, named InfoDiscover (Lin & Ho, 2002). And so the application of this to web page pre-processing should be investigated.

Furthermore, it should be noted that when viewing a webpage, two content sections rendered closely on the screen may not appear in close proximity to each other in the raw HTML. Thus the mini-document generation which relies on the proximity of content to hyperlinks may become confused. However, associating a hyperlink with text as it appears on the screen, rather than as it appears in the raw HTML of the page, would be a significant research topic in itself.

7.5 Reflections on AIS for Web Mining

Previously some reflections on the qualities and characteristics of AISIID and AISEC have been made but it is worth reflecting more generally on the use of AIS for web mining and use in a continuous scenario.

Where implementation is concerned it is thought that there were a number of choices made that really did add positively to the implemented AIS. Firstly, the use of gene libraries as implemented in AISEC was thought to indispensable. It is unlikely that AISEC would have worked at all if random mutations were applied to the feature vectors. This was not contradicted by AISIID as WordNet was used as a type of gene library and as such the words generated were always linked to a set of original and relevant words in some way. I.e. the mutations were not entirely random.

Secondly, it is thought that in a continuous learning scenario, the use of some sort of co-stimulation framework was of great use. This is needed to keep the learning system "on the right track"; punishing it when incorrect decisions are made and rewarding it when correct decisions are made. It was seen that even a very simple implementation of this, i.e. that implemented in AISEC, was enough to produce good results. Without such a mechanism any continuous learning system will not be able to track concept drift and shift and it is believed that this characteristic is a necessary attribute of any continuous learning system.

This work is intended to join the growing ranks of research regarding the use of AIS as a continuous learning technique. When compared to a Bayesian technique, AISEC was shown to be more reactive to changes in data and therefore a more useful technique when dealing with continuously changing data. As has been noted before, there exist some previous works on using AIS for continuous learning. For example, there exist works on dynamic clustering (Hart & Ross, 2002; Neal, 2002; Hart & Ross, 2003) and robotics (Ishiguro et al., 1997a; Ishiguro et al., 1997b; Wantanabe et al., 1999; Hart et al., 2003; Luh & Liu, 2004), each of these, along with this thesis, offers empirical evidence that AIS is useful for continuous learning but to the author's knowledge there exists no convincing proof that this is the case.

In the future it is believed that one niche that AIS will carve for itself is in the realm of continuous data analysis and this thesis goes towards providing evidence for this. This is a thought echoed by the authors of (Hart & Timmis, 2005) in which the ability for an AIS to demonstrate "lifelong learning" is advocated. However, it is recommended that researchers in AIS may wish to perform more experiments with AIS to show that they really are predisposed to continuous learning, taking this characteristic from a "gut

feeling" of the AIS community to an evidence-based observation. Specially constructing datasets with a provable drift and testing these would be one initial step, although these will probably be restricted to numerical datasets compared with text based data as the latter could be problematic to generate. This evidence would be very helpful to determine the performance of an AIS in a continuous scenario, although a mathematical proof that AIS in general are better than other algorithms in this scenario is probably intractable.

Importantly, it is acknowledged that homeostasis is required to support this lifelong learning. So it is believed that in the future, research on using AIS in a continuous scenario with internal dynamics mediated with homeostasis (thus removing as many arbitrary parameters as possible) will be the key to moving AIS forward in the realm of biologically inspired algorithms.

7.6 Interestingness Revisited

At this point it is necessary to revisit the results of the user tests as reported in the previous chapter. Recall, it was found that the search engine, Google, provided a set of pages that were graded by users with a higher score than AISIID. Although this was not seen to be statistically significant, it was an unexpected result as Google was only included in the investigation as a baseline. Why would Google, which ranks pages only in terms of relevance, score so highly compared to a system that has been designed to rank pages in terms of interestingness?

Two explanations are probable. Firstly, the size of Google's database and the quality of its ranking mechanism allow it to discover more interesting pages by default. Google has been retrieving pages from the web to populate its database since the mid 1990s; it also has a huge amount of storage and efficient database search algorithm. In all, over 8,000,000,000 pages are available in the database for ranking. This is combined with development of the page rank algorithm that began in 1995 and has involved many experts all over the world, make it an enormously well developed system. Thus in terms of relevance, the quality of the results obtained by a Google search is extremely high. As has been stated many times, it is believed that two aspects combine to determine the interestingness of a page; namely its relevance combined with its novelty, surprisingness or unexpectedness. Thus if the relevance aspect of a page is exceptionally high, the overall interestingness of the page may be accordingly high even if the novelty is moderate. This suggests that in the future a simpler measure of finding pages may be preferential if it allows for the retrieval of significantly more pages.

The second explanation is that the current understanding of interestingness in text mining and possibly data mining may be inaccurate or misunderstood. It would seem that users, when not told what interestingness actually is have decided that relevant pages are interesting. This does indeed appear to fit the observations from the previous chapter. The lowest ranked system was WebCompare. This ranks pages only in terms of their interestingness. At the other end of the scale, Google received by far the highest "interestingness" score for the users, yet Google only seeks to deliver pages that are relevant. In the middle of these was AISIID, a system that attempts to fuse the two elements of interestingness and relevance to provide a ranking.

In this case, in the realm of text or web mining it appears that users regard relevance as a priority. As the attributes of interestingness in the web mining domain were originally inspired by some of the attributes of interestingness in the domain of classification and association rules, this poses the further question, is it possible that these attributes simply do not transfer well from one aspect of data mining to the other. Without further research in this area it is impossible to tell if this is the case or not it is possible that the fusion of relevance and interestingness as demonstrated previously in this thesis is one step towards a better performing metric as the evidence suggested users prized this approach over surprisingness alone. As such, this suggestion is thought to be of interest regarding this thesis' contribution.

With regard to the rules of interestingness transferring poorly from the evaluation of classification or association rules, there is one further thought. That is, are the attributes of interestingness regarding classification rules incorrect in some way themselves, and this is having a knock-on effect when discovering interesting text. This suggestion is a bold one, and without any further evidence is one which would be unwise to make. As metrics involving interestingness of rules have been developing over many years, to suggest they may be lacking based on the investigations in this chapter could be regarded, at first glance as too strong a criticism for which there is little evidence. However, just in the last year evidence has come to light that this may indeed be the case. Continuing the work in (Carvalho et al., 2003), in (Carvalho et al., 2005) the authors evaluated 11 objective rule interestingness measures applied to 8 different datasets. Users were used to assess the interestingness of the resulting rules. Only 35% of the tests were found to produce rules that were interesting to a user and no relationship between a rule interestingness measure and a dataset that consistency produced good results was found.

Thus, in isolation, any explanations as to why Google may have resulted in a higher interestingness would only be conjecture, yet in this context it may add to a small number of investigations questioning the accepted metrics of interestingness in data mining. The subjective testing of AISIID has forced questions to be asked regarding the meaning of interestingness in terms of text mining and a further questioning of the true meaningfulness of the objective measures of rule based interestingness from which these ideas of text based interestingness stem.

7.7 Summary

This thesis was introduced with the goal of contributing to the two areas of AIS and web content mining by investigating the discovery of interesting information using an artificial immune system. More precisely, two AIS for web mining were proposed in this thesis. The first system, AISEC, classified e-mails as either interesting or uninteresting. AISEC made a contribution to the field by investigating the use of an AIS for continuously classifying data and as such it might be one step towards a system for "life-long learning" as identified as a future direction for AIS in (Hart & Timmis, 2005). It is satisfying to see that this algorithm has already made a significant impact (Jang, 2004; Kilgour, 2004; Ayara et al., 2005; Ayara, 2006).

The second system, AISIID, is expected to make a contribution both to AIS and data mining. Like AISEC, AISIID also is very much a problem driven system, an approach advocated in (Freitas & Timmis, 2003). AISIID worked in a novel way, by actively searching for information and included a number of other aspects such as an automated feedback mechanism that may be of interest to the AIS community. In addition, contribution has been made to the data mining community as, for example, WordNet had never been used before for the generation of interesting words. However it is the results of the subjective evaluations of the users about the interestingness of the discovered web pages that may be of greatest interest to data mining. The results from these user studies allowed the suggestion to be made that the rules of determining interestingness in the context of classification and association rules may not transfer well into the domain of text mining. In addition to this, the work in (Carvalho et al., 2005) provides evidence that the process of discovering interesting rules may need revaluating and if this is the case, it is likely that interestingness regarding text will also need further study.

In the future it is expected that users will demand more from their information retrieval or classification tools. Whether that would be keeping track of the messages

they receive in their inbox in real time and adjusting the order in which they are presented, or whether it be performing more intelligent searches on the web using enhanced searches, the results of which have been ranked using linguistics and some notion of interestingness, it is expected that this future will lie in intelligent systems. While the conclusions reached regarding the identification of interestingness were unexpected, they should not be overlooked, especially in the wake of the other recent literature. With time spent now on an effort to fully understand what users actually find interesting in documents, the area can advance in the future towards better methods of performing web mining. This, supported by intelligent systems of which artificial immune systems is one example, can only be a good thing for an end user, allowing for a reduction in effort spent searching. It is believed that this thesis is one small, but important step towards meeting this important challenge.

References

Aggarwal, C. C. (2004). On Leveraging User Access Patterns for Topic Specific Crawling. *Data Mining and Knowledge Discovery, 9*(2), 123-145.

Aggarwal, C. C., & Yu, P. S. (2002). *An Automated System for Web Portal Personalization.* Very Large Databases (VLDB) 2002, Hong Kong, China. Morgan Kaufmann, pp. 1031-1040

Aha, D. W. (1992). Tolerating Noisy, Irrelevant and Novel Attributes in Instance-Based Learning Algorithms. *International Journal of Man-Machine Studies, 6*(1), 267-287.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-Based Learning Algorithms. *Machine Learning, 6*(1), 37-66.

Aickelin, U., Bentley, P., Cayzer, S., Kim, J., & McLeod, J. (2003). *Danger Theory: The Link between AIS and IDS.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 147-155

Aickelin, U., & Cayzer, S. (2002). *The Danger Theory and Its Application to Artificial Immune Systems.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 141-148

Aickelin, U., Greensmith, J., & Twycross, J. (2004). *Immune System Approaches to Intrusion Detection.* 3rd International Conference on Artificial Immune Systems (ICARIS 2004), Catania, Sicily. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 316-329

Alder, H. L., & Roessler, E. B. (1968). *Introduction to Probability and Statistics*: W. H. Freeman.

