Towards A Danger Theory Inspired Artificial Immune System for Web Mining

Andrew Secker
ads3@kent.ac.uk
Computing Laboratory, University of Kent, Canterbury, Kent, UK, CT2 7NF
Telephone: +44 (0)1227 764000 ext. 4278, Facsimile: +44 (0)1227 762811

Alex A. Freitas
aaf@kent.ac.uk
Computing Laboratory, University of Kent, Canterbury, Kent, UK, CT2 7NF
Telephone: +44 (0)1227 827220, Facsimile: +44 (0)1227 762811

Jon Timmis
jt6@kent.ac.uk
Computing Laboratory, University of Kent, Canterbury, Kent, UK, CT2 7NF
Telephone: +44 (0)1227 823636, Facsimile: +44 (0)1227 762811

Towards a Danger Theory Inspired Artificial Immune System for Web Mining

ABSTRACT

The natural immune system exhibits many properties that are of interest to the 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. As part of a larger project to construct a large-scale dynamic web-mining system, this chapter reports initial work on constructing an E-mail classifier system. The Artificial Immune System for E-mail Classification (AISEC) is described in detail and compared with a traditional approach of naive Bayesian classification. Results reported compare favorably with the Bayesian approach and this chapter highlights how the Danger Theory from immunology can be used to further improve the performance of such an artificial immune system.

Keywords: Artificial Immune System, Classification, Data Mining, Machine Learning

INTRODUCTION

Web-mining is an umbrella term used to describe three quite different types of data mining, namely content mining, usage mining and structure mining (Chakrabarti, 2003). Of these, we are concerned with web content mining which Linoff & Perry (2001) define as "*the process of extracting useful information from the text, images and other forms of content that make up the pages*" (p. 22). This work is concerned with performing text mining on the web for the purposes of classification, but this is a hard task to achieve well. Firstly, the data contained on these pages is extremely low grade, noisy and inconsistent in format and secondly the problem space may be vast. As of August 2003 the Internet's largest search engine, Google, indexes 3.3×10^9 web pages (Google, 2003). Finally the ease with which pages are published, moved or removed gives rise to an extremely dynamic medium.

It is our ultimate goal to construct a system to mine from the web pages that the user will find interesting. That is, the user may consider them novel, surprising or unexpected. This is a slightly different problem from the classic classification task, as the class assigned to the page will depend not only on its content, but some current context. Work in (Liu, Ma, & Yu, 2001) describes a system to mine surprising pages from competitor's websites. At a high level it is possible that we may take inspiration from this, such as using a piece of user-

specified information to infer the subject on which the user requires information and the interestingness of the retrieved result and thus lead to future work. Statistical techniques such as a naïve Bayesian algorithm (Mitchell, 1997) have proved successful when used for the classic classification task but we propose the use of a system we believe may be more adaptable than a Bayesian algorithm: an Artificial Immune System.

Over the last few years, 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 vast collection of domains (deCastro & Timmis, 2002). There are a number of motivations for using the immune system as inspiration for both data mining and web mining algorithms which include recognition, diversity, memory, self regulation, and learning (Dasgupta, 1999). Being based on an AIS algorithm, by its very nature the system will preserve generalization and forget little used information. Thus giving a system such as this the ability to dynamically calculate interestingness based on context and adapt to changing user preferences. Being an adaptive learning system it will not require expert set-up, instead it will learn, for example, a particular intranet structure and tailor itself to user's tastes.

In itself an AIS based web mining system would be a significant advance in the field of immune inspired algorithms. However it is our ultimate goal to go further than the areas of both web mining and artificial immune systems by taking inspiration from an immunological theory called "Danger theory" (Matzinger, 2002a). We believe that algorithms inspired by this theory are suited to continuous learning tasks on large and dynamically changing data sets. In this theory, an immune response is launched based on a notion of perceived danger based on a current context. Thus inspiring the context dependent measure of interestingness required in the final system. The scalability of immune based systems has been called into question (Kim & Bentley, 2001) and we believe the notion of a localized immune response in the Danger theory may offer some solutions by only activating the immune algorithm within context dependent area, as explained later.

Now our final goal, a web mining tool to retrieve interesting information from the web, is defined, there are a number of steps we must take to reach it. The first is to show that an immune based algorithm can successfully perform a text mining task for the purposes of classification with an accuracy comparable to that of a standard

technique such as a naïve Bayesian classifier. In this chapter we use a well know probabilistic technique, a naïve Bayesian classifier, for the purposes of comparison. This is a necessary step as there are few references in the literature to turning immune inspired techniques to such a task. One exception is (Twycross, 2002) but the significant difference to this is we propose a system for continuous learning, thus making just this first stage a significant advance for the field which we can use as a step towards a Danger theory inspired system.

