AISIID: an artificial immune system for interesting information discovery on the web

Andrew Secker[a], Alex A. Freitas[a] and Jon Timmis[b]

a. Computing Laboratory, University of Kent, Canterbury, Kent, UK.. {a.d.secker, a.a.freitas}@kent.ac.uk

b. Department of Computer Science and Department of Electronics, University of York, Heslington, York, UK.. jtimmis@cs.york.ac.uk

Elsevier use only: Received date here; revised date here; accepted date here

Abstract

There exist numerous systems for mining the web in search of relevant information but few exist for the discovery of interesting information. 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 article investigates the use of Artificial Immune Systems (AIS) applied to discovery of interesting information. AIS are thought to confer the adaptability and learning required for this task. AISIID (Artificial Immune system for Interesting Information Discovery) is described in some detail, then an evaluative study is undertaken involving the subjective evaluation of the results by users. AISIID is found to discover pages rated more interesting by users than a comparative system.

Keywords: Web Content Mining; Artificial Immune System; Interesting; Information; AISIID

1. Introduction

With the ever growing wealth information on the internet, effective tools for distinguishing between interesting and non-interesting material are becoming essential. 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 subject of the discovery of interesting information on the web is practically absent in the research literature, with the exception of the WebCompare system [25], yet we consider it an important issue. While the results returned from a web page search will often be relevant, in the sense that they will contain the original search terms (allowing for the ambiguity of language), the user may be overwhelmed with an unmanageable number of search results. In addition to this, there exists a paradox with regard to current keyword based search techniques and the discovery of interesting information. Interesting information tends to be 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. The goal of this research is to construct a system to mine 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 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.

Web content mining can supply a user with the information that they seek, 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. To cope with these problems, 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.

The implementation of computer algorithms based on immune principles and components, or Artificial Immune Systems (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 large collection of domains [10]. 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 [9]. 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.

In this article, the web mining system AISIID (Artificial Immune system for Interesting Information Discovery) is described as an investigation into an AIS-based mechanism for the discovery of interesting information on the web. This is then compared in a subjective user test against the only known comparable system, WebCompare.

1.1. Motivation

Before continuing it is worth motivating this research by expanding our reasons for believing web page interestingness is important when mining information. Traditional keyword-based search techniques, both on the internet and otherwise, contain an inherent problem: the results obtained are likely to supply the user mainly with 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. Current search techniques do not cater for this scenario, as the very nature of the method by which users currently specify search criteria is at odds with it. 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. 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.

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 [19]. 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. This work takes the view of Liu et.al. [25] and hypothesise that a user will find a web page interesting if that page is relevant to the user's search and also fulfils at least one of the following criteria. The content of the page is:

1. Novel
2. Surprising
3. Unexpected

To expand, 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 in 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.

To this end, the lexical database WordNet (discussed later) can be employed to generate words that may be novel, surprising and unexpected to a user, and therefore 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 [25]), and the first to do this in an adaptable manner.

1.1.1. Why Use Immune Inspiration for Web Mining?

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 [10] describes these properties which are in turn based on the work in [9]. 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.
4. **Self-organization**: Well designed AIS may continually 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 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. Noise tolerance is a characteristic of many 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 artificial 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 [34]. 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.

1.2. Background

This section briefly introduces some concepts important to the rest of this paper. Due to space constraints only the basic information has been given with the reader being referred to the literature for a more in-depth review.

1.2.1. Artificial Immune Systems

This short explanation of immune systems, natural and artificial, explains the most important concepts and components to allow for an understanding of the algorithm presented later in this document. For a comprehensive review of the biology and inspiration behind artificial immune systems, the reader is direct towards literature such as [29] and [10].

Artificial immune systems are defined by de Castro and Timmis as "adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving" [10]. The immune system is a vast, complex, interconnected network of agents and processes. While the innate immune system is of great importance to our wellbeing, it is the adaptive immune system that most AIS algorithms take inspiration from. As its name would suggest, the adaptive immune system may change and adapt over time to provide protection against previously unseen dangers. It is this learning and adaptability that AIS algorithms seek to exploit. A class of immune cell, called a lymphocyte, forms the basis of the adaptive immune system. Of particular interest is a lymphocyte called the B-cell and it is from the behaviours of this type of cell that most of the AISIID algorithm is founded. Antibodies are expressed on the surface of B-cells and it is their task to mark invaders (called pathogens) for destruction by other lymphocytes. They do this by binding, chemically, to the surface of the pathogen. As the match between pathogen and antibody does not need to be exact and every B-cell expresses antibodies with slightly different shapes an enormous array of pathogens can be recognised by the immune system. The strength of binding between antibody and pathogen is referred to as the affinity between the two. Thus, in the artificial domain, we can describe the shape of an antibody, describe the shape of a pathogen and define a mathematical function to determine the strength of match between the two.

When a previously unknown pathogen is encountered, the population of cells must adapt to maximise their affinity with it and thus provide the most efficient response. If antibodies on the surface of a B-cell have an affinity with a pathogen, they will clone with a rate proportional to the affinity. Thus the cells that provide the closest match clone faster in order to overwhelm the pathogen. In addition to this, mutation occurs with a rate inversely proportional to the affinity, thus there is strong selective pressure over successive generations of cells that causes affinity to be maximised with this pathogen.

Apart from B-cells, another type of lymphocyte exists, this is called a T-cell. When T-cells are created their T-cell receptor, which may bind in the same way as an antibody, may be in a configuration that will allow it to bind to proteins expressed by the host. A natural safeguard has developed to prevent the immune system attacking the host. A T-cell requires two signals to become activated. Signal 1 is the binding with high affinity between the T-cell and an antigen (a molecular pattern, not necessarily a pathogen). The second signal is given only if the antigen is presented properly, that is, presented by a special type of cell called an antigen presenting cell. Any T-cell receiving signal 1 in the absence of signal 2 (often called a costimulation signal) will be purged from the system, thus ensuring no autoreactive cells are left to harm the host.