Alves, R. T., Delgado, M. R., Lopes, H. S., & Freitas, A. A. (2004). *An Artificial Immune System for Fuzzy-Rule Induction in Data Mining.* Parallel Problem Solving from Nature (PPSN-2004). Lecture Notes in Computer Science 3242. Springer-Verlag, pp. 1011-1020

Amazon. (2005). *Amazon.com.* Retrieved June 2005 from http://www.amazon.com/

Anderson, C. C., & Matzinger, P. (2000). Danger: The View From the Bottom of the Cliff. *Seminars in Immunology, 12*(3), 231-238.

Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., & Spyropoulos, C. (2000a). *An Evaluation of Naive Bayesian Anti-Spam Filtering.* Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, Barcelona, Spain. pp. 9-17

Androutsopoulos, I., *et al.* (2000b). *Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach.* 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lyon, France. pp. 1-13

Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995). *WebWatcher: A Learning Apprentice for the World Wide Web.* 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, Stanford, USA. pp. 6-12

Ayara, M. (2006). *An Immune Inspired Approach for Adaptable Error Detection in Embedded Systems.* Ph.D. Thesis, University of Kent, Canterbury, England.

Ayara, M., Timmis, J., R., d. L., & Forrest, S. (2005). *Immunising Automated Teller Machines.* ICARIS 2005, Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 404-417

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Harlow: Addison Wesley Longman.

Basu, S., Mooney, R. J., K.V.Pasupuleti, & Ghosh, J. (2001). *Evaluating the Novelty of Text-Mined Rules Using Lexical Knowledge.* 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), San Francisco, California, USA. ACM Press, pp. 233 - 238

Benbrahim, H., & Bramer, M. (2005). *Experiments in Hypertext Categorisation.* First UK Knowledge Discovery in Data Symposium, Liverpool, UK. pp. 42-48

Bentley, P. (2001). *Digital Biology*. London: Headline.

Bentley, P. J., Greensmith, J., & Ujjin, S. (2005). *Two Ways to Grow Artificial Tissue for Artificial Immune Systems.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 139-152

Bersini, H. (2002). *Self-Assertion Versus Self-Recognition: A Tribute to Francisco Varela.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 107-112

Bezerra, G. B., Barra, T. V., de Castro, L. N., & Von Zuben, F. J. (2005). *Adaptive Radius Immune Algorithm for Data Clustering.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 290-303

Blake, C. L., & Merz, C. J. (1998). *UCI Repository of Machine Learning Databases.* Retrieved May 2003 from http://www.ics.uci.edu/~mlearn/MLRepository.html

Brill, E. (1992). *A Simple Rule-Based Part-of-Speech Tagger.* 3rd Conference on Applied Natural Language Processing (ANLP-92), Trento, Italy. ACM Press, pp. 152-155

Brutlag, J. D., & Meek, C. (2000). *Challenges of the Email Domain for Text Classification.* Seventeenth International Conference on Machine Learning (ICML 2000), Stanford, USA. pp. 103-110

Burnett, F. M. (1959). *The Clonal Selection Theory of Acquired Immunity*: Cambridge University Press.

Carvalho, D. R., Freitas, A. A., & Ebecken, N. (2005). *Evaluating the correlation between objective rule interestingness measures and real human interest.* European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2005). Lecture Notes in Artificial Intelligence 3721. Springer, pp. 453-461

Carvalho, D. R., Frietas, A. A., & Ebecken, N. F. F. (2003). *A Critical Review of Rule Surprisingness Measures.* Data Mining IV - International Conference on Data Mining, Rio de Janeiro, Brazil. WIT Press, pp. 545-556

Cayzer, S., & Aickelin, U. (2002a). *On the Effects of Idiotypic Interactions for Recommendation Communities in Artificial Immune Systems.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 154-160

Cayzer, S., & Aickelin, U. (2002b). *A Recommender System Based on the Immune Network.* CEC 2002, Honolulu. pp. 807-813

Cayzer, S., Smith, J., Marshall, J. A. R., & Kovacs, T. (2005). *What Have Gene Libraries Done For AIS?* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 86-99

Chakrabarti, S. (2000). Data Mining for Hypertext: A Tutorial Survey. *SIGKDD Explorations, 1*(2), 1-11.

Chakrabarti, S. (2003). *Mining the web (Discovering Knowledge from Hypertext Data).* San Francisco: Morgan Kaufmann.

Chakrabarti, S., Berg, M. v. d., & Dom, B. (1999). *Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery.* 8th World Wide Web Conference, Toronto, Canada. pp. 545-562

Chan, P. K. (1999). *A Non-Invasive Learning Approach to Building Web User Profiles.* Workshop on Web Usage Analysis and User Profiling, Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, USA. pp. 7-12

Chao, D. L., & Forrest, S. (2002). *Information Immune Systems.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 132-140

Chaves, R. P. (2001). *WordNet and Automated Text Summarization.* Sixth Natural Language Processing Pacific Rim Symposium, Tokyo, Japan. pp. 109-116

Chen, C. C., Chen, M. C., & Sun, Y. (2001). *PVA: A Self-Adaptive Personal View Agent System.* 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA. ACM Press, pp. 257-262

Chen, L., & Sycara, K. (1998). *WebMate: A Personal Agent for Browsing and Searching.* 2nd International Conference on Autonomous Agents (Agents '98), Minneapolis, USA. ACM Press, pp. 132 - 139

Clark, E., Hone, A., & Timmis, J. (2005). *A Markov Chain Model of the B-Cell Algorithm.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 318-330

Crawford, E., Kay, J., & McCreath, E. (2001). *Automatic Induction of Rules for e-mail Classification.* Australian Document Computing Symposium (ADCS 2001), Coffs Harbour, Australia. pp. 13-20

Creecy, R. H., Masand, B. M., Smith, S. J., & Waltz, D. L. (1998). Knowledge Engineering. *Communications of the ACM, 35*(8), 48-63.

Creighton, J. H. (1994). *A First Course in Probability Models and Statistical Inference*: Springer-Verlag.

Cruz-Cortés, N., Trejo-Pérez, D., & Coello Coello, C. A. (2005). *Handling Constraints in Global Optimisation Using an Artificial Immune System.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 234-247

Cunningham, P., Nowlan, N., Delany, S. J., & Haahr, M. (2003). *A Case-Based Approach to Spam Filtering that Can Track Concept Drift* (Technical Report TCD-CS-2003-16): Trinity College, Dublin, Ireland.

Cutello, V., Narzisi, G., Nicosia, G., & Pavone, M. (2005). *Clonal Selection Algorithms: A Comparative Case Study Using Effective Mutation Potentials.* ICARIS 2005, Banff, Canada. Lecture Notes In Computer Science 3627. Springer-Verlag, pp. 13-28

Dasgupta, D. (1999). *Artificial Immune Systems and Their Applications*: Springer-Verlag.

Dasgupta, D., & Gonzalez, F. (2002). An Immunity-Based Technique to Characterize Intrusions in Computer Networks. *IEEE Transactions on Evolutionary Computation, 6*(3), 1081-1088.

Dasgupta, D., Ji, Z., & Gonzalez, F. (2003). *Artificial Immune Systems (AIS) Research in the Last Five Years.* CEC 2003, Canberra, Australia. IEEE, pp. 123-130

Dasgupta, D., KrishnaKumar, K., Wong, D., & M.Berry. (2004). *Negative Selection Algorithm for Aircraft Fault Detection.* 3rd International Conference on Artificial Immune Systems (ICARIS 2004), Catania, Sicily. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 1-13

de Castro, L., & Von Zuben, F. (2001). Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, 6*(3), 239-251.

de Castro, L. N. (2001). *Immune Engineering: Development of Computational Tools Inspired by the Artificial Immune Systems.* Ph.D. Thesis, DCA – FEEC/UNICAMP, Campinas/SP, Brazil.

de Castro, L. N. (2003). *The Immune Response of an Artificial Immune Network (aiNet).* CEC 2003, Canberra, Australia. IEEE, pp. 146-153

de Castro, L. N., & Timmis, J. (2002a). *Artificial Immune Systems: A New Computational Intelligence Approach*: Springer-Verlag.

de Castro, L. N., & Timmis, J. (2002b). *Artificial Immune Systems: A Novel Paradigm to Pattern Recognition.* SOCO-2002, Paisley, UK. pp. 67-84

de Castro, L. N., & Timmis, J. (2003). Artificial Immune Systems as a Novel Soft Computing Paradigm. *Soft Computing, 7*(8), 526-544.

de Castro, L. N., & Von Zuben, F. J. (2000). *The Clonal Selection Algorithm with Engineering Applications.* GECCO 2000, Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA. pp. 36-39

de Castro, L. N., & Von Zuben, F. J. (2002). aiNet: An Artificial Immune Network for Data Analysis. In Abbass, H., Sarker, R. & Newton, C. (Eds.), *Data Mining: A Heuristic Approach* (pp. 231-259): Idea Group.

de Jong, K. (1999). *Genetic Algorithms: A 30 Year Perspective.* Festschrift Conference in Honour of John H. Holland, University of Michigan.

Diao, Y., Lu, H., & Wu, D. (2000). *A Comparative Study of Classification Based Personal E-Mail Filtering.* 4th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2000), Kyoto, Japan. Lecture Notes in Computer Science 1805. Springer-Verlag, pp. 408-419

Didion, J. (2005). *Java WordNet Library Homepage.* Retrieved November 2005 from http://sourceforge.net/projects/jwordnet

Domingos, P. (1998). *Occam's Two Razors: The Sharp and the Blunt.* 4th International Conference on Knowledge Discovery and Data Mining (KDD '98), New York City, USA. AAAI Press, pp. 37-43

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*: MIT Press.

Faloutsos, C., & Oard, D. W. (1995). *A Survey of Information Retrieval and Filtering Methods* (CS-TR-3514): University of Maryland.

Farmer, J. D., Packard, N. H., & A.S.Perelson. (1986). The Immune System, Adaptation and Machine Learning. *Physica, 22*(D), 187-204.