The algorithm named "AISEC" (Artificial Immune System for E-mail Classification) described in this chapter is a novel immune algorithm specifically designed for text mining and, as such, a first step towards our goal. In the following pages we begin by describing in a little more detail the background to the project including an explanation of the artificial immune system paradigm and the Danger theory. We then describe the implementation and testing of the AISEC system, our first step towards the realization of an immune inspired content mining system. Finally we conclude by discussing how the work in this chapter contributes to our goal, the strengths of an AISEC-like system, possible improvements and our ideas for the future of web mining using immune inspired metaphors.

BACKGROUND

Although research into AIS began in the realm of computer security for virus detection and suchlike, some AIS based algorithms lend themselves particularly well to data mining such as that described in (Hunt & Cooke, 1996). (Watkins & Timmis, 2002) describes the artificial immune system AIRS which was shown to classify test data with an accuracy comparable to many standard algorithms. For a summary of a number of immune inspired algorithms for data mining, the reader is directed to (Timmis & Knight, 2002). One single reference can currently be found in the literature to an immune inspired system for text mining. (Twycross, 2002) details an AIS for classification of HTML documents into two classes: those which were on a given topic or not. The algorithm was tested on pages taken from the Syskill and Webert Web Page Ratings from the UCI data repository (Blake & Merz, 1998). This dataset consists of HTML pages, each on one of four different topics. The task was for this immune inspired system to predict if an unseen page was on a given topic or not when the system was trained using a number of example pages. The system was compared with a naïve Bayesian classifier and achieved a higher predictive accuracy in three out of four domains. The results showed that the system was relatively insensitive to the size of the training set which was in contrast to the Bayesian system with which it was compared.

Artificial Immune Systems

Before we continue, we would like to briefly describe the important parts of an artificial immune system in the context of the natural immune system. Throughout we will only concentrate on the elements of the immune system relevant to the AISEC classification system described here. For a more general review of the immunology behind artificial immune systems the reader is directed towards the literature such as (Sompayrac, 1999).

The mammalian immune system works at three distinct levels, physical barriers (e.g. skin), the innate immune system and the adaptive immune system. AIS are concerned with the latter as only this exhibits the desirable properties for a computational intelligence system such as learning and memory. The natural immune system is based around a set of immune cells called *lymphocytes* and it is the manipulation of populations of these by various processes that give the system its dynamic nature.

From a data mining perspective, an important component of a natural immune system is a *receptor*. These receptors are found on the surface of immune cells of the adaptive immune system called *B-cells* and *T-cells*, collectively known as lymphocytes. Each receptor is unique in shape and capable of binding to a slightly different range of molecular patterns from others. Typically a receptor (Figure 1) will bind to proteins expressed on the surface of an invading cell and any object capable of binding to one of these receptors by chemical interactions is called an *antigen.* A subset of the antigens are those that can harm the host, such as viruses and bacteria, and are referred to as pathogens. Similarly, at the core of an AIS is a set of immune cells, each described by a feature vector (Figure 1). The cell will represent a point in the solution space; a notion biologists refer to as a location in *shape space* (Figure 2C). In the system described in this chapter, the antigens are the objects to be classified and typically use the same representation as the immune cells, i.e. a feature vector.

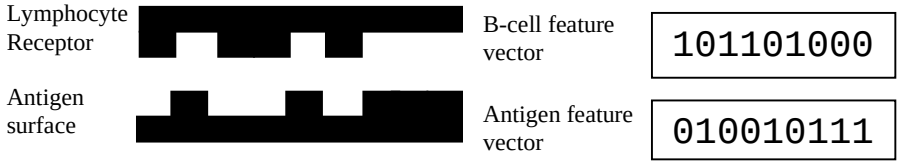

Figure 1. Analogy between B-cell receptor and artificial immune cell feature vector. The feature vector for the artificial cells could, for example, be a Boolean representation representing the presence or absence of words in a document.