1.3. Interestingness and the Web Compare System

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 [4, 14, 21, 22, 23, 24, 28, 30]. 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 unstructured and tends to contain much more noise and irrelevant information than the structured dataset mined by conventional algorithms. 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.

One exception to this is the paper [25], which 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 here. 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 here 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*." and thus can be seen to mirror our own definition. The use of the term "*relevant*" in the above

definition is important as not every piece of unknown information is interesting.

The authors of [25] implemented a number of metrics to assess the interestingness of a web page, the combination of which was called the WebCompare system. Although the metrics described are objective measures, the opinion of a user is always required to validate the assumptions made in constructing them. 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.

In the implementation, after pre-processing, documents are represented as points in vector-space using TFIDF (Term Frequency, Inverse Document Frequency) to weight features, or "concepts" where a concept is a set of keywords that occur together in a page above a certain user-specified minimum support threshold. The TFIDF weighting method is a common way of assigning weights to features (usually words) in a document. 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, or by the maximum frequency of any word found in the document. 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. To compute the term weight of term t in document d in terms of TFIDF, Equation 1 can be used.

$$\mathtt{weight}_{t,d} = TF_{d,t} \times IDF_{D,t} \quad (1)$$

The following metrics were implemented in WebCompare:

1. Finding a corresponding competitor page of a user's page.
2. Finding unexpected terms in a competitor page with respect to a user's page.
3. Finding unexpected pages in competitor with respect to a user's page.
4. Finding unexpected concepts in a competitor page with respect to a user's page
5. Finding unexpected outgoing links on a competitor site.

The technical implementation of (3) is described in section 3.1.1. 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 and this was the strategy employed by the authors. WebCompare was received positively by the three users asked for their opinion. The 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.

1.4. WordNet

WordNet [13] is an electronic repository of words, phrases and relationships between them. In its current version – version 2.0 (available online [35]), 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 each word (or phrase) belongs in a set of synonyms, called a *synset.* A synset is a collection of words that could be interchanged in a context and as such allow the expansion of a concept. There are a number of relationships defined between synsets. Of particular relevance to this investigation are the following relationships:

1. **Generalisation**: a *hypernym* relationship, *Y* is a *hypernym* of *X* if every *X* is a (kind of) *Y*. Not all hypernyms share the same root in the hierarchy.
2. **Specialisation**: a *hyponym* relationship. *Y* is a hyponym of *X* if every *Y* is a (kind of) *X*.
3. **Opposites**: an *antonym* relationship.

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 [33], in which WordNet is used to improve the performance of an information retrieval

system and the paper [18], 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.

2. The AISIID System

AISIID is concerned with a different problem to that solved by traditional search engines such as Google [16] and Yahoo [37]. 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 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 interestingness (novelty, surprisingness, unexpectedness) may be low. However, users receive the results quickly and the results retrieved have been extracted from a 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, which may contain thousands of words, rather than a few keywords. To take advantage of this increased information the resulting system processes pages in much more detail, but the time taken is increased greatly in comparison with conventional search systems. Specifying search criteria in this way also mitigates the problem that users cannot specify unexpected search terms to be overcome, as enough information is present to allow the automated estimation of unexpected concepts.

AISIID is not, therefore, 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 to user's feedback 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. As such AISIID represents a departure from the departure from the conventional web search paradigms

2.1. Overview of AISIID

AISIID uses a population of artificial immune cells and processes inspired by clonal selection to both search for and rank web pages. Both these actions are intertwined and almost inseparable. Many aspects of AISIID are uncommon among web mining and/or AIS algorithms, 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 [13].
- The use of an automated "user feedback" mechanism (akin to the generation of a co-stimulation signal Section 1.2.1).
- 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.

The AISIID process is summarised as follows. The user first specifies a small collection of web pages that summarise his or her knowledge on the search subject. A set of artificial immune cells is generated. Starting on one of the user specified pages, each artificial 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 (something to recognise) and is therefore available for an affinity evaluation. 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 combined 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. Interesting webpages have high affinity with the artificial 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