Fawcett, T. (2003). "In vivo" Spam Filtering: A Challenge Problem for KDD. *ACM SIGKDD Explorations, 5*(2), 140-148.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*: MIT Press.

Feldman, R. (2002). Text mining. In Klosgen, W. & Zytkow, J. (Eds.), *Handbook of Data Mining and Knowledge Discovery* (pp. 749 - 757): Oxford University Press.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge: MIT press.

Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994). *Self-Nonself Discrimination in a Computer.* IEEE Symposium on Research in Security and Privacy, Los Alamitos, USA. IEEE Computer Society Press, pp. 202-212

Freitas, A. A. (1998). *On Objective Measures of Rule Surprisingness.* Principles of Data Mining and Knowledge Discovery (Proceedings of 2nd European Symposium PKDD'98), Nantes, France. Lecture Notes in Artificial Intelligence 1510. Springer-Verlag, pp. 1-9

Freitas, A. A. (2000). Understanding the Crucial Differences Between Classification and Discovery of Association Rules - A Position Paper. *ACM SIGKDD Explorations, 2*(1), 65-69.

Freitas, A. A. (2002a). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*: Springer-Verlag.

Freitas, A. A. (2002b). Evolutionary Computation. In Klosgen, W. & Zytkow, J. (Eds.), *Handbook of Data Mining and Knowledge Discovery* (pp. 698-706): Oxford University Press.

Freitas, A. A., & Timmis, J. (2003). *Revisiting the Foundations of Artificial Immune Systems: A problem-orientated Perspective.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 229-241

Friedman, N., & Kohavi, R. (2002). Bayesian Classification. In Klosgen, W. & Zytkow, J. M. (Eds.), *Handbook of Data Mining and Knowledge Discovery* (pp. 282-288): Oxford University Press.

Fuchs, E. J., & Matzinger, P. (1996). Is Cancer Dangerous to the Immune System? *Seminars in Immunology, 8*(5), 271-280.

Gallucci, S., & Matzinger, P. (2001). Danger Signals: SOS to the Immune System. *Current Opinion in Immunology, 13*(1), 114-119.

Gaspar, A., & Hirsbrunner, B. (2002). *From Optimisation to Learning in Changing Environments: The Pittsburgh Immune Classifier System.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 190-199

Gaspar, A., & Hirsbrunner, B. (2004). *PICS: Pittsburgh Immune Classifier System.* AISB 2004 Symposium on the Immune System and Cognition (ImmCog-2004), Leeds, UK.

Germain, R. N. (2004). An Innately Interesting Decade of Research in Immunology. *Nature Medicine, 10*(12), 1307-1320.

Ghani, R., Slattery, S., & Yang, Y. (2001). *Hypertext Categorization Using Hyperlink Patterns and Meta Data.* 18th International Conference on Machine Learning (ICML 2001), Massachusetts, USA. Morgan Kaufmann, pp. 178-185

Google. (2005a). *Google Directory*. Retrieved May 2005 from http://www.google.com/Top/

Google. (2005b). *Google: Google Web APIs (beta)*. Retrieved November 2005 from http://www.google.com/apis/

Google. (2006a). *Google Guide: Similar pages*. Retrieved January 2006 from http://www.googleguide.com/similar_pages.html

Google. (2006b). *Google: Google Search Homepage*. Retrieved January 2006 from http://www.google.com

Graham, P. (2003). *A Plan for Spam*. Retrieved 23 April from http://www.paulgraham.com/spam.html

Greensmith, J. (2003). *New Frontiers for an Artificial Immune System.* Masters Dissertation, University of Leeds, Leeds.

Greensmith, J., Aickelin, U., & Cayzer, S. (2005). *Introducing Dendritic Cells as a novel Immune-Inspired Algorithm for Anomaly Detection.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 153-167

Greensmith, J., & Cayzer, S. (2003). *An Artificial Immune System Approach to Semantic Document Classification.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 136-146

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*: MIT Press.

Hand, D. J., & Zhang, Z. (2005). *Spotting the Difference: Detecting Local Structures in Large Data Sets.* 1st UK Knowledge Discovery in Databases Symposium, Liverpool, England. pp. 13-20

Hang, X., & Dai, H. (2004). *An Immune Network Approach for Web Document Clustering.* 2004 IEEE/WIC/ACM International Conference on Web intelligence, Beijing, China.

Harmer, P. K., & Lamont, G. B. (2000). *An Agent Based Architecture for a Computer Virus Immune System.* GECCO 2000, Workshop on AIS and Their applications, Las Vegas, USA. pp. 45-46

Hart, E. (2005). *Not All Balls Are Round: An Investigation of Alternative Recognition-Region Shapes.* ICARIS 2005, Banff, Canada. Lecture Notes In Computer Science 3627. Springer-Verlag, pp. 29-42

Hart, E., & Ross, P. (2002). *Exploiting the Analogy Between Immunology and Sparse Distributed Memories: a System for Clustering Non-Stationary Data.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 49-58

Hart, E., & Ross, P. (2003). *Improving SOSDM: Inspirations from Danger Theory.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 194-203

Hart, E., Ross, P., Webb, A., & Lawson, A. (2003). *A Role for Immunology in "Next Generation" Robot Controllers.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 46-56

Hart, E., & Timmis, J. (2005). *Application Areas of AIS: The Past, The Present and The Future.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 483-497

Hilderman, R., & Hamilton, H. (2000). *Principles for Mining Summaries Using Objective Measures of Interestingness.* 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '00), Vancouver, Canada. IEEE Press, pp. 72-81

Hilderman, R. J., & Hamilton, H. J. (2001). *Knowledge Discovery and Measures of Interest*: Kluwer Academic Publishers.

Hoffman, G. W. (1986). A Neural Network Model Based on the Analogy with the Immune System. *Journal of Theoretical Biology, 122*, 33-67.

Hofmeyr, S., & Forrest, S. (1999). *Immunity by Design: An Artificial Immune System.* Genetic and Evolutionary Computation Conference (GECCO 1999). Morgan-Kaufmann, pp. 1289-1296

Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an Artificial Immune System. *Evolutionary Computation, 7*(1), 45-68.

Holden, N., & Freitas, A. A. (2004). *Web Page Classification With an Ant Colony Algorithm.* Parallel Problem Solving from Nature (PPSN 2004), Birmingham, UK. Lecture Notes In Computer Science 3242. Springer-Verlag, pp. 1092-1102

Holland, J. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*: MIT Press.

Hotho, A., Staab, S., & Stumme, G. (2003). *WordNet Improves Text Document Clustering.* Semantic Web Workshop at SIGIR-2003, Toronto, Canada. ACM Press,

Hunt, J., Timmis, J., Cooke, D., Neal, M., & King, C. (1998). JISYS: Development of an Artificial Immune System for Real World Applications. In *Artificial Immune Systems and Their Applications* (pp. 157-186): Springer-Verlag.

Hunt, J. E., & Cooke, D. E. (1996). Learning Using an Artificial Immune System. *Journal of Network and Computer Applications, 19*(2), 189-212.

Inoue, A., & Ralescu, A. L. (1999). *E-mail Classification Support Reflecting a User's Perception.* COIL Workshop in User Modelling, Bristol, England.

Ishiguro, A., Kondo, T., & Watanabe, Y. (1997a). *Emergent Construction of Artificial Immune Networks for Autonomous Mobile Robots.* 1997 IEEE International Conference on System Man and Cybernetics, Orlando, USA. IEEE, pp. 1222-1228

Ishiguro, A., Watanabe, Y., Kondo, T., Shirai, Y., & Uchikawa, Y. (1997b). *A Robot With a Decentralized Consensus-Making Mechanism Based on the Immune*

System. 3rd International Symposium on Autonomous Decentralized Systems. IEEE, pp. 231-237

Janeway, C. A. (1994). How the Immune System Recognises Invaders. In *Life, Death and the Immune System* (pp. 28-36): Scientific American.

Jang, J. S. (2004). *An Empirical Investigation into an Artificial Immune System for Email Classification (AISEC).* Masters Dissertation, University of Kent, Canterbury, England.

Jerne, N. K. (1974). Towards a Network Theory of the Immune System. *Annals of Immunology, 125*(C), 373-389.

Joachims, T., Freitag, D., & Mitchell, T. (1996). *WebWatcher: A Tour Guide for the World Wide Web* (Technical Report CMU-CS-96-xxx). Pittsburgh: School of Computer Science, Carnegie Mellon University.

Katirai, H. (1999). *Filtering Junk E-Mail: A Performance Comparison between Genetic Programming and Naive Bayes* (Technical Report): Department of Electrical and Computer Engineering, University of Waterloo, Ontario.

Kelsy, J., Timmis, J., & Hone, A. (2003). *Chasing Chaos.* CEC 2003, Canberra, Australia. IEEE Press, pp. 413-419

Kibler, D., Aha, D. W., & Albert, M. K. (1989). Instance-Based prediction of Real-Valued Attributes. *Computer Intelligence, 5*(2), 51-57.

Kilgour, A. (2004). *Developing a Practical Artificial Immune System for Email Classification.* Masters Dissertation, University of Kent, Canterbury, UK.

Kim, J., & Bentley, P. (1999). *The Human Immune System and Network Intrusion Detection.* 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany.