An *affinity function* may be defined to determine a measure of similarity between an immune cell and an antigen or between two immune cells. If the value calculated is greater than a threshold the antigen is said to be within the *recognition region* of the immune cell or that the lymphocyte will *recognize* the antigen. This reflects the natural system where *regions of complementarity* are needed to provide enough electromagnetic force between an antibody's receptor and an antigen to pull these two cells together. In Figure 1, for example, the region of complementarity extends over the entire length of the receptor. The match between the receptor and antigen need not be exact and so when a binding takes place it does so with a certain strength called an *affinity*. These terms are described diagrammatically in Figure 2.

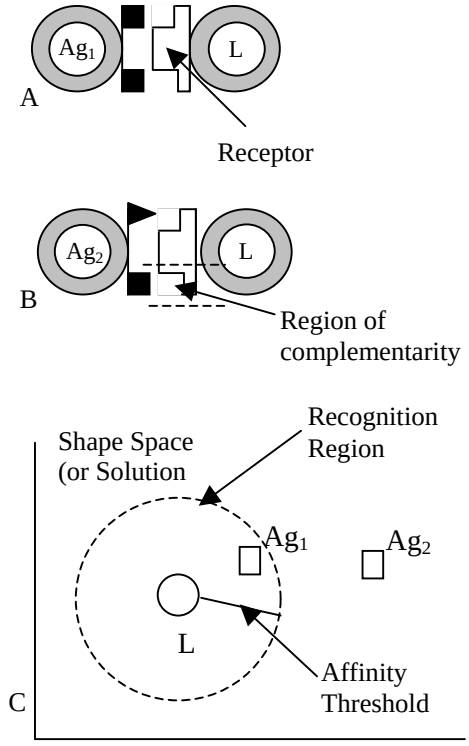

Figure 2. (A) depicts a lymphocyte (L) binding with high affinity to an antigen (Ag_1), whereas (B) depicts a binding between an antigen (Ag_2) with fewer regions of complementarity compared with the same lymphocyte. This results in a bind with lower affinity, and so L may not become activated by Ag_2. (C) shows the relative positions of L and the complement of Ag_1 and Ag_2 in shape space. Ag_1 is recognized by L as the affinity between the two is higher than the affinity threshold.

Having discussed the representation of a lymphocyte and a notion of similarity between lymphocytes and antigens, we can now describe the processes that manipulate populations of these lymphocytes. We begin with the process by which lymphocytes are created. This creation occurs in the *bone marrow* and the newly created

lymphocytes are known as *naïve* lymphocytes as they have not yet become stimulated. During generation the shape of the cell's receptor is dictated by a random concatenation of different gene components. The receptor does require a basic shape to function and so elements taken from libraries of genes are used to encode the relevant parts of the receptor. It is the job of gene library algorithms to generate repertoires of immune cells. These gene libraries are used where the feature vector requires a certain structure, discrete or symbolic values are required or random generation of the feature vector is otherwise inappropriate.

Upon activation, the two types of lymphocyte will behave differently. Considering first a B-Cell, whose job it is to tag an antigen for destruction, this B-cell must bind tightly to the antigen and stay bound until the antigen can be destroyed. It is quite possible for no B-cells in the body to have high enough affinity to bind. For this reason, an activated B-cell will begin a process of cloning and receptor mutation called *clonal selection*. Strong selective pressures during this proliferation process have the effect of maximizing affinity with the antigen and so increasing the effectiveness of the immune response. In the AIS world, an activated immune cell may adapt to new data in a similar way. Upon activation the artificial cell may undergo a process of cloning with a rate proportional to the antigenic affinity. Each new clone is mutated with a rate inversely proportional to the affinity with the antigen. Both of these processes have the goal of moving the cell closer to the antigen within the solution space. An adaptation process such as this is a common paradigm found in many evolutionary algorithms but asexual reproduction and mutation with rate dependent on some fitness measure are an important difference between AIS and these others. After the activation a few clones with high affinities will live on to provide some memory of the event in the form of *memory cells*, although this is sill a point for debate. The process described above is summarized by Figure 3.