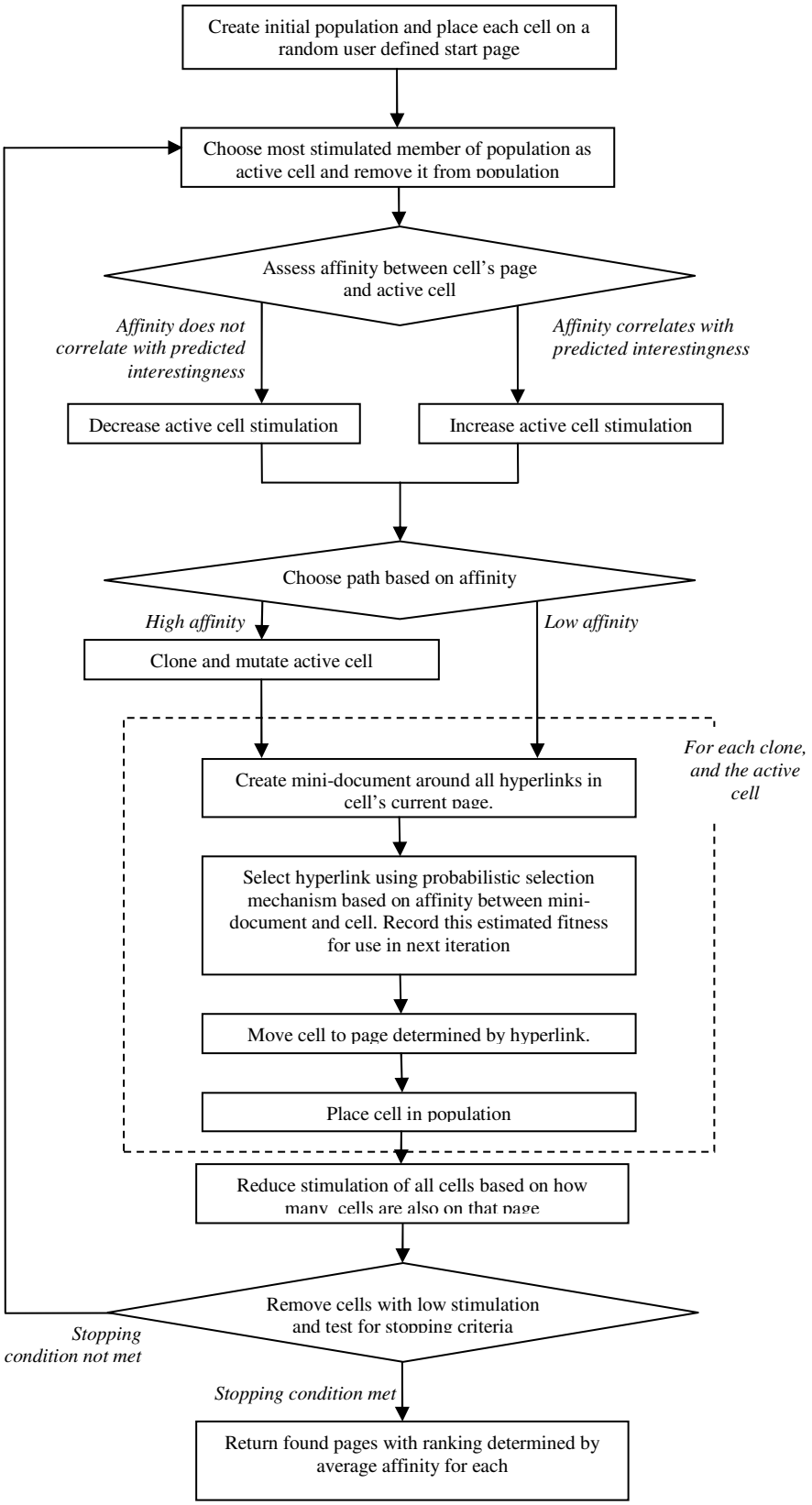

Fig. 1. AISIID system flowchart

found it during the run of the algorithm. A diagrammatic depiction of the algorithm flow is shown in Fig. 1

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, upon termination the user may be presented with a set of the most interesting words regarding the search, further contributing to that user's knowledge and understanding and allowing the user to query why certain pages have been suggested.

2.1.1.1. Two Signal Approach

Work in [27] 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 employed in that reference 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 [6, 7].

AISIID uses a confirmation signal mechanism, in which the confirmation signal is given automatically. When a artificial 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 2.2.5.3) then the artificial cell has made a correct decision, signal two (a confirmation signal) occurs and the cell will be rewarded. If the estimate and actual value differ greatly then the cell must be penalised.

2.1.1.2. Robust Hyperlink Following

AISIID attempts to minimise wasted computational time and network bandwidth by only retrieving information on the user-specified topic. Whilst artificial immune 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 less interesting pages before it reaches interesting ones again. [36] states "*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 system. 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.

This promotes robustness in the spider, one of the advantages of using an AIS type algorithm. One characteristic of the web is that content is often separated from navigation. Pages that contain large amounts of content 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 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.

2.2. Algorithm Description

This section follows the layered AIS framework of [10] i.e. representation, affinity measure and algorithms and processes.

2.2.1. Notation

Pseudocode is presented in this section in which `BC` will refer to an initially empty set of naïve artificial 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` (Relevant Words Vector). E.g. `<spaghetti, chips>.`
- bc_{ITV} is the set of transformations related to `bc` (Interesting Transformation Vector). 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 an artificial cell (temporary copies, new clones, etc). In addition, the following parameters are used, with their legal ranges shown in Table 1.

Table 1.
Parameters and legal ranges for AISIID

Parameter	Legal Range
K_{stim}	> 0
K_{clo}	0 - 1
K_{mut}	0 - 1
K_{size}	> 0
K_{top}	> 0
K_{radius}	> 1
$K_{supress}$	> 0
$K_{proxsup}$	0 - 1

- 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_{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).

The following input data is used:

- Let K_{train} refer to the set of interesting pages selected by the user

2.2.2. 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

Each of these attributes match up with the notation described in Section 2.2.1. In (1) the user's knowledge is summarised, the cell must store this to be able to determine the *relevance* of any artificial antigens (webpages). This vector carries a set of words relevant to the user's search and is therefore referred to as the *Relevant Words Vector* (RWV). As this is an attribute of a cell, it will be referred to as bc_{RWV}. It is assumed a user's prior knowledge and interest will not change during a single run of the system, and as such this summary is fixed (at K_{top}) for the duration of the algorithm's run. The summary of the user's knowledge comprises of a vector of words, stored as Strings. The RWV is not variable in size and will carry the K_{top} most important words as ranked out of all words found in the training documents. The mechanism by which words are ranked is described later.

The cell attribute (2) does not itself 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. 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 one WordNet operation that may legally be applied to the corresponding element of the RWV. These transformations form the adaptable part of the artificial immune cell and so, in contrast to the RWV, will change. The ITV is again an attribute of the cell and so is referred to as bc_{ITV}. For simplicity, this vector is implemented as a vector of integers in the range [0,3] each identifying one of the four unique WordNet operations

1. Antonym
2. Synonym
3. Hyponym
4. Hypernym

The AIS will adapt the elements of the ITV to find the most interesting pages for the user, this is guided by the affinity function to be described later. Generating words using the ITV, RWV and WordNet is one of the most important novelties of this work and so the process is expanded upon in some detail in the following section.

Concerning cell attribute (3), AISIID's cells occupy a position on the web, this current position is simply stored as a URL (bc_{pos}). Finally each cell carries a real number representing a level of stimulation for that cell (4) (bc_{stim}). Cells with low stimulation are removed from the population (detailed later).