Kim, J., & Bentley, P. (2001a). *An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection.* Genetic and Evolutionary Computation Conference 2001 (GECCO 2001), San Francisco, USA. pp. 1330 - 1337

Kim, J., & Bentley, P. (2001b). *Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Clonal Selection with a Negative Selection Operator.* Congress on Evolutionary Computation (CEC 2001), Seoul, Korea. pp. 1244-1252

Kim, J., & Bentley, P. (2002a). *Immune Memory in the Dynamic Clonal Selection Algorithm.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 59-67

Kim, J., & Bentley, P. (2002b). *A Model of Gene Library Evolution in the Dynamic Clonal Selection Algorithm.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 182-189

Kim, J., & Bentley, P. (2002c). *Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection.* Congress on Evolutionary Computation (CEC-2002), Honolulu. IEEE, pp. 1015-1020

Kim, J., Ong, A., & Overill, R. E. (2003). *Design of an Artificial Immune System as a Novel Anomaly Detector for Combating Financial Fraud in the Retail Sector.* CEC 2003, Canberra, Australia. IEEE Press, pp. 405-412

Kim, J., Wilson, W. O., Aickelin, U., & McLeod, J. (2005). *Cooperative Automated Worm Response and Detection Immune Algorithm (CARDINAL) Inspired by T-Cell Immunity and Tolerance.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 168-181

Klinkenberg, R. (1999). *Learning Drifting Concepts with Partial User Feedback.* FGML-99, Magdeburg, Germany. pp. 212-225

Klinkenberg, R., & Renz, I. (1998). *Adaptive Information Filtering: Learning in the Presence of Concept Drifts.* ICML/AAAI-98 Workshop - Learning for Text Categorization, California, USA. AAAI Press, pp. 33-40

Knight, T., & Timmis, J. (2001). *AINE: An Immunological Approach to Data Mining.* IEEE International Conference on Data Mining, San Jose, USA. IEEE Press, pp. 297-304

Knight, T., & Timmis, J. (2002). *A Multi-Layered Immune Inspired Approach to Data Mining.* 4th International Conference on Recent Advances in Soft Computing, Nottingham, UK. pp. 266-271

Knight, T., & Timmis, J. (2003). A Multi-layered Immune Inspired Machine Learning Algorithm. In Lotfi, A. & Garibaldi, M. (Eds.), *Applications and Science in Soft Computing* (pp. 195-202): Springer-Verlag.

Kononenko, I. (1991). *Semi-Naive Bayesian Classifier.* European Working Session on Learning: Machine Learning (EWSL-91), Porto, Portugal. Lecture Notes in Artificial Intelligence 482. Springer-Verlag, pp. 206-219

Kosala, R., & Blockeel, H. (2000). Web Mining Research: A Survey. *SIGKDD Explorations, 2*(1), 1-15.

Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. (2002). The Web and Social Networks. *IEEE Computer, 35*(11), 32-36.

Lancaster, F. W. (1976). *Vocabulary Control for Information Retrieval.* Washington D. C: Information Resources Press.

Langman, R. E., & Cohn, M. (1986). The Complete Idiotype Network is an Absurd Immune System. *Immunology Today, 7*(4), 120-121.

Laware, G. W. (2005). Metadata Management: A Requirement for Web Warehousing and Knowledge Management. In A.Scime (Ed.), *Web Mining: Applications and Techniques* (pp. 1-26). London: Idea Group Publishing.

Leacock, C., & Chodorow, M. (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database* (pp. 225-283). Cambridge: MIT Press.

Lee, C. (1994). *An Instance-Based Learning Method for Databases: An Information Theoretic Approach.* 9th European Conference on Machine Learning (ECML 94), Catania, Italy. Lecture Notes In Computer Science 1224. Springer-Verlag, pp. 387-390

Leng, P. (2005). *Computing Association Rules from Incomplete Support Counts.* 1st UK Knowledge Discovery in Databases Symposium, Liverpool, England. pp. 29-34

Liao, T. W., Zhang, Z., & Mount, C. R. (1998). Similarity Measures for retrieval in Case-Based Reasoning Systems. *Applied Artificial Intelligence, 12*(4), 267-288.

Lin, S. H., & Ho, J. M. (2002). *Discovering Informative Content Blocks from Web Documents.* 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada. ACM Press, pp. 588-593

Ling, C. X., & Wang, H. (1997). Computing Optimal Attribute Weight Settings for Nearest Neighbour Algorithms. *Artificial Intelligence Review, 11*(1-5), 255-272.

Linoff, G. S., & Berry, M. J. A. (2001). *Mining the web (Transforming Customer Data into Customer Value)*: Wiley.

Liu, B., & Chang, K. C.-C. (2005). Editorial: Special Issue on Web Content Mining. *SIGKDD Explorations, 6*(2), 1-4.

Liu, B., & Hsu, W. (1996). *Post-Analysis of Learned Rules.* 13th National Conference on Artificial Intelligence (AAAI '96), Menlo Park, USA. AAAI Press, pp. 828-834

Liu, B., Hsu, W., & Chen, S. (1997). *Using General Impressions to Analyze Discovered Classification Rules.* 3rd International Conference on Knowledge Discovery and Data Mining (KDD '97), Newport Beach, USA. pp. 31-36

Liu, B., Hsu, W., Mun, L.-F., & Lee, H.-Y. (1999). Finding Interesting Patterns Using User Expectations. *IEEE Transactions on Knowledge and Data Engineering, 11*(6), 817-832.

Liu, B., Hu, M., & Hsu, W. (2000). *Multi-Level Organisation and Summarisation of the Discovered Rules.* 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000), Boston, USA. ACM Press, pp. 208-217

Liu, B., Ma, Y., & Yu, P. S. (2001). *Discovering Unexpected Information From Your Competitors' Web Sites.* 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001), San Francisco, USA. ACM Press, pp. 144-153

Luh, G.-C., & Liu, W.-W. (2004). *Reactive Immune Network Based Mobile Robot Navigation.* 3rd International Conference on Artificial Immune Systems

(ICARIS 2004), Catania, Sicily. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 119-132

Luke, S. (2000). *The Mersenne Twister in Java*. Retrieved May 2003 from http://www.cs.umd.edu/users/seanl/gp/

Manaco, G., Masciari, E., & Ruffolo, M. (2002). *Towards An Adaptive Mail Classifier.* Italian Association for Artificial Intelligence Workshop (AIIA 2002).

Manning, C. D., & Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press.

Markellou, P., Rigou, M., & Sirmakessis, S. (2005). Mining for Web Personalisation. In A.Scime (Ed.), *Web Mining: Applications and Techniques* (pp. 27-48). London: Idea.

Marrack, P., & Kappler, J. W. (1994). How the Immune System Recognises the Body. In *Life, Death and the Immune System* (pp. 39-49): Scientific American.

Marwah, G., & Watkins, A. (2002). *Artificial Immune Systems for Classification: Some Issues.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 149-153

Matsumoto, M., & Nishimura, T. (1998). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modelling and Computer Simulation, 8*(1), 3-30.

Matzinger, P. (1994). Tolerance, Danger, and the Extended Family. *Annual Review of Immunology, 12*, 991-1045.

Matzinger, P. (1998). An Innate Sense of Danger. *Seminars in Immunology, 10*(5), 399-415.

Matzinger, P. (2002a). The Danger Model: A Renewed Sense of Self. *Science, 296*, 301-305.

Matzinger, P. (2002b). *The Real Function of the Immune System or Tolerance and The Four D's*. Retrieved October 2002 from http://cmmg.biosci.wayne.edu/asg/polly.html

Merkin, B. G. (1996). *Mathematical classification and clustering*: Kluwer Academic Publishers.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*: MIT Press.

Mitchell, T. M. (1997). *Machine Learning*: McGraw-Hill.

Morrison, T., & Aickelin, U. (2002). *An Artificial Immune System as a Recommender for Web Sites.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 161-169

Moukas, A. G. (1996). *Amalthaea: Information Discovery and Filtering Using a Multiagent Evolving Ecosystem.* Conference on Practical Applications of Agents and Multiagent Technology, London, UK.

Moukas, A. G. (1997). *Amalthaea: Information Filtering and Discovery using a Multiagent Evolving System.* Masters Dissertation, M.I.T, USA.

Moukas, A. G., & Maes, P. (1998). Amalthaea: An Evolving Multi-Agent Information Filtering and Discovery System for the WWW. *Autonomous Agents and Multi-Agent Systems, 1*(1), 59-88.

Mutton, P. (2004). *Inferring and Visualizing Social Networks on Internet Relay Chat.* 8th International Conference on Information Visualization (IV04), London, UK.

Nanas, N., Uren, V., & Roeck, A. d. (2004). *Nootropia: A User Profiling Model Based on a Self-Organizing Term Network.* 3rd International Conference on Artificial Immune Systems (ICARIS 2004), Catania, Italy. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 146-160

Nasraoui, O., Dasgupta, D., & Gonzalez, F. (2002). *The Promise and Challenges of Artificial Immune System Based Web Usage Mining: Preliminary Results.* SIAM Workshop on Web Analytics, Arlington, VA. pp. 29-39

Nasraoui, O., Gonzalez, F., Cardona, C., Rojas, C., & Dasgupta, D. (2003). *A Scalable Artificial Immune System Model for Dynamic Unsupervised Learning.* GECCO 2003, Chicago, USA. Lecture Notes in Computer Science 2723. Springer-2723erlag, pp. 219-230

Neal, M. (2002). *An Artificial Immune System for Continuous Analysis of Time-Varying Data.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 76-83

Neal, M., Hunt, J., & Timmis, J. (1998). *Augmenting an Artificial Immune Network.* International Conference on Systems and Man and Cybernetics, San Diego, USA. pp. 3821-3826

Newborough, J., & Stepney, S. (2005). *A Generic Framework for Population-Based Algorithms, Implemented on Multiple FPGAs.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes In Computer Science 3627. Springer-Verlag, pp. 43-55

Nossal, I. J. V. (1994). Life, Death and the Immune System. In *Life, Death and the Immune System* (pp. 1-12): Scientific American.

Oda, T., & White, T. (2003a). *Developing an Immunity to Spam.* GECCO 2003, Chicago, USA. Lecture Notes in Computer Science 2723. Springer-Verlag, pp. 231-242

Oda, T., & White, T. (2003b). *Increasing the Accuracy of a Spam-Detecting Artificial Immune System.* CEC 2003, Canberra, Australia. IEEE Press, pp. 390-396

Oda, T., & White, T. (2005). *Immunity from Spam: An Analysis of an Artificial Immune System for Junk Email Detection.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer, pp. 276-289

OTPE. (2005). *The OTPE Project: Current Maps*. Retrieved July 2005 from http://opte.prolexic.com/maps/

Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). An Ant Colony Algorithm for Classification Rule Discovery. In Abbass, H., Sarker, R. & Newton, C. (Eds.), *Data Mining: a Heuristic Approach* (pp. 191-208): Idea Group Publishing.

Pazzani, M. J. (1996). Searching For Dependencies in Bayesian Classifiers. In Fisher, D. & Lenz, H. (Eds.), *Learning from Data: AI and Statistics V* (pp. 239-248): Springer-Verlag.