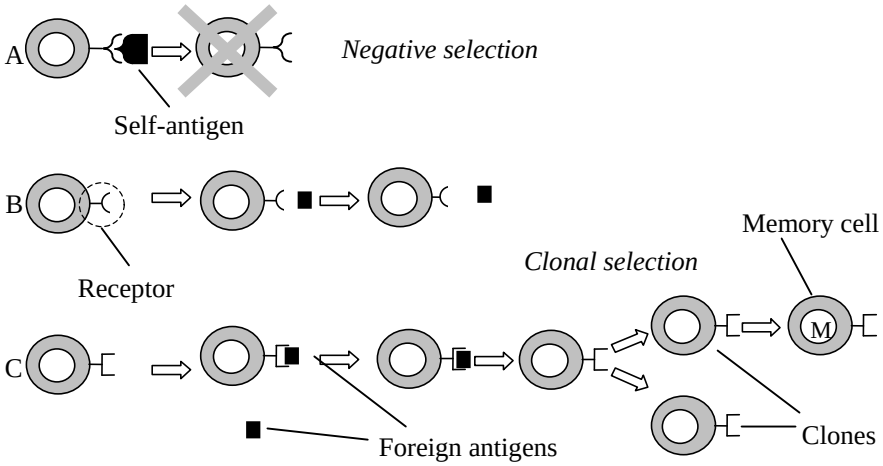

Figure 3. (A) Cell receptor matches pattern belonging to the host, cell is removed by negative selection. (B) Cell receptor does not match antigenic shape and so cell is left unstimulated. (C) Cell receptor matches foreign antigenic shape (unlike cell in B) and so is selected for cloning. Cell becomes activated and produces clones, some of which become memory cells. A modified version of a diagram from (deCastro & Timmis, 2002).

The fact that a lymphocyte may bind to any cell comes with a problem: if when T-cells (the body's powerful adaptive killers) are produced their receptor is in a random configuration why do these not bind to cells of the host? In the natural system the body is purged of these cells before they are able to circulate and initiate an *auto-immune response* by a process called *negative selection* (Forrest, Perelson, Allen, & Cherukuri, 1994) as depicted in Figure 3. Immature T-cells mature in a part of the body called the thymus where the chance of encountering a foreign antigen is negligible. While in the thymus these young T-cells will die upon binding with anything and so it can be assumed that the only cells able to leave as mature cells are those that are not capable of binding to cells of the host. We say they are only capable of binding with *non-self*. Negative selection is a major component in a number of AIS algorithms. In these, a set of cells is compared against a set of patterns corresponding to self and the resulting affinities evaluated. Any antibody with a high affinity to an element of this self set will be eliminated thus leaving antibodies capable of recognizing only non-self examples.

The thymus is one small part of the host and it would be impossible to remove all potentially self-reactive cells based on negative selection alone because not all 'self' patterns may be present. The *two-signal* model as described in (Matzinger, 2002b) is used for purging self reactive cells once they have left the thymus. A T-cell needs two signals to become activated. The recognition of an antigen by a T-cell is said to be signal one, the second signal or *co-stimulation* signal is a confirmation and is given to the T-cell by an *antigen presenting cell* upon proper presentation of the antigen. If the T-cell has received signal one in the absence of signal two, it has bound to a cell not properly presented, assumed to be part of the host. The T-cell is removed. This extra layer of protection allows potentially self-reactive cells to roam the body without beginning an *autoimmune* reaction.

Danger Theory

As described in the introduction it is our goal to take inspiration from a theory called the Danger theory to realize a unique web mining system for discovery of interesting information. There are currently few AIS publications that even mention the existence of Danger theory. Notable exceptions are (Williamson, 2002) in which the author mentions Danger theory in a small section and (Aickelin & Cayzer, 2002) which is currently

the only dedicated paper discussing the potential application of Danger theory to AIS. Now we have explained the traditional view of the immune system, we can briefly describe the Danger theory, why it is significant and how we may put it to use.

Widely attributed to Polly Matzinger, the Danger theory (Matzinger, 2002a, 2002b) attempts to explain the nature and workings of an observed immune response in a way different to the more traditional view. This view is that immune cells cannot attack their host because any cells capable of doing so are deleted as part of their maturation by negative selection. However, this view has come under some criticism as observations demonstrate that it may sometimes be necessary for the body to attack itself and conversely the immune system may not attack cells it knows to be foreign. Matzinger argues a more plausible way to describe the triggering of an immune response is a reaction to a stimulus the body considers *harmful*. Conceptually a very small change but a complete paradigm shift in the field of immunology (Anderson & Matzinger, 2000). This model allows foreign and immune cells to exist together, a situation impossible in the traditional standpoint. However, when under attack, cells dying unnaturally may release a *danger signal* (Gallucci & Matzinger, 2001), that disperses to cover a small area around that cell: a *danger area.* It is within this and only within this area that the immune system becomes active and will concentrate its attack against any antigen within it.