2.2.3. Affinity Function

The affinity of a cell with a webpage is calculated using a combination of the words found in the cell's RWV and words generated by the cell's ITV, this set of words can be called an Interesting Words Vector (IWV). The affinity calculation begins by generating the set of interesting words from the cell's RWV using the process in the previous section (in practice these words may be cached to increase efficiency). The webpage is processed into the form of an antigen as described in section 2.2.4.1 to give a set of words. To calculate the affinity, the mean of these two scores is taken (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 relevance of a page is computed as shown in Equation 2 where the number of words in the RWV that also appear on the webpage are counted and then normalised by the length of the RWV.

$$\text{relevance} = \frac{\sum_{i=1}^{|RWV|} \delta_i}{|RWV|} \text{ where } \delta_i = \begin{cases} 1 \text{ if } RWV_i \in W \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

Where RWV_i is the i-th component (word) of vector RWV and *W* is the set of words in the webpage. Likewise, the calculation for interestingness is shown in Equation 3, the IWV is compared against the webpage and the count of the number of words present in both the webpage and the IWV is normalised by the length of the IWV. It should be noted that the ITV is not used directly, rather the words in the relevant word vector are combined with the transformations defined in the ITV to create a set of interesting words (IWV) and it is this latter vector of words which is compared with the webpage. The process by which this is done is expanded in the next subsection. Thus the vector's "phenotype" is used in this instance.

$$\text{interest} = \frac{\sum_{i=1}^{|IWV|} \delta_i}{|IWV|} \text{ where } \delta_i = \begin{cases} 1 \text{ if } IWV_i \in W \\ 0 \text{ otherwise} \end{cases} \quad (3)$$

Finally, the affinity between cell and webpage (*W*) is calculated as shown in Equation 4. 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 and as such is left for future research.

$$\text{affinity} = \frac{1}{2} \times (\text{relevance} + \text{interest}) \quad (4)$$

Given a current cell, `bc`, and a webpage processed into the form of an antigen, `ag`, the affinity between `bc` and `ag` is computed as shown in Pseudocode 1. $\texttt{count}_{x,y}$ 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 $\texttt{count}_{\texttt{INT,ag}}$ 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..1].

2.2.3.1. Using WordNet Relationships to Generate Interesting Words

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 Section 1.3 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 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 applying the *antonym* relationship to the RWV as it will, by definition, contain expected 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, 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 is performed.

```
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 1. Affinity between immune cell and antigen (web page)

The Java WordNet Library (JWNL) is used to create an interface between AISIID and WordNet. JWNL is freely available from the Sourceforge website [11] 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. This must be done for all parts of speech in which the word legally exists, and this check is easily performed in WordNet. All operations of this nature will return a set of words (synset) as WordNet maps synsets to synsets, not words to other words.

For the synonym and antonym relationship, a single synset maps to another single synset. For the hierarchical operations: hypernym and hyponym, the hierarchy is followed with a synset returned from each level of the hypo/hypernym hierarchy within a given number of levels of *w*.

WordNet sysnets do not contain only 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 when comparing IWV elements to a webpage as it was observed that a significant proportion of the words generated by WordNet are compounds and as such they cannot be ignored and must be treated appropriately.

.

```
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 2. AISIID main algorithm

2.2.4. Processes

The following sections describe the main processes that, when combined, make up the AISIID algorithm. The main algorithm consists of 8 stages within a loop. Each of these stages is detailed in Pseudocode 2, 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 control. 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

2.2.4.1. Initialisation

The purpose of initialisation is to create an initial set of artificial 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 whether 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 as features pertaining to a *common topic* will be reinforced (as these appear throughout the pages) as the content of this noise is likely to differ between all initialisation pages. The more diverse (yet still "on topic") the set, the better the potential for good results.

Each webpage will be processed into the form of an antigen 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 (a common preprocessing step used to reduce the number of irrelevant attributes [5]) is performed over the set of tokens.

2.2.4.2. 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 carefully.

Naively it would be possible to rank the words by word frequency but some words will naturally occur more than others and so while this is a possible solution, it is certainly not optimal. The common solution TFIDF ranking, 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 "interesting" class but also those of a "general" class. Thus, it is impossible to gauge the real quality of a feature from the interesting class when it is only compared with the all features of that interesting class. Given that during the initialisation stages the algorithm is only aware of the interesting pages supplied by the user, then assessing TFIDF weightings over a set of documents where all documents are drawn from this single class is possible, but meaningless, as there is no general class to compare against. Due to this limitation it was decided that Google could be used to emulate the use of a set of general documents.

The process of weighting words found in the initialisation documents proceeds as follows. The initialisation documents are first concatenated to form one single document. From this single document the term

frequency of each feature is calculated in the usual manner.

The inverse document frequency of a term is calculated using a search submitted to the search engine Google. 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 reasonable estimate of the true document frequency on the whole web (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 [15]. The results of any automated search will return the number of web pages containing the search term, which in turn gives an estimate of document frequency of that keyword over an approximation of 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. At the time of running the experiments (January 2006), this was 8,058,044,651.

The TFIDF weighting (w) of a word (i) in the initialisation document is finally computed as shown in Equation 5.

$$w_i = \frac{f_i}{\max_{f_i}} \times \log_2 \frac{N}{n_i} \tag{5}$$

This is a re-implementation of Equation 1, 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.

Once the TFIDF values for the words in the starting pages have been computed, the words are ranked by their TFIDF value and the highest weighted K_{top} are selected to form the cell's RWV. An initial set of artificial immune cells is then created using the same 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.

```
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
12      SCORE ← SCORE ∪ {w,wscore}
13    W_top = Determine top K_top 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 3. Initialisation of ASISIID

The procedure in Pseudocode 3 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.

2.2.5. Running

After initialisation, the main loop of the algorithm consists of a cell following a hyperlink to a webpage, assessing the quality of that page using the affinity function and reacting accordingly. This may include cloning/mutating depending on the affinity between the cell and the page. As AISIID is single-threaded, an order must be established with which to process the members of the population. 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 performing at finding interesting web pages. During each iteration the most stimulated cell, that at the head of the queue, should be tested first, and so it is removed from the queue to be tested. This is referred to as the "active cell", and the procedures described in sections 0 to 0 are applied.

2.2.5.1. Cell Movement and Choice

Each artificial immune cell must select the page it is to move to next, this allows the search space to be explored and to aid search diversity this is done in a probabilistic manner. A hyperlink is chosen by running a roulette wheel selection procedure [14] over all hyperlinks in the page on which the cell currently resides. The mechanism by which hyperlink weights are generated (to bias the roulette wheel) is as follows. Each hyperlink is weighted using a measure of the relevance of the text surrounding it. This text is considered in isolation from the rest of the text and so each hyperlink is associated with its own "mini document". 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. Fig. 2 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". 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.

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.

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

Fig. 2 Generation of mini-document from webpage

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:

$$\text{weight}_{\text{md,w}} = \frac{\sum_{i=1}^{|RWVw|} \delta_i}{|RWV_w|} \text{ where } \delta_i = \begin{cases} 1 \text{ if } RWVw, i \in md \\ 0 \text{ otherwise} \end{cases} \quad (6)$$

In Equation 6 the weight of mini document *md* and a webpage *w* is calculated using a count of all words occurring in the Relevant Word Vector (*RWV*) of *md* and *w*, which is normalised by the size of the *RWV* held by webpage *w*. Pseudocode 4 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 5th word in the ordered list.

While this method of scoring the hyperlink is relatively simplistic, it is necessary, as more complex metrics based on weighting words using TFIDF are not available. 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. 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.