Perelson, A. S., & Oster, G. F. (1979). Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self Non-Self Discrimination. *Journal of Theoretical Biology, 81*(4), 645-670.

Porter, M. F. (1997). An Algorithm For Suffix Stripping. In Sparck Jones, K. & Willet, P. (Eds.), *Readings in Information Retrieval*: Morgan Kaufmann.

Pretschner, A., & Gauch, S. (1999). *Personalisation on the Web* (Technical Report ITTC-FY2000-TR-13591-01): Information and Telecommunication Technology Centre, Department of Electrical Engineering and Computer Science, University of Kansas.

Quinlan, J. R. (1987, 1987). *Generating Production Rules From Decision Trees.* 10th International Joint Conference on Artificial Intelligence (IJCAI '87), Milan, Italy. Morgan Kaufmann, pp. 304-307

Quinlan, J. R., & Cameron-Jones, R. M. (1995). *Oversearching and Layered Search in Empirical Learning.* 14th International Joint Conference on Artificial Intelligence IJCAI '95, Montreal, Canada. Morgan Kaufmann, pp. 1019-1024

Rennie, J. D. M. (2000). *ifile: An Application of Machine Learning to Mail Filtering.* KDD-2000 Workshop on Text Mining, Boston, USA.

Robins, M. J., & Garrett, S. M. (2005). *Evaluating Theories of Immunological Memory Using Large Scale Simulations.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 193-206

Rogers, D. J., & Tanimoto, T. T. (1960). A Computer Program for Classifying Plants. *Science, 132*, 1115-1118.

Rowe, N. C. (2005). Exploiting Captions for Web Data Mining. In Scime, A. (Ed.), *Web Mining: Applications and Techniques* (pp. 119-144). London: Idea.

Rozsypal, A., & Kubat, M. (2001). *Using the Genetic Algorithm to Reduce the Size of a Nearest-Neighbour Classifier and to Select Relevant Attributes.* 18th International Conference on Machine Learning (ICML 2001), Massachusetts, USA. Morgan Kaufmann, pp. 449-456

Saltzberg, S. (1991). *Distance Metrics for Instance-Based Learning.* 6th International Symposium on Methodologies for Intelligent Systems (ISMIS-91), Charlotte, USA. Lecture Notes in Computer Science 542. Springer-Verlag, pp. 339-408

Schaffer, C. (1993). Overfitting Avoidance as Bias. *Machine Learning, 10*(2), 153 - 178.

Secker, A., Freitas, A. A., & Timmis, J. (2003a). *AISEC: an Artificial Immune System for E-mail Classification.* Congress on Evolutionary Computation 2003 (CEC2003), Canberra, Australia. 1. IEEE, pp. 131-138

Secker, A., Freitas, A. A., & Timmis, J. (2003b). *A Danger Theory Inspired Approach to Web Mining.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 156-167

Secker, A., Freitas, A. A., & Timmis, J. (2005). Towards A Danger Theory Inspired Artificial Immune System for Web Mining. In Scime, A. (Ed.), *Web mining: Applications and Techniques* (pp. 145-168). London: Idea Group Publishing.

Segal, R. B., & Kephart, J. O. (1999). *MailCat: An Intelligent Assistant for Organizing E-Mail.* 3rd International Conference on Autonomous Agents (Agents '99), Seattle, USA. ACM Press, pp. 276-282

Shahabi, C., & Chen, Y.-S. (2003). *Web Information Personalization: Challenges and Approaches.* 3rd International Workshop on Databases in Networked Information Systems (DNIS 2003), Aizu-Wakamatsu, Japan. pp. 5-15

Sheth, B. D. (1994). *A Learning Approach to personalised Information Filtering.* Masters Dissertation, M.I.T., USA.

Silberschatz, A., & Tuzhilin, A. (1996). What Makes Patterns Interesting in Knowledge Discovery Systems. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 970-974.

Silla, C. N., Kaestner, C. A. A., & Freitas, A. A. (2003). *A Non-Linear Topic Detection Method for Text Summarization Using WordNet.* 1st Workshop on Information Technology and Human Language, Sao Carlos, Brazil. ICMC-USP,

Singh, S., & Thayer, S. (2001). *Immunology Directed Methods for Distributed Robotics: A Novel, Immunity-Based Architecture for Robust Control & Coordination.* SPIE: Mobile Robots XVI.

Smith, G. D. (2005). *Meta-Heuristics in the KDD Process.* 1st UK Knowledge Discovery in Databases Symposium, Liverpool, England. pp. 35-41

Sompayrac, L. (1999). *How the Immune System Works*: Blackwell Science.

Song, R., Liu, H., Wen, J. R., & Ma, W. Y. (2004). Learning Important Models for Web Page Blocks Based on Layout and Content Analysis. *SIGKDD Explorations, 6*(2), 14 - 23.

SpamAssassin. (2006). *The Apache SpamAssassin Project*. Retrieved June 2005 from http://spamassassin.apache.org/

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations, 1*(2), 12-23.

Stanfill, C., & Waltz, D. (1986). Toward Memory-based reasoning. *Communications of the ACM, 29*(12), 1213-1228.

Steinman, L. (1994). Autoimmune Disease. In *Life, Death and the Immune System* (pp. 75-86): Scientific American.

Stepney, S., Smith, R. E., Timmis, J., & Tyrrell, A. M. (2004). *Towards a Conceptual Framework for Artificial Immune Systems.* 3rd International Conference on Artificial Immune Systems (ICARIS 2004), Catania, Sicily. Lecture Notes in Computer Science 3239. Springer, pp. 53-64

Stibor, T., Bayarou, K., & Eckert, C. (2004). An Investigation into r-chunk Detector Generation on Higher Alphabets. In (Vol. 3102, pp. 26-30): Springer-Verlag.

Stibor, T., Mohr, P., & Timmis, J. (2005a). *Is Negative Selection Suitable for Anomaly Detection?* GECCO 2005, Washington, USA. ACM Press, pp. 321-328

Stibor, T., Timmis, J., & Eckert, C. (2005b). *A Comparative Study of Real-Valued Negative Selection to Statistical Anomaly Detection Techniques.* 4th

International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 262-275

Sun, A., Lim, E. P., & Ng, W. K. (2002). *Web Classification Using Support Vector Machine.* 4th International Workshop on Web information and Data Management, McLean, USA. ACM Press, pp. 96 - 99

Suzuki, E., & Zytkow, J. M. (2000). *Unified Algorithm for Undirected Discovery of Exception Rules.* 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2000), Lyon, France. Lecture Notes in Computer Science 1910. Springer-Verlag, pp. 169-180

Tan, P. N., Kumar, V., & Srivastava, J. (2002). *Selecting the Right Interestingness Measure For Association Patterns.* 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada. ACM Press, pp. 32-41

Tang, N., & Vemuri, V. R. (2005). *An Artificial Immune System Approach to Document Clustering.* 20th ACM Symposium on Applied Computing (SAC 2005), Santa Fe, USA. ACM Press, pp. 918 - 922

Tanudjaja, F., & Mui, L. (2002). *Persona: A Contextualized and Personalized Web Search.* 35th Annual Hawaii International Conference on System Sciences (HICSS'02), Big Island, Hawaii. pp. 53-

Tarakanov, A. O., Skormin, V. A., & Sokolova, S. P. (2003). *Immunocomputing*: Springer.

Tauber, A. (2002). *The Biological Notion of Self and Non-self.* Retrieved November 2002 from http://plato.stanford.edu/archives/sum2002/entries/biology-self/

Taylor, D. W., & Corne, D. W. (2003). *An Investigation of Negative Selection for Fault Detection in Refrigeration Systems.* ICARIS 2003, Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer Verlag,

Timmis, J. (2000). *Artificial Immune Systems: A Novel Data Analysis Technique Inspired by The Immune Network Theory.* Ph.D. Thesis, University of Wales, Aberystwyth, Wales.

Timmis, J. (2001). *aiVIS: Artificial Immune System Visualisation.* EuroGraphics UK, London, UK. pp. 61-69

Timmis, J., Edmonds, C., & Kelsey, J. (2004). *Assessing the Performance of Two Immune Inspired Algorithms and a Hybrid Genetic Algorithm for Function Optimisation.* Congress on Evolutionary Computation (CEC 2004), Portland, USA. IEEE Press, pp. 1044-1051

Timmis, J., & Neal, M. (2000). *Investigating the Evolution and Stability of a Resource Limited Artificial Immune System.* GECCO 2000 Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA. pp. 40-41

Timmis, J., & Neal, M. (2001). A Resource Limited Artificial Immune System for Data Analysis. *Knowledge Based Systems, 14*(3-4), 121-130.

Timmis, J., Neal, M., & Hunt, J. (2000). An Artificial Immune System for Data Analysis. *Biosystems, 3*(1), 143-150.

Tufis, D., & Mason, O. (1998). *Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger.* 1st International Conference on Language Resources and Evaluation, Granada, Spain. pp. 589-596

Twycross, J. (2002a). *An Immune System Approach to Document Classification.* Masters Dissertation, University of Sussex, UK.

Twycross, J. (2002b). *An Immune System Approach to Document Classification* (HP Labs Technical Reports HPL-2002-288): HP Labs Bristol, UK.

Twycross, J., & Aickelin, U. (2005). *Towards a Conceptual Framework for Innate Immunity.* 4th International Conference on Artificial Immune Systems (ICARIS 2005), Banff, Canada. Lecture Notes In Computer Science 3627. Springer-Verlag, pp. 112-125

Twycross, J., & Cayzer, S. (2002). *An Immune System Approach to Document Classification* (HP Labs Technical Reports HPL-2002-292): HP Labs Bristol, UK.

Twycross, J., & Cayzer, S. (2003). *An Immune-based Approach to Document Classification.* International Conference on Intelligent Information Processing and Web Mining 2003, Zakopane, Poland. Springer-Verlag, pp. 33-46

Varela, F. J., Coutinho, A., Dupire, B., & Vaz, N. N. (1988). Cognitive Networks: Immune, Neural and Otherwise. In Perelson, A. S. (Ed.), *Theoretical Immunology* (Vol. 2, pp. 359-375): Addison Wesley.