As the immune response is initiated by the tissues themselves in the form of the release of danger signal, it is thought the nature of this response may also be dictated by the tissues, another immunological paradigm shift. It has long been known that in a given part of the body an immune response of one class may be efficient, but in another may harm the host. Different types of danger signal may influence the type of immune response. This gives rise to a notion that tissues protect themselves and use the immune system to do so and is in stark contrast to the traditional viewpoint in which the roles are reversed and it is the immune system's role to protect tissues. There is still much debate in the immunological world as to whether the Danger theory is the correct explanation for observed immune function but if the Danger theory is a good metaphor on which to base an artificial immune system then it can be exploited.

The Danger Theory and Artificial Immune Systems

The concepts we identify in the Danger theory which we believe are of use are a context dependent and localized response, the class of which may be determined also based on context. Firstly, the natural immune

system reacts to a danger signal but, in an AIS this signal may signify almost anything. For example, in network security, a host may raise a danger signal if it is attacked, but in a text mining context, as suggested in (Aickelin & Cayzer, 2002), we may raise an “interesting document” signal or in the context of e-mail classification a “mailbox is full” signal may be appropriate. This signal, whatever its nature, will be raised based on a current context, where this context may be some measure of interestingness or mailbox capacity for the above examples, which may change on a day-to-day basis. Furthermore, in the natural immune system when a cell begins to release a danger signal the danger area is spatial. An example of a use of this spatial area in a web mining system may be the generation of an “interesting” area around a document found on a website. All pages within, say, one hyperlink of this page are also in the “interesting” area. Unlike the natural immune system however, we are not constrained to a spatial area, and in (Aickelin & Cayzer, 2002) the possibility of a temporal danger area is also discussed. Finally the type of response may also be determined based on a current context. The Danger theory suggests that natural tissues release different types of danger signal based on the different type of pathogenic attack. In a web mining system different types of signal may be released based on the type of media causing the stimulus. An interesting e-mail may release an “interesting” signal of one class while an interesting web page may release a signal of one other class.

We have given some thought to the implementation of such an algorithm, details of which may be found in (Secker, Freitas, & Timmis, 2003), although a number of interesting research questions still remain unanswered. For example, unlike most AIS algorithms, the tissue cells play a large part in a danger inspired system but how should the behavior of these cells be implemented? For example, it may be helpful to implement a set of tissue cells in addition to the set of lymphocytes. Each individual cell may then react to a slightly different stimulus? Furthermore we may also ask how the signal released by these cells should be interpreted. Should a signal from one cell be enough to stimulate an immune response or should activation occur only after a number of cells have been stimulated? If this latter approach is chosen we may then consider an activation function for the immune system such that a certain concentration of signal over a given space or time will initiate a response. In this section we have posed a number of questions regarding the implementation of a danger inspired system. The final design of such a system and therefore the answers to these questions would be very much dependent upon the problem domain. It is these sorts of questions we would like our final web mining system to answer but before we can begin realization of such a system we must first determine if an immune inspired algorithm is a suitable choice for the task of text mining.

Bayesian Classification

At the end of this chapter we compare the system proposed in the following section against a standard technique, in this case a naïve Bayesian classifier. Naïve Bayesian classifiers (Friedman & Kohavi, 2002; Weiss & Kulikowski, 1991; Mitchell, 1997) are a popular technique used for classification and especially popular for the classification of e-mail (see e-mail classification section) and we consider a brief explanation of the Bayesian learning paradigm a constructive addition at this stage.

In the classification task of machine learning it is our goal to assign a class to an instance based on the values of a number of attributes. A Bayesian classifier will not attempt to define a particular relationship between these attributes and the class of the instance, instead the probabilities of an instance belonging to each possible class is estimated, based on the training data, and the instance is assigned the class that is most probable. As Bayesian classifiers have roots in statistical mathematics they possess properties that are mathematically provable, and therefore desirable for many applications. One of these is it can be shown that in theory a Bayesian classifier will reach the smallest possible classification error given a sufficiently large training set. Although in practice this may not be the case do to the need for simplifying assumptions, described later. In addition to this, probabilistic methods may be employed to deal with missing values and asymmetric loss functions. That is, situations where the cost of misclassifying examples of one class may far outweigh the cost of misclassifying examples of another. For example, classifying an interesting e-mail as uninteresting and removing it is a lot less desirable than to allow uninteresting e-mail into the user's inbox (Diao, Lu, & Wu, 2000).