```
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 4. Create Mini-document procedure

If the file type is of no use, or destination of the URL is not found then the cell will first backtrack to the page from where it came. 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.

2.2.5.2. 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 and therefore assess the interestingness of the page. The current page is processed as described in section 2.2.4.1 and WordNet based transformations are applied to each word in the RWV to create a IWV as described in 2.2.3.1 This set of words is used in the affinity calculation between the page as described previously (2.2.3).

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 0).

2.2.5.3. Automated Feedback

The co-stimulation model is used to stimulate or suppress cells based on their quality. It is reiterated that it is not practical for a user to 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 and as such they may not move to the "best" page out of a set of potential pages. Indeed, it is quite possible for a cell to move to a webpage irrelevant to the current search. The absolute affinity score cannot be used to confirm the quality of the cell, as the affinity with the page could be low (as the cell has moved to an irrelevant page) while the cell is otherwise useful. Cell stimulation is therefore varied based on the *difference* between the estimated and actual affinity with a web page. A given cell may predict a page will have a low interestingness. 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 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.

$$\text{stimulation}_{t+1} = \text{stimulation}_{t} - \text{abs}\left(10 \times \left(aff_{est} - aff_{c,w}\right)\right) \quad (7)$$

Equation 7, 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.

```
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. Procedure for cloning and mutating a cell

2.2.5.4. Cloning and Mutation

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. To maximise the search around this interesting page, the cell 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, as is typical in clonal selection algorithms. The process is straightforward and is described in Pseudocode 5.

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). After mutation, the location of each clone is temporarily set as the same page as its parent. The clone then moves one hyperlink away from that page using the hyperlink selection mechanism detailed previously. Once it has moved, the cell is initialised with a default stimulation level and is placed in the population queue.

Pseudocode 5 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, while number of positions of the ITV vector to be mutated in each clone is inversely proportional to the affinity of the cell. The symbol $\lfloor x \rfloor$ denote the floor of x, that is x rounded down to the nearest integer.

2.2.6. Population Control

The final part of the main loop in the algorithm concerns the removal of cells from the population. It is important to guard against redundant cells (those in an 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 is checked. This may have been reduced wither by the mechanism above or because it made a bad estimate of page interestingness (section 2.2.5.3) If it is found to be below a threshold the cell is removed from the population, otherwise it remains.

2.2.7. Returning Results

When a stopping criterion is met, the user is presented with a ranked 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 gives the user an idea regarding *why* particular pages 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 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 [14]. Indeed, in [25], the authors 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.

3. Analysis of Performance

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 WebCompare was developed: "*There is also no existing system that is able to perform our task. Thus, we could not do a comparison*" [25].

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 output of AISIID is, by its very nature, subjective. The most revealing way to test the output directly therefore is with a user study. This situation was 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*". 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. It should be noted that the aim of 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.

3.1.1. Generating Comparison Pages with WebCompare

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* [competitor website] *with respect to U* [user website]". 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 8 as follows. The unexpectedness score of each word is first computed using Equation 10, 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 10. Normalisation is a reasonable step, as the *C* and *U* documents may be of greatly different sizes, rendering calculations based on absolute frequency subject to inaccuracy. The sum of unexpectedness scores for every term appearing on a page is then computed and normalized by the number of words in the document to give the final score, as shown in Equation 10.

$$\mathtt{unexpT}_{r,i,t} = \begin{cases} 1 - \dfrac{tf_{r,j}}{tf_{r,i}} & \mathtt{if} \quad \dfrac{tf_{r,j}}{tf_{r,i}} \le 1 \\ 0 \ \mathtt{Otherwise} \end{cases} \tag{8}$$

$$\mathtt{unexpP}_i = \frac{\sum_{r=1}^{m} \mathtt{unexpT}_{r,c,u}}{m} \tag{9}$$

$$tf_{i,j} = \frac{f_{i,j}}{\max f_{i,j}} \tag{10}$$

This particular metric was re-implemented 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. 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 WebCompare 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 ($\mathtt{K}_{\mathtt{train}}$). 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.

3.1.2. Experimental Protocol

In order to compare AISIID and WebCompare, 15 users were asked to take part in a user study. Users were all familiar with the internet and search engines, and it is likely that each user would have searched for his or her chosen subject many times before. Each user was asked to supply a small number of URLs (typically 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.

AISIID was run 3 times, each run retrieving 2500 pages. Using multiple runs is standard practise when using non-deterministic algorithms such as this and preliminary tests showed that 2500 pages gave a reasonable trade off between depth of search and number of times the algorithm is run. In practice this meant that the 3 runs could be completed overnight. 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 over all runs. The mean affinity of each page was used to rank the list of URLs found by AISIID. All URLs encountered over the 3 runs were used as inputs to WebCompare as the competitor set. It should be made clear that both AISIID and WebCompare have the opportunity to score exactly the same pages, thus any variation in the opinion of users to the rankings created by each system must be due to the differences in the systems rather than any difference in the pages. The page unexpectedness score generated by WebCompare was used to rank the pages into 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.