Voorhees, E. M. (1993). *Using WordNet to Disambiguate Word Senses for Text Retrieval.* 16th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, USA. ACM Press, pp. 171-180

Voorhees, E. M. (1998). Using WordNet for Text Retrieval. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database* (pp. 285-303). Cambridge: MIT Press.

Wantanabe, Y., Ishiguro, A., & Uchikawa, Y. (1999). Decentralised Behaviour Arbitration Mechanism for Autonomous Mobile Robots Using Immune Network. In Dasgupta, D. (Ed.), *Artificial Immune Systems and Their Applications* (pp. 187-209): Springer-Verlag.

Watkins, A. (2001). *AIRS: A Resource Limited Artificial Immune Classifier.* Masters Dissertation, Mississippi State University, MS. USA.

Watkins, A. (2005). *Exploiting Immunological Metaphors in the Development of Serial, Parallel, and Distributed Learning Algorithms.* PhD. Thesis, University of Kent, Canterbury, England.

Watkins, A., & Boggess, L. (2002a). *A New Classifier Based on Resource Limited Artificial Immune Systems.* Congress on Evolutionary Computation (CEC 2002), Honolulu, USA. IEEE Press, pp. 1546-1551

Watkins, A., & Boggess, L. (2002b). *A Resource Limited Artificial Immune Classifier.* Congress on Evolutionary Computation (CEC 2002), Honolulu, USA. IEEE, pp. 926-931

Watkins, A., & Timmis, J. (2002). *Artificial Immune Recognition System (AIRS): Revisions and Refinements.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 173-181

Watkins, A., & Timmis, J. (2004). *Exploiting Parallelism Inherent in AIRS.* 3rd International Conference on Artificial Immune Systems (ICARIS 2004),

Catania, Sicily. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 427-438

Watkins, A., Timmis, J., & Boggess, L. (2004). Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. *Genetic Programming and Evolvable Machines, 3*(5), 291-317.

Webb, G. I. (1994). *Generality is More Significant than Complexity: Toward an Alternative to Occam's Razor.* 17th Australian Joint Conference on AI, Armidale, Australia. World Scientific, pp. 60-67

Webb, G. I. (1996). Further Experimental Evidence Against the Utility of Occam's Razor. *Journal of Artificial Intelligence Research, 4*, 397-417.

Weiss, S. M., Indurkhya, N., Zhang, T., & Damerau, F. J. (2005). *Text Mining: Predictive Methods for Analysing Unstructured Information*: Springer-Verlag.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer Systems that Learn*: Morgan Kaufmann.

Weissman, I. L., & Cooper, M. D. (1994). How the Immune System Develops. In *Life, Death and the Immune System* (pp. 15-25): Scientific American.

White, J. A., & Garret, S. M. (2003). *Improved Pattern Recognition with Artificial Clonal Selection.* 2nd International Conference on Artificial Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 181-193

Widyantoro, D. H., *et al.* (1999). *Alipes: A Swift Messenger in Cyberspace.* 1999 AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, USA. pp. 60-67

Wierzchon, S. T., & Kuzelewska, U. (2002). *Stable Cluster Formation in an Artificial Immune System.* 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK. pp. 68-75

Witten, I. H., & Frank, E. (1999a). Decision Trees. In *Data mining: Practical machine learning tools and techniques with Java implementations* (pp. 58-59). San Francisco, CA: Morgan Kaufmann.

Witten, I. H., & Frank, E. (1999b). Divide and Conquer: Constructing Decision Trees. In *Data mining: Practical Machine Learning Tools and Techniques With Java Implementations* (pp. 89-97). San Francisco, CA.: Morgan Kaufmann.

WordNet. (2004). *WordNet: a lexical database for the English language.* Retrieved September 2004 from http://wordnet.princeton.edu/

Wright, S. (2002). *Personalisation, How a Computer Can Know You Better Than Yourself.* 3rd Multimedia Systems Conference, Southampton, UK.

Wu, F., & Hsu, C. (2005). Using Context Information to Build a Topic Specific Crawling System. In A.Scime (Ed.), *Web Mining: Applications and Techniques* (pp. 50-68). London: Idea.

Yahoo! (2005). *Yahoo! Directory.* Retrieved July 2005 from http://dir.yahoo.com/

Yahoo! (2006). *Yahoo! Search Homepage.* Retrieved January 2006 from http://search.yahoo.com

Yang, J., & Park, S.-Y. (2002). Email Categorization Using Fast Machine Learning Algorithms. *Discovery Science 2002*, 316-323.

Yang, Q., Zhang, H. H., & Li, T. (2001). *Mining Web Logs for Prediction Models in WWW Caching and Prefetching.* 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA. ACM Press, pp. 473-478

Yi, J., & Sundaresan, N. (2000). *A Classifier for Semi-Structured Documents.* 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000), Boston, USA. ACM Press, pp. 340 - 344

Zeidenberg, M. (1990). *Neural Network Models in Artificial Intelligence*: Ellis Horwood.

Zhang, J. (1992). *Selecting Typical Instances in Instance-Based Learning.* 9th International Conference on Machine Learning (ICML 2000), Aberdeen, Scotland, UK. Morgan Kaufmann, pp. 470-479

Zhou, C., Xiao, W., Tirpak, T. M., & Nelson, P. C. (2003). Evolving Accurate and Compact Classification Rules with Gene Expression Programming. *IEEE Transactions on Evolutionary Computation, 7*(6), 519-531.

Appendix A Stopwords

ABOUT, ALL, AMONG, AN, AND, ARE, AS, AT, BE, BEEN, BETWEEN, BOTH, BUT, BY, DO, DURING, EACH, EITHER, FOR, FOUND, FROM, FURTHER, HAS, HAVE, HOWEVER, IF, IN, INTO, IS, IT, ITS, MADE, MAKE, MANY, MORE, MOST, MUST, NO, NOT, OF, ON, OR, SAME, SEVERAL, SOME, SUCH, THAN, THAT, THE, THEIR, THESE, THEY, THIS, THOSE, THROUGH, TO, TOWARD, UPON, USED, USING, WAS, WERE, WHAT, WHICH, WHILE, WHO, WILL, WITH, WITHIN, WOULD.

Appendix B Roulette Wheel Selection

The following appendix briefly describes roulette wheel selection. As this section is somewhat succinct, the interested reader is directed towards the evolutionary algorithms literature such as (Freitas, 2002) for more information.

Roulette wheel selection is a fitness proportional selection strategy commonly used in evolutionary algorithms. The essential idea of selection throughout all evolutionary algorithms is that the highest quality individuals have a higher probability of being selected for reproduction and so a higher probability of passing genetic information to future generations. Thus it is an analogue to the Darwinian "survival of the fittest" process.

Given a set of individuals in a population, the roulette wheel selection mechanism will choose individuals with a frequency directly proportional to that individual's weight in the population (where weight could be fitness, affinity, etc.). The technique can be explained as follows: all members of a population are given a section of a contiguous line, with the length of the segment in proportion to the individual's fitness. A random point on the line is chosen and the individual whose section spans that chosen point is selected.

This gives rise to a number of issues as enumerated in (Freitas, 2002):

1. It is assumed all individuals have a non-negative weighting
2. It assumes fitness is to be maximised
3. It requires the computation of a global sum of finesses over the whole population, thus limiting potential for distribution.
4. If one individual has a weight significantly greater than the other members of the population, this one individual will tend to dominate.
5. Conversely, if all members of the population have similar weights, each has roughly the same change of selection and so the choice can be almost random.

B.1 Pseudocode

Pseudocode B.1 describes this selection technique. Given a population P of individuals, i, each with an associated weight i_w The sum of individual fitness's is determined (S). A uniformly distributed random number, R, in the range 0 to S is generated. Each member of the population is inspected one by one. When the sum of the finesses of the members of the population to that point is greater than R, select the current individual.

```
1   S ← 0
2   FOREACH (I ∈ P)
3     S ← S + iw
4   R ← random number in range [0,S]
5   X ← 0
6   FOREACH(I ∈ P)
7    X ← X + iw
8    IF (X>R) THEN select i
9   END
```

Pseudocode B.1. Roulette wheel selection

References

Freitas. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*: Springer.

Appendix C Statistical Testing

In this appendix, two separate measures for statistical comparison of the means two sets of observations are briefly described. More specifically, the application of one of the described tests will determine a probability that a null hypothesis - the means of two sets of normally distributed populations are equal - will hold. This situation is shown diagrammatically in Figure C.1. The curves in Figure C.1 are derived from samples (observations) of two populations, they are not constructed from two separate entire populations, as in this latter case there is no possibility that the means are equal. In Figure C.1, curve A shows two distributions that are not likely to have equal means, while B shows a pair of distributions that are much more likely to have equal means.

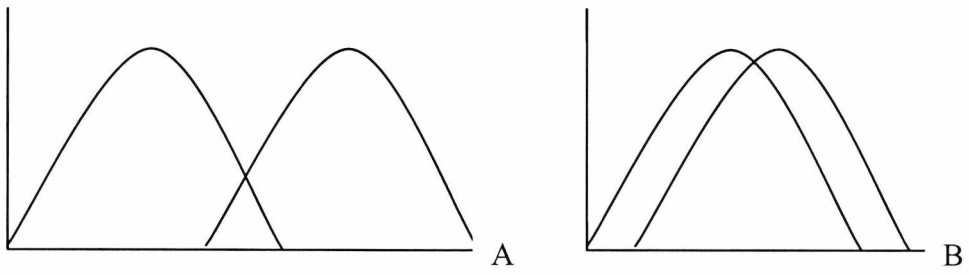

Figure C.1. Example distributions.

Of the two tests described later in this appendix, Student's t-test is suitable for situations where few (less than 30) observations are made, while the standard test of significance (*z* test) is more appropriate for larger sample sizes..

It is convention to declare a null hypothesis, that *the means of two sets of observations are equal*, and declare that this null hypothesis will be rejected only if the probability that it is true (denoted by P_{null}) is smaller than 0.05. This value for declaring success is not formal but rather is an almost universally held convention. Thus, if it is found the difference between means of the populations will occur by chance with a probability less than 0.05 then the result is said to be significant ($P_{null} < 0.05$). If the

distributions are found to occur by chance with probability less than 0.01 ($P_{null} < 0.01$) then the result is said to be highly significant.