The Bayes theorem is the cornerstone of Bayesian learning. Figure 4 describes how we can derive the equation used for the naïve Bayesian classifier from the Bayes theorem and an equation to return the most probable class given a set of features. As described by (Mitchell, 1997), the probability of observing hypothesis *h* given the training data *D*, may be given by formula (1). In Bayesian learning we assign the most probable class v_{mp} from a finite set, *V*, based on a set of attribute values $<a_1,a_2 \ldots a_n>$ as described by formula (2). The Bayes theorem (1) and the equation to determine the most probable class (2) can be combined to produce (3) as shown in Figure 4.

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \quad (1)$$

$$v_{mp} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2 ... a_n) \quad (2)$$

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

$$v_{mp} = \arg\max_{v_j \in V} P(a_1, a_2 ... a_n \mid v_j)P(v_j) \quad (3)$$

Figure 4. Derivation of Naïve Bayesian equation

In Figure 4 equation (3), $P(v_j)$ can be estimated simply by counting the frequency with which each class appears in the training data, however the first term is a lot harder to determine. In practice we would need to see every possible instance in the problem space a number of times in order to provide reliable estimates. The naïve Bayes classifier introduces the assumption that attribute values are conditionally independent an therefore the probability of observing $a_1,a_2 \ldots a_n$ is the product of the probabilities observing each attribute independently. This results in the approach used by the naïve Bayesian classifier as defined in Equation 1.

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$

Equation 1. Naïve Bayesian Classifier

The terms in Equation 1 are usually calculated using frequency counts over the training data. However, it is quite likely that we will encounter a term unknown to the system. Assuming the frequency count to simply be 0 (i.e. P(new word|junk) = 0/100 = 0) would rule out this class entirely as this zero term results in the calculated probability for this class always evaluating to 0. A number of methods have been suggested for substituting a suitable probability for this value, although each comes with its own form of associated bias. Some implementations simply ignore this term, but a common strategy is to replace the probability with a small, non-zero number. Examples of this would be replacement with 1/n, where n is the number of training examples, which has the advantage that this represents the increasing certainty that this element must have an almost-zero value with the increasing size of the training set. The probability may also be replaced by 1/m where m is the number of attributes and is the strategy we adopt later in this chapter. For a worked example of naïve Bayesian classification used for classifying documents, the reader is referred to (Mitchell, 1997) p 180.

E-mail classification

As explained later in this chapter it is our task to turn an AIS towards the classification of electronic mail (e-mail). There have been a number of strategies for this task discussed in the literature and the systems proposed broadly fall into two groups: spam filters and e-mail organizers. Spam (Graham, 2003) is a term used to describe e-mail that is unsolicited, sent in bulk and usually with a commercial objective. These systems typically classify incoming messages into only two classes, legitimate e-mail and spam e-mail, before these e-mails reach the user client. Two techniques which have been common are collaborative methods in which many users share their knowledge of junk e-mail to construct a central 'blacklist' and rule-based in which rules are used for classification of incoming e-mail. Although as spam is constantly changing in content and style, the accuracy of both these techniques may suffer. For this reason machine learning techniques, are increasingly employed to tackle the problem of spam e-mail. Typically all these filters hide spam messages from the user, but for this to be acceptable safeguards may be usually put in place to ensure false classification of legitimate e-mail (which may be important to the user) is not removed accidentally. (Androutsopoulos et al., 2000) is one such example in which this asymmetric loss function is accounted for. The authors compare a naïve Bayesian approach of spam removal to a memory based approach and assume that discarding a legitimate e-mail is as bad as classifying 999 spam e-mails as legitimate. The classifiers are biased accordingly. Recently the authors of (Cunningham et al., 2003) investigated a case-based approach to spam filtering with the added feature that, like the system we detail in this chapter, it may track concept drift. This is a phenomenon where the concept of what the user finds interesting may change over time and so too may the content of uninteresting e-mails. An example of this may be the use of the word "ca$h" where the word "cash" was once used in spam e-mail such as advertisements.