The 3 sets of results from AISIID were merged into a single list, and the 3 sets of results from WebCompare were merged into another list. The top 20 URLs as scored by AISIID, the top 20 URLs as scored by WebCompare are the URLs to be seen by the user. Any duplicates within each list were removed (as the same page could be visited on more than one of the runs). Any user defined starting pages that were found in the list were also removed (it is possible for an immune cell to find a way through a chain of hyperlinks and find itself back at these pages).

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. It should be stressed that this was a blind test and while users could see a list of URLs, they were not informed how the URLs were retrieved.

3.1.3. Statistical Test of Significance

Student's t-test [2, 8] is a statistical test of significance frequently used in and data mining and machine learning texts [26, 31]. 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 [8].

The t-test was designed for a situation where a small number, typically less than 30, observations have been made (as the sampling technique in the Guinness brewery involved taking bottles off the production line, Guinness obviously wanted to perform tests on their product with the smallest sample size possible). For larger sample sizes the use of the normal distribution is preferable. Unlike the normal distribution, the number of degrees of freedom must be calculated based on the number of samples taken. The degrees of freedom shape the t-distribution based on the sample size used to determine the standard deviation. Informally, this allows the final probability value corresponding to a t-value to change based on the sample size thus making it suitable for small samples.

There exist two types of t-test: the independent and the paired tests. The independent t-test places no restriction on the observations and as such is the one used in the following section. In addition to this, 2-tailed tests are used throughout this section. The combination of an independent test and a 2-tails test results in the figures given being the most pessimistic.

3.1.4. User Test Results

A total of 15 users agreed to participate in the test. We emphasise that this number of users was considerably larger then the number of users recruited to test WebCompare in [25] where only three users evaluated the system. We also emphasise that in this test users will give numerical ratings which will allow a proper statistical comparison, rather then general opinions.

Table 2 summarises the user's individual search topics.

Table 3 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.

Table 2.
Summary of test users and their subjects

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 3.
Mean subjective interestingness scores for AISIID user tests

	AISIID		WebCompare	
User ID	Mean	Std. Dev.	Mean	Std. Dev.
User 1	2.30	2.64	0.90	1.37
User 2	2.50	3.03	0.00	0.00
User 3	0.42	0.51	0.11	0.45
User 4	1.70	2.52	0.00	0.00
User 5	8.20	1.79	2.50	3.30
User 6	0.25	0.55	0.10	0.45
User 7	4.95	3.52	1.10	0.31
User 8	2.00	2.22	0.11	0.31
User 9	3.55	3.61	0.30	0.57
User 10	1.55	2.39	0.00	0.00
User 11	3.70	3.97	1.50	1.15
User 12	2.65	3.36	0.00	0.00
User 13	6.95	3.69	1.47	1.81
User 14	2.95	3.40	0.00	0.00
User 15	4.90	2.27	1.30	1.53
Mean	3.24	2.23	0.63	0.79

It can be seen from that over the 15 tests, AISIID has scored, on average, considerably higher than WebCompare. AISIID was found to have a mean score of 3.24 over all the tests, whilst WebCompare achieved a surprisingly low mean score of just 0.63. The 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 but it can be seen that AISIID scored a maximum of 8.20 while WebCompare's maximum score was just 2.50.

It is important to determine whether the ratings given to the pages retrieved by AISIID are statistically better than those retrieved by WebCompare. Student's t-test [2, 8] 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 [8]) and as such is particularly suited to the results of these experiments where 15 observations have been made. The threshold for significance is $P_{null}<0.05$ while values of $P_{null}<0.01$ are thought to be highly significant. The null hypothesis is that the means of the observed scores given to AISIID and WebCompare do not differ. The probability of the null hypothesis holding was found to be $P_{null} = 0.0002$, much less than the threshold for significance.

3.1.4.1. Assessment of WebCompare

Some thought must be given to the surprisingly low score of WebCompare. It is thought that WebCompare does not fare well compared with AISIID for a combination of reasons which are generally the result of empirical observations. Firstly, the WebCompare metrics have no direct measure of relevance. The WebCompare system makes the strong assumption that every page on a competitor's website will be relevant to the user's search. 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 then 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, possibly 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 this 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 it is thought that because of this shorter pages are 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 rarely 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. 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.

One advantage WebCompare does have over AISIID is its speed. Tests showed that AISIID will use approximately 30 minutes of CPU, although this is variable depending on the length of the retrieved pages. This is compared to approximately 20 seconds CPU time for WebCompare. However, this comparison is not straight forward. WebCompare takes the processed output from AISIID as its input and so AISIID is doing some pre-processing work for WebCompare. In tests, AISIID would take between 4 and 5 hours to complete, values which are highly dependent on the speed of the network. The difference between this and the CPU time reported being the time taken waiting for the network. Thus, as both algorithms require a set of pages to be retrieved, in reality the network bottleneck will ensure that both complete in comparable time.

3.1.5. User Test Observations

During testing, the users were encouraged to give their opinions on both the process of identifying the initial set of starting pages, and comment on the quality and characteristics of the results.

Recall the selection of the initial starting pages is important as they will ensure the system correctly infers what the user does and does not know. It is vital, therefore that the user is able to specify pages rich in their prior knowledge. Users frequently commented that it was hard for them to specify pages that will adequately summarise what they know on a subject, whilst containing very little extraneous information. Generally a page will tend to contain some pieces of information the user does not yet 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 as their 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. These conspire to produce a substandard result, 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.