A one-tailed test only tests whether a value is particularly higher or partially lower than expected. Whereas a two tailed test determines whether the deviation from a value is unexpected and a deviation can occur either side of an expected result. This is shown diagrammatically in Figure C.2. As the calculations in this thesis tend to determine whether the means of two populations are significantly different (higher or lower, it doesn't matter), a two-tailed test is often the correct option. (One-tailed and two-tailed tests are sometimes referred to as one-sided and two-sided bounds as found in (Mitchell, 1997)).

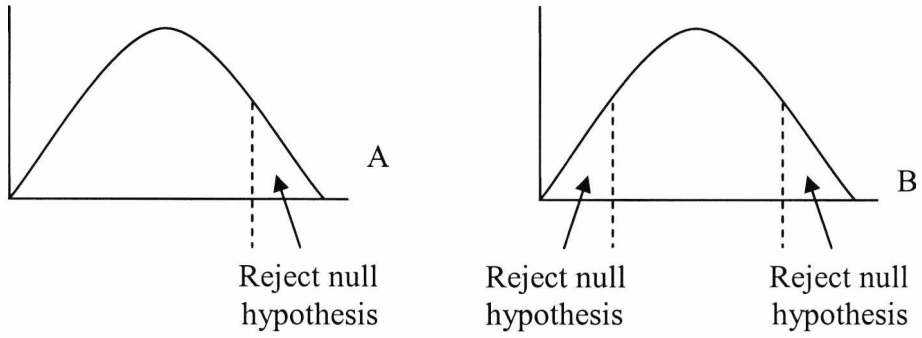

Figure C.2 Representation of one-tail (A) and two-tail (B) tests.

For the purposes of this thesis, t-test calculations are computed using Microsoft Excel's built-in t-test function (note, the built-in statistical function was used, not the optional add-in). This function will return a P_{null} value for any two sets of values without the need for manual calculation or lookup tables. Likewise, given a z value when using the large sample method, Excel's `normsdist` function may be used to determine a probability value for a corresponding z value. The use of Excel is preferred for convenience, efficiency and accuracy.

C.1 Student's t-test

Student's t-test is a statistical test of significance found frequently in both statistics (Alder & Roessler, 1968; Creighton, 1994) and data mining texts (Mitchell, 1997; Ayara, 2005; Tan et al., 2005; Watkins, 2005). W.M. Gossett first described the test in 1908 when researching quality control methods for his employer, the Guinness brewery in Dublin. As Guinness did not allow employees to publish in-house research he published his method under the pseudonym: "A. Student". The resultant value referred

to throughout the paper was denoted "t" and thus the name "Student's t-test" was coined (Creighton, 1994).

The t-test is most suited to a situation where a small number, typically less than 30, observations have been made. For larger sample sizes the use of the normal distribution (section C.2) is preferable, however, there is no harm in using the t-test for larger sample sizes, as when n>30, the shape of the t-distribution will approximate the normal distribution.

There exist two types of t-test: the independent and the paired tests. In the paired test, each observation in one set must correspond to an observation in the second set. This type of t-test is applicable to situations where, for example, the state of an object is measured at two separate points in time. The independent t-test places no such restriction on the observations and is the one used throughout this thesis. The variances of the two populations used in a t-test should strictly be equal, but this assumption is very often dropped for simplicity (Alder & Roessler, 1968).

C.1.1 Independent t-test

The independent t-test is defined in (Alder & Roessler, 1968) shown in Equation C.1.

$$t = \frac{\left(\overline{X} - \overline{Y}\right) - m_{\overline{x}-\overline{y}}}{s_{\overline{x}-\overline{y}}} \qquad \text{C.1}$$

Where $\overline{X}$ and $\overline{Y}$ are the means of each sample (set of observations of the value of each variable). As the null hypothesis under test is that these samples have identical mean values, the difference between the mean values ($m_{\overline{x}-\overline{y}}$) is set to 0. The value of the term $s_{\overline{x}-\overline{y}}$ is determined as shown in Equation C.2.

$$\begin{aligned} s_{\overline{x}-\overline{y}} &= \sqrt{s_{\overline{x}}^{\,2} + s_{\overline{y}}^{\,2}} \\ s_{\overline{x}} &= \frac{s}{\sqrt{n_1}} \\ s_{\overline{y}} &= \frac{s}{\sqrt{n_2}} \end{aligned} \qquad \text{C.2}$$

Where n_1 is the sample size (number of observations) of X and n_2 is the sample size of Y while s is given by Equation C.3.

$$s = \sqrt{\frac{\sum (X - \bar{X})^2 + \sum (Y - \bar{Y})^2}{n_1 + n_2 - 2}} \qquad \text{C.3}$$

C.1.2 Degrees Of Freedom and Final Value

Once the t-value has been computed, the associated probability value regarding the null hypothesis can be determined from a lookup table, however one extra stage is required. The number of degrees of freedom for the sample size must be calculated. The degrees of freedom shape the t-distribution based on the sample size used to determine the standard deviation. Informally, it allows the final probability value corresponding to a t-value to change based on the sample size. Thus, empirically, it can be seen that it can be easier to reject the null hypothesis as the number of observations increases. The number of degrees of freedom (ν) is computed as shown in Equation C.4, where n is the number of samples in a population.

$$\nu = n_1 + n_2 - 2 \qquad \text{C.4}$$

Once the t-value and value for ν have been computed, the resulting probability (alpha) value can be determined by using a lookup table. Given the probability value (α) for a one-tailed test, the value for a two tailed test can be computed as shown in Equation C.5.

$$P_{null} = 2(1 - \alpha_{t,\nu}) \qquad \text{C.5}$$

C.2 Large Sample Method

While the t-test is suited to small sample sizes, the normal distribution can be employed for large samples. That is, the standard normal distribution z score can be computed by Equation C.6 (Alder & Roessler, 1968) where $\bar{X}$ and $\bar{Y}$ are the mean values of observations of the variables X and Y while $\sigma_{\bar{x}}$ and $\sigma_{\bar{y}}$ are the standard deviations of the means and n_1 and n_2 are the numbers of observations in the set of observed values of variables X and Y respectively. Note, the term m_1-m_2 included in the equations presented in (Alder & Roessler, 1968), page 129, can be removed as in this case, the test is used to determine the probability that the means are equal, i.e. m_1-m_2 = 0 and as such has no effect. It can therefore be seen to be analogous to Equation C.1.

$$z = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{\sigma_{\overline{x}}^{\,2}}{n_1} + \frac{\sigma_{\overline{y}}^{\,2}}{n_2}}} \qquad \text{C.6}$$

The resultant probability value for z can then be determined in the usual manner, from a lookup table. The resultant probability will almost always give the probability of one sample mean being significantly more (or less as the normal distribution is symmetrical) than the other sample mean. Therefore, if a two-tailed test is required, taking advantage of the symmetry of the distribution it is straightforward to determine the probability of the null hypothesis holding by Equation C.7.

$$P_{null} = 2(1 - \alpha_z) \qquad \text{C.7}$$

References

Alder, H. L., & Roessler, E. B. (1968). *Introduction to Probability and Statistics*: W. H. Freeman.

Ayara, M. (2005). *An Immune Inspired Approach For Adaptable Error Detection in Embedded Systems.* Ph.D. Thesis, University of Kent, Canterbury, England.

Creighton, J. H. (1994). *A First Course in Probability Models and Statistical Inference*: Springer-Verlag.

Mitchell, T. M. (1997). *Machine Learning*: McGraw-Hill.

Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*: Addison Wesley.

Watkins, A. (2005). *Exploiting Immunological Metaphors in the Development of Serial, Parallel, and Distributed Learning Algorithms.* PhD. Thesis, University of Kent, Canterbury, England.

Appendix D Investigation into TFIDF Ranking

This investigation was required to support the implementation decision described in section 5.4.5.3. That is, this investigation is required to determine whether a heuristic, as used in this thesis, is valid. The heuristic is that the most frequent 500 words on a web page are likely to include the top 50 according to a ranking by their TFIDF values.

Recall, to compute the TFIDF of a word using the technique set out in section 5.4.5.2 requires that the word be submitted to Google to thus determine the word's document frequency. This is a very time consuming operation, taking approximately 2 seconds, to complete for each word[6]. Secondly, Google restricts the number of automated searches any one user can make per day. It is therefore of importance that the number of words submitted to Google is minimised while having (almost) no impact on the results.

D.1 Investigation Method

30 different web pages were used for this test. These web pages were picked in a random manner from the internet with 5 of these pages were chosen by using the "random article" feature on Wikipedia (Wikipedia, 2006). The set of pages is more likely to contain content pages rather than navigation pages or links pages, although both are represented in the set. Any page with less than or equal to 50 unique words was discarded as this will trivially contain the top 50 words. The pages were pre-processed according to the method consistent throughout this thesis (stopwords, punctuation and

[6] Although this can take considerably longer. As Google's automated system continues to grow in popularity it is under ever increasing load. If a query is submitted to an already overloaded server the server will reject the request, the client should not retry the query for 30 seconds after this rejection greatly increasing the mean request time during certain times of the day.

all html removed, text tokenised, etc.). The term frequency of each unique word was determined, then the TFIDF weight for each word was determined using Equation 5.4.

By sorting both the TF and TFIDF values in descending order it is possible to take the top 50 words as ranked by TFIDF and search the sorted TF list for each word. A record is made of the position of each of those top 50 words (by TFIDF rank) in the list in decreasing order of TF value.

D.2 Results and Discussion

Table D.1 shows the results obtained. The column "Lowest Position of Word" shows how many words as ranked by TF must be examined to include all words in the top 50 when ranked by TFIDF. "Unique Words" shows the number of unique words on that web page, while "Percentage of Words" is the percentage of unique words that must be examined to ensure the top 50 words (ranked by TFIDF) are included in the list ranked by TF.