E-mail organizers differ from spam filters in that they may work with more than two classes of e-mail, and the job of this type of classifier tends to be to assign a folder to a message based on its content from within the user client. For example, assigning the labels "work" or "friends" to a message and assigning it the appropriate folder. Two e-mail organization systems from the literature are MailCat (Segal & Kephart, 1999), which integrated into the Lotus Notes client, and ifile (Rennie, 2000), which may integrate into the EXMH mail client. MailCat uses a Term Frequency-Inverse Document Frequency (TFIDF) approach to class assignment, a popular technique in the world of text mining. By contrast, ifile uses a naïve Bayesian technique (similar to that described in the previous section) to sort messages into folders. Four users tested the ifile system and the results

show that users could expect a typical classification accuracy of between 85% and 90%. This Bayesian classification technique proves common in the literature. For example (Diao, Lu, & Wu, 2000) compares a naïve Bayesian system against the C4.5 decision tree algorithm and it was found that, although C4.5 can classify e-mail with greater accuracy, the Bayesian system was more robust overall. Similarly, (Yang & Park, 2002) compares the TFIDF approach (described above) with a naïve Bayesian classifier and conclude that the Bayesian system provides a better classification accuracy in almost all cases. This TFIDF approach is also investigated in (Brutlag & Meek, 2000) who also compare this to discriminant classifiers and a classifier based on a language model approach. The results showed that neither one of these three techniques was constantly superior and that the accuracy varies more between mail stores than the tested classifiers. A review of a number of research based and practical systems for spam e-mail removal and more general e-mail organization can be found in (Crawford, Kay, & McCreath, 2001).

AN AIS FOR E-MAIL CLASSIFICATION

As a step towards our goal we felt it was important to produce a text mining system based on an immune inspired algorithm. This must then be tested in a dynamic domain. For this reason we took the decision to gauge the performance of our text mining system on the task of e-mail. Our chosen task is to distinguish between e-mail the user would not be interested in, and legitimate e-mail, which to the user is important or interesting with the choice being made dependent upon previous experience. We consider e-mail classification to be essentially a web content mining task as defined in the introduction, as the text contained in the e-mail is used for the purposes of classification and e-mail is a part of the Internet environment. As explained in the introduction, this system has been written as a step towards an algorithm for mining interesting information from the web and so even though it is performing a task similar to a spam filter, as described in the previous section, we acknowledge it has no special measures to cope with this asymmetric loss function. The penalty for misclassifying an interesting document when the final system is run is not nearly as severe as misclassifying an e-mail. The novel system we propose posses a number of features, the combination of which dissociates it with those systems previously described. The main difference is that we address a continuous learning scenario. This contrasts with the vast majority of those systems above which are trained once and then left to run. In addition to this we address concept drift, a feature implicit in the continuous learning scenario and a feature few other e-mail systems possess. One further advantage of an AIS is that our e-mail classifier requires no specific feature

selection mechanisms. In contrast to some systems described above we do not pre-select a set of words from the training data, instead a selection is performed in a data driven manner implicitly by the evolutionary operators.

The "AISEC" Algorithm

AISEC seeks to classify unknown e-mail into one of two classes based on previous experience. It does this by manipulating the populations of two sets of immune cells. Each immune cell combines some features and behaviors from both natural B-cells and T-cells. For simplicity we refer to these as B-cells throughout. These two sets consist of a set of naïve (sometimes called free) B-cells and a set of memory B-cells, a biologically plausible notion as described in the background section. Once the system has been trained, each B-cell encodes an example of an uninteresting e-mail. New e-mails to be classified by the system are considered to be antigens. To classify an e-mail (antigen), it is first processed into the same kind of feature vector as a B-cell and presented to all B-cells in the system. If the affinity between the antigen and any B-cell is higher than a given threshold, it is classified as uninteresting otherwise it is allowed to pass to the user's normal inbox. If the antigen (e-mail) is classified uninteresting it will be removed to a temporary store. If the user deletes an e-mail from the temporary store it is confirmed to represent an uninteresting e-mail. The B-cell that classified it as uninteresting is useful and is rewarded by promotion to a long-lived memory B-cell (assuming it was not already) and is selected for reproduction. This constant reproduction combined with appropriate cell death mechanisms give the AISEC algorithm its dynamic nature. A high level outline of this process is shown in Figure 5.