A few tests were found to result in very low absolute scores but this leads to a useful observation regarding a possible shortcoming of WordNet and therefore its use to finding interesting pages. The tests for user 3 (areas of shapes), user 4 (mathematical graph drawing) and user 6 (Markov chains) produced unexpectedly poor 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 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.

Secondly, pages about some topics, especially in the case of Markov chains, were relatively uncommon; indeed the user admitted that even finding good pages through standard search engines was a challenge.

3.1.6. Interrogating the Results

Allowing users to interrogate the output is vital to support the decision making process [12, 14]. 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. 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 that are published do appear reasonable in the context of the searches.

The most interesting words for each user 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 of these URLs the cell that had the highest interestingness score with that URL is identified. All words from that cell's interesting word vector (IWV) are then added to a list. The frequency with which each word occurs in the list is computed and allow these words to be ranked in order of frequency, and thus assumed importance. As the IWV of each cell is variable

in length, 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. 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 these 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. 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 are 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.

Some of the words included in the list shown to the user were not useful. It is thought that many of these were caused by WordNet using the wrong part of speech when generating 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 those detailed in [3] or [32], fewer words will be irrelevant and the accuracy of AISIID has the potential to be increased.

4. Future Work: Improving AISIID

During implementation and testing a number of technical improvements that may increase the quality of AISIID's output were identified. 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 other file formats such as Adobe's Portable Document Format (PDF) or Postscript would allow users to find a wider variety of results.

Attribute weighting is a problem encountered throughout the data mining literature, especially when considering instance based learning [1, 20]. 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 simple 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.

Such an investigation to determine the relative importance of these two attribute weighting strategies would be a significant study, but one with the potential to greatly improve the quality of the result.

Other improvements include making more use of the metadata contained in the HTML, reversing cell's movements if it begins to make its way into an unproductive area of the web or use phrases or concepts (combinations of adjacent words) to improve the accuracy of information retrieved as they may contain more information than words used in isolation. Webpage pre-processing could be improved to separate relevant from irrelevant webpage content (where irrelevant content would typically include advertisements or banners) and part of speech recognition algorithms could be employed to reduce the number of irrelevant words being generated by WordNet.

As noted in Section 1.1.1, 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 as certain procedures could be parallalised, 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.

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 continuous learning system may be exploited and AISIID could be taken to another level.

5. Conclusion

The discovery of interesting information on the web poses some unique challenges. This paper has evaluated an artificial immune system, called AISIID, that was developed to meet these. 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. 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. A number of novel solutions to the challenging nature of searching for interesting documents on the web were incorporated in AISIID. Of note 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. 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.

In the introduction, a number of properties of artificial immune systems were identified that mirror desirable properties of web mining systems. The property of pattern recognition is used during every iteration of the algorithm with cells matching their internal patterns of interesting and relevant words with the content of web pages and reacting accordingly. The pattern recognition is, however, non-specific and this leads to implicit noise tolerance. Pages are returned that may not be perfect but if enough cells match the page, albeit imperfectly, the page will be returned to the user.

Self organisation can also be seen in the way cells tend to crowd around the most "interesting" areas of the web (as known to the algorithm). It would be wasteful (in terms of both network bandwidth and processor cycles) for cells to perform exploration far away from areas of high interestingness, instead many cells are seen to perform local searches around known areas of high interestingness (where the chances of finding even more interesting information is high) while a few will make their way many hyperlinks away exploring for new areas of interestingness. This highly desirable self organising behaviour is an emergent property of the system and can be seen in the visualisation in Fig. 3. In this figure, each immune cell is a red node with the size of the node growing in proportion with the number of cells on that page. Each edge shown represents a hyperlink a cell has followed. Near the bottom of the figure a cluster of pages is easily visible representing an area of high interestingness, this is where the vast majority of the cells reside in this example. The circled area shows a trail in which one cell has explored the area around this cluster of cells and has happened upon another area of high interestingness to the top-right of the figure, visible as a growing cluster of cells.

These processes are all underpinned by a diverse set of cells. As each cell carries its own image of what is interesting, a broad, diverse, range of ideas is reflected in the results as returned to the user. Enhanced diversity, in which cells may recognise a diverse range of media types as presented on the internet, is a possible improvement left for the future.

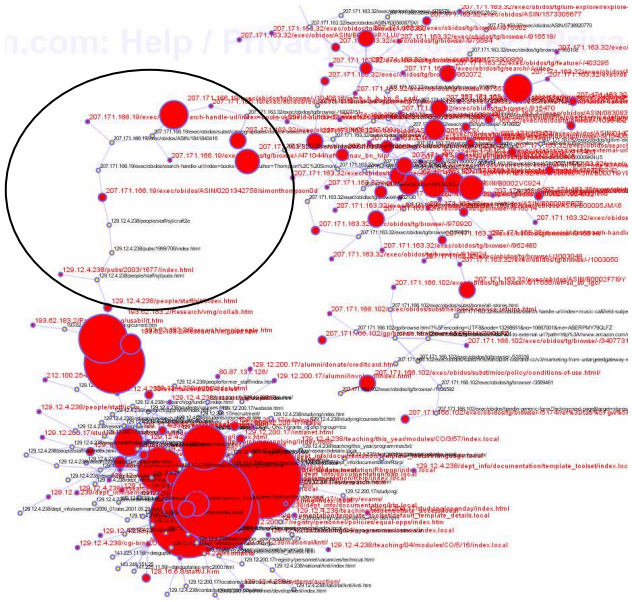

Fig. 3 Visualisation of AISIID spidering