	Lowest Position of Word	Unique Words	Percentage of Words
1	232	320	73%
2	113	119	95%
3	404	423	96%
4	312	339	92%
5	151	2232	7%
6	168	186	90%
7	142	833	17%
8	244	1305	19%
9	173	192	90%
10	108	1146	9%
11	117	757	15%
12	122	638	19%
13	59	1546	4%
14	244	837	29%
15	266	747	36%
16	123	1467	8%
17	155	542	29%
18	352	433	81%
19	139	241	58%
20	253	621	41%
21	220	667	33%
22	191	194	98%
23	269	438	61%
24	118	343	34%
25	130	835	16%
26	275	282	98%
27	242	249	97%
28	288	311	93%
29	148	540	27%
30	461	502	92%

Table D.1. Full results of TFIDF test

It can be seen that a wide variety of page lengths were used, ranging from 192 to 1546 words. The results suggest that using a given proportion of the words ranked by TF to infer the top 50 when ranked by TFIDF is unlikely to give satisfactory results. It was hoped that an efficient manner in which to determine the number of words to examine to ensure the top 50 by TFIDF are included would be to use a proportion of those on the webpage, the actual proportion being determined by some function. However, this does not appear to be possible and so is ignored for the remainder of this section.

Statistic	Number of TF Ranked Words	Unique Words	Percentage of Words
Mean	207.30	642.83	51.89%
Standard Deviation	94.44	482.59	35.27
Maximum	461	2232	98%
Minimum	59	119	4%

Table D.2. Summarised Results

It can be seen that none of the values in the "Lowest Position of Word" column are greater then or equal to 500, meaning the heuristic used did not fail during the tests. Using the figures in Table D.2 it is possible to determine whether the use of the top 500 words as ranked by TF is likely to include the top 50 words when ranked by TFIDF. This can be done using a statistical significance test based on the normal distribution to determine the probability of an entry in the "Number of TF ranked words" column having a value of 500 or more. The null hypothesis is that a value of 500 or greater is likely to appear in the "50 words" column. The value of z in this case if given by the following calculation.

$$z = \frac{X - \mu}{\sigma}$$

$$z = \frac{500 - 207.3}{94.44}$$

$$z = 3.1$$

$$P_{null} < 0.01$$

Using Microsoft Excel to lookup the associated probability value for this z value, the probability of the null hypothesis holding is found to be less than 0.01. Thus the null hypothesis can be rejected. In fact it can be calculated that the heuristic adopted will fail once in approximately 1,031 pages.

D.3 Summary

In reality, the probability of the null hypothesis holding is likely to be an overestimate, i.e. the heuristic is more robust than this figure may suggest. The sample of pages includes an artificially high frequency of long pages. Most of the pages examined were content pages and thus contain rather a lot of text. Submitting pages such as the Google homepage (Google, 2006), which contains only 25 unique words, would not have tested the heuristic in the correct situation. So of course pages with fewer than 50 words can be represented in a list of up to 50 words.

Based on the probability of the null hypothesis holding as derived above, the heuristic used is believed to be a reasonable one as one failure in over 1,000 pages is quite acceptable for the purposes of this investigation.

URLs of the Web Pages from Table D.1

1 http://en.wikipedia.org/wiki/Artificial_immune_system
2 http://www.cs.kent.ac.uk/archive/people/staff/jt6/aislinks.html
3 http://www.amazon.co.uk
4 http://news.bbc.co.uk/
5 http://ca.expasy.org/links.html
6 http://www.cs.kent.ac.uk
7 http://dilbertblog.typepad.com/
8 http://www.cs.kent.ac.uk/people/rpg/ads3/temp/mirrored/enders.htm
9 http://www.ebay.co.uk
10 http://www.cs.kent.ac.uk/people/rpg/ads3/temp/mirrored/ethernet.htm
11 http://en.wikipedia.org/wiki/Formula_One_regulations
12 http://news.google.co.uk
13 http://rinkworks.com/stupid/cs_abuse.shtml
14 http://www.jibble.org
15 http://blog.jstott.me.uk/
16 http://en.wikipedia.org/wiki/Led_zeppelin
17 http://en.wikipedia.org/wiki/Ad_hominem
18 http://maddox.xmission.com/
19 http://www.socialresearchmethods.net/kb/stat_t.htm
20 http://www.cs.kent.ac.uk/~ads3
21 http://www.camerasunderwater.co.uk/info/starting.html
22 http://www.bbc.co.uk/radio1/chrismoyles
23 http://www.scubaboard.com/archive/index.php/f-15.html
24 http://news.bbc.co.uk/sport1/hi/other_sports/snooker/4434545.stm
25 http://www.statsoft.com/textbook/esc.html
26 http://www.theregister.co.uk/2006/01/17/google-earth-investigation/
27 http://www.trnmag.com/index.html
28 http://www.thinkgeek.com
29 http://en.wikipedia.org/wiki/Underwater_hockey
30 http://en.wikipedia.org/wiki/Main_Page

References

Google. (2006). *Google: Google Search Homepage*. Retrieved January 2006, from http://www.google.com

Wikipedia. (2006). *Wikipedia Front Page*. Retrieved January 2006, from http://www.wikipedia.org

Appendix E User Test URLs

User 1: Sites about bioinformatics

http://www.biowww.net/biobooks_1_data-mining-bioinformatics.html
http://kd.cs.uni-magdeburg.de/ws03.html
http://users.aber.ac.uk/afc/research.html
http://www.cs.rpi.edu/~zaki/BIOKDD04/

User 2: Sites about IPTV

http://www.microsoft.com/tv/default.mspx
http://en.wikipedia.org/wiki/IPTV
http://www.iptv-news.com/
http://www.iptv-forum.com/2006/

User 3: Sites about mathematically finding the areas of shapes.

http://mathworld.wolfram.com/PolygonArea.html
http://mathforum.org/library/drmath/view/64552.htm
http://www.geovista.psu.edu/sites/geocomp99/Gc99/076/gc_076.htm
http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u33.html
http://www.attewode.com/Calculus/AreaMeasurement/area.htm

User 4: Sites about graph drawing

http://www.gd2003.org/cfp.html
http://en.wikipedia.org/wiki/Graph_drawing
http://rw4.cs.uni-sb.de/users/sander/html/gstools.html
http://en.wikipedia.org/wiki/Graph_%28mathematics%29
http://www.cs.brown.edu/people/rt/gd.html
http://www.ics.uci.edu/~eppstein/gina/gdraw.html

User 5: Sites about trans-membrane proteins

http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/C/CellMembranes.html
http://web.mit.edu/esgbio/www/cb/membranes/proteins.html
http://www.chembio.uoguelph.ca/educmat/phy456/456lec03.htm
http://www.gpcr.org/7tm/htmls/GPCR.html
http://en.wikipedia.org/wiki/Transmembrane_proteins

User 6: Sites about Markov chains

http://www.staff.city.ac.uk/r.j.gerrard/courses/2dsm/dsm03_4.htm
http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/Summary8.html
http://cs.bilgi.edu.tr/~bulent/MarkovChains.html
http://tecfa.unige.ch/~lemay/thesis/THX-Doctorat/node55.html
http://citeseer.ist.psu.edu/context/84837/0

User 7: Sites about Java OpenGL

```
http://www.javaworld.com/javaworld/jw-05-1999/jw-05-media.html
http://opengl.j3d.org
http://www.opengl.org/resources/java/
http://www.codeproject.com/java/opengl.asp
http://www.culturekitchen.com/potatoland/archives/002334.html
http://home.earthlink.net/~rzeh/YAJOGLB/doc/YAJOGLB.html
```

User 8: Sites about swarm intelligence

```
http://www.swarm-bots.org/
http://www.swarmintelligence.org/
http://goanna.cs.rmit.edu.au/~xiaodong/cec04-swarm/
http://www.cs.kent.ac.uk/projects/swarm/
http://www.computelligence.org/sis/
```

User 9: Sites about Prokofiev, a Russian composer

```
http://www.bbc.co.uk/music/profiles/prokofiev.shtml
http://www.prokofiev.org/biography/conserv.html
http://www.geocities.com/Vienna/1891/op25.html
http://www.siue.edu/~aho/musov/sergei.html
http://www.goldsmiths.ac.uk/departments/music/
prokofiev-archive/index.php
http://www.tiscali.co.uk/reference/encyclopaedia/hutchinson/
m0012662.html
```

User 10: Sites about antigravity

```
http://jnaudin.free.fr/lifters/main.htm
http://tesladownunder.iinet.net.au/Lifters.htm
http://www.antigravity.org/BigSpinModelOfGravity.html
http://www.space.com/businesstechnology/
060215_technovel_antigravity.html
http://www.wired.com/news/technology/0,1282,52432,00.html
```

User 11: Sites about Montessori schooling

```
http://members.vienna.at/wolschner/children.html
http://www.montessori.ac.uk/
http://www.montessori-ami.org/
http://www.montessori.org.uk/
http://www.pgmontessori.ca/index.html
```

User 12: Sites about the World of Warcraft computer game

```
http://www.wow-europe.com/en/
http://www.worldofwar.net/
http://www.thottbot.com/
http://wow.allakhazam.com/
http://www.worldofwarcraft.com/burningcrusade/
http://www.gamespot.com/pc/rpg/worldofwarcraftexp1/
news.html?sid=6136860&mode=previews
http://en.wikipedia.org/wiki/
World_of_Warcraft:_The_Burning_Crusade
http://www.1up.com/do/previewPage?cId=3145157&did=1
```

User 13: Sites about the Neverwinter Nights computer game

http://www.gamebanshee.com/neverwinternights/
http://nwvault.ign.com/
http://www.planetneverwinter.com/
http://nwn.bioware.com/underdark/character_builds.html
http://www.nwnwiki.org/

User 14: Sites about extreme unicycling

http://www.unicycling.org/unicycling/mounts/
http://www.unicycling.org/unicycling/faq.html
http://www.unicycle.uk.com/FAQ.asp?iCategory=65&FAQParentID=34
http://www.unicycling.com/brett/e_muni/faq.html
http://www.unicycling.com/muni/

User 15: Sites about star formation

http://www.sr.bham.ac.uk/nam2005/
full_programme.html#thurs_plenary_am
http://www.star.ucl.ac.uk/groups/molap/
http://www.astro.cf.ac.uk/groups/starform/
http://www.ipac.caltech.edu/Outreach/Edu/sform.html
http://eaa.iop.org/index.cfm?action=summary&doc=eaa%2F1872%40
eaa-xml
http://www.space.com/scienceastronomy/astronomy/spacedust.htm