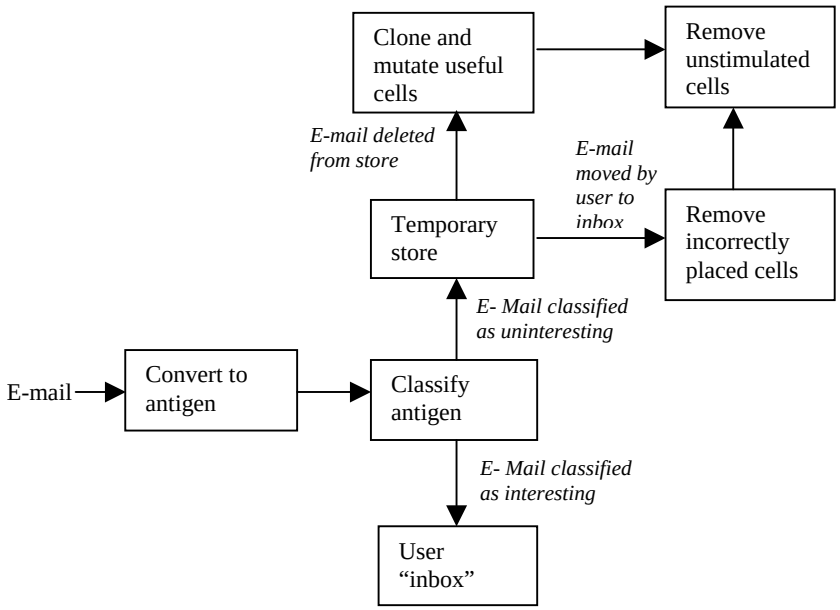

Figure 5. High level view of the AISEC algorithm's process after initial training

Once an e-mail has been placed in the user's inbox either by classification or by the user him or herself it is no longer accessible to the algorithm. When the user removes mail to save space it is assumed he/she will do so by removing mail from the mail client's inbox, thus having no effect on the algorithm. As the folder where uninteresting e-mail is placed is nothing more than a temporary store it should be emptied regularly.

During design a number of special considerations were given to the specialist nature of the text mining domain. The incorporation of these considerations in the final algorithm served to further distance our system from other AIS. These design decisions are discussed below:

Representation of one data class: In a web-mining context, learning types of documents a user finds interesting may be tiny compared with those a user finds uninteresting. B-cells therefore represent only the uninteresting e-mail class. A helpful simplification for the purposes of efficiency and more akin to the way the natural system works. Natural lymphocytes only encode possible pathogenic patterns and everything else is assumed harmless.

Gene libraries: Two libraries of words, one for subject words and one for sender words are used. These contain words known to have been previously used in uninteresting e-mail. When a mutation is performed, a word from this library replaces a word from a cell's feature vector. Mutating a word in any other way, by replacing characters for example, would result in a meaningless string in almost all cases. All new cells entering the naïve cell set are mutants of existing cells

Co-stimulation: Uninteresting e-mail is not deleted but stored away. A B-cell must have become stimulated to classify this e-mail, so it can be assumed the first signal has already occurred. User feedback is then used to provide or not provide a second signal. At a time of the user's convenience this store may be emptied. It will be these user actions that will drive a number of dynamic processes. If an e-mail is deleted from this store by the user the system has performed a correct classification, the user really wasn't interested in that e-mail and so a co-stimulation signal has occurred. The cell is rewarded by being allowed to reproduce. If, on the other hand, the user does not delete the e-mail it is assumed the system has performed a misclassification, signal two does not occur and artificial cells may be deleted as appropriate.

Two recognition regions: Around each B-cell is a recognition region; the region within which the affinity between this cell and an antigen is above a threshold. It is within this region a cell may stimulate another. A single region was found to be insufficient for both the triggering of evolutionary processes and classification. A smaller region, a classification region, was introduced for use in classification only. Empirical studies suggested

the introduction of this second region was shown to increase the classification accuracy from around 80% to around 90% on the test set.

Cell death processes: To both counteract the increase in population size brought about by reproduction and keep the system dynamic, cell death processes must be implemented. A naïve B-cell has not proved its worth and is simply given a finite lifespan when created, although it may lengthen its life by continually recognizing new pieces of data confirmed as uninteresting. Memory B-cells may also die, but these cells have proved their worth and it can be hard for the system to generate clones capable of performing well. For this reason unlike naïve B-cells, memory cells are purged in a data driven manner. When a new memory cell *mc*, is added to the memory cell set all memory cells recognizing *mc* have a stimulation counter reduced. When this count reaches zero they are purged from the system. This dissuades the system from producing an overabundance of memory cells providing coverage over roughly the same area, when one is quite sufficient.

The Algorithm in Detail

Before we begin, let us establish the following notational conventions:

- Let `BC` refer to an initially empty set of naïve B-cells
- Let `MC` refer to an initially empty set of memory B-cells
- Let `Kt` refer to the initial number of memory cells generated during training
- Let `Kl` refer to the clone constant which controls the rate of cloning
- Let `Km` refer to