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 It was found, however, that WebCompare was quicker in these tests than AISIID, but in reality when the retrieval of web pages is taken into account both are likely to complete in comparable times.. This investigation was enlightening as (a) WebCompare had only been tested by 3 users previously, whereas in this investigation 15 users were asked to judge the output, (b) in this work the user's evaluation was delivered as a numerical score rather than simply a user's opinion allowing more meaningful interpretation and (c) WebCompare was compared with another algorithm (ASISIID), a procedure not available to the authors of the original WebCompare system. Some faults with WebCompare were identified and possible solutions presented. It is thought that this paper therefore makes a contribution to the literature regarding WebCompare as such a study had not been attempted previously.

The 15 users were asked to look over the interesting words generated by AISIID and WordNet, all users asked found this process useful and helped them understand why particular pages had been recommended. Limitations were found in the ability for WordNet to transform words; proper nouns and technical language were found as two examples of this.

In the future it is expected that users will demand more from their information retrieval or classification tools. 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 would already have the knowledge it needs to decide if something is interesting for a user without that user having to supply any additional information. This knowledge would be 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. In addition such a system would be of great use and would really exploit the lifelong learning characteristics [17] that it is believed AIS might possess.

References

[1] D. W. Aha, D. Kibler and M. K. Albert, Instance-Based Learning Algorithms, Machine Learning 6, (1991) 37-66.

[2] H. L. Alder and E. B. Roessler, Introduction to Probability and Statistics (W. H. Freeman, 1968).

[3] E. Brill, A Simple Rule-Based Part-of-Speech Tagger, in: Proc. 3rd Conference on Applied Natural Language Processing (ANLP-92), (ACM Press, 1992) 152-155.

[4] D. R. Carvalho, A. A. Frietas and N. F. F. Ebecken, A Critical Review of Rule Surprisingness Measures, in: Proc. Data Mining IV - International Conference on Data Mining, (WIT Press, 2003) 545-556.

[5] S. Chakrabarti, Mining the web (Discovering Knowledge from Hypertext Data) (Morgan Kaufmann, San Francisco, 2003).

[6] P. K. Chan, A Non-Invasive Learning Approach to Building Web User Profiles, in: Proc. Workshop on Web Usage Analysis and User Profiling, Fifth International Conference on Knowledge Discovery and Data Mining, (1999) 7-12.

[7] C. C. Chen, M. C. Chen and Y. Sun, PVA: A Self-Adaptive Personal View Agent System, in: Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (ACM Press, 2001) 257-262.

[8] J. H. Creighton, A First Course in Probability Models and Statistical Inference (Springer-Verlag, 1994).

[9] D. Dasgupta, Artificial Immune Systems and Their Applications (Springer-Verlag, 1999).

[10] L. N. de Castro and J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach (Springer-Verlag, 2002).

[11] J. Didion, Java WordNet Library Homepage, http://sourceforge.net/projects/jwordnet, Accessed November 2005.

[12] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, Advances in Knowledge Discovery and Data Mining (MIT Press, 1996).

[13] C. Fellbaum, WordNet: An Electronic Lexical Database (MIT press, Cambridge, 1998).

[14] A. A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms (Springer-Verlag, 2002).

[15] Google, Google: Google Web APIs (beta), http://www.google.com/apis/, Accessed November 2005.

[16] Google, Google: Google Search Homepage, http://www.google.com, Accessed January 2006.

[17] E. Hart and J. Timmis, Application Areas of AIS: The Past, The Present and The Future, in: Proc. 4th International Conference on Artificial Immune Systems (ICARIS 2005), Lecture Notes in Computer Science, (Springer-Verlag, 2005) 483-497.

[18] N. Holden and A. A. Freitas, Web Page Classification With an Ant Colony Algorithm, in: Proc. Parallel Problem Solving from Nature (PPSN 2004), Lecture Notes In Computer Science, Vol. 3242 (Springer-Verlag, 2004) 1092-1102.

[19] F. W. Lancaster, Vocabulary Control for Information Retrieval (Information Resources Press, Washington D. C, 1976).

[20] C. X. Ling and H. Wang, Computing Optimal Attribute Weight Settings for Nearest Neighbour Algorithms, Artificial Intelligence Review 11, (1997) 255-272.

[21] B. Liu and W. Hsu, Post-Analysis of Learned Rules, in: Proc. 13th National Conference on Artificial Intelligence (AAAI '96), (AAAI Press, 1996) 828-834.

[22] B. Liu, W. Hsu and S. Chen, Using General Impressions to Analyze Discovered Classification Rules, in: Proc. 3rd International Conference on Knowledge Discovery and Data Mining (KDD '97), (1997) 31-36.

[23] B. Liu, W. Hsu, L.-F. Mun and H.-Y. Lee, Finding Interesting Patterns Using User Expectations, IEEE Transactions on Knowledge and Data Engineering 11, (1999) 817-832.

[24] B. Liu, M. Hu and W. Hsu, Multi-Level Organisation and Summarisation of the Discovered Rules, in: Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000), (ACM Press, 2000) 208-217.

[25] B. Liu, Y. Ma and P. S. Yu, Discovering Unexpected Information From Your Competitors' Web Sites, in: Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001), (ACM Press, 2001) 144-153.

[26] T. M. Mitchell, Machine Learning (McGraw-Hill, 1997).

[27] A. Secker, A. A. Freitas and J. Timmis, AISEC: an Artificial Immune System for E-mail Classification, in: Proc. Congress on Evolutionary Computation 2003 (CEC2003), (IEEE press, 2003) 131-138.

[28] A. Silberschatz and A. Tuzhilin, What Makes Patterns Interesting in Knowledge Discovery Systems, IEEE Transactions on Knowledge and Data Engineering 8, (1996) 970-974.

[29] L. Sompayrac, How the Immune System Works (Blackwell Science, 1999).

[30] P. N. Tan, V. Kumar and J. Srivastava, Selecting the Right Interestingness Measure for Association Patterns, in: Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (ACM Press, 2002) 32-41.

[31] P. N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining (Addison Wesley, 2005).

[32] D. Tufis and O. Mason, Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger, in: Proc. 1st

International Conference on Language Resources and Evaluation, (1998) 589-596.

[33] E. M. Voorhees, Using WordNet for Text Retrieval, in: C. Fellbaum, ed., WordNet: An Electronic Lexical Database, (MIT Press, Cambridge, 1998) 285-303.

[34] A. Watkins and J. Timmis, Exploiting Parallelism Inherent in AIRS, in: Proc. 3rd International Conference on Artificial Immune Systems (ICARIS 2004), Lecture Notes in Computer Science, (Springer-Verlag, 2004) 427-438.

[35] WordNet, WordNet: a lexical database for the English language, http://wordnet.princeton.edu/, Accessed September 2004.

[36] F. Wu and C. Hsu, Using Context Information to Build a Topic Specific Crawling System, in: A.Scime, ed., Web Mining: Applications and Techniques, (Idea, London, 2005) 50-68.

[37] Yahoo! Yahoo! Search Homepage, http://search.yahoo.com, Accessed January 2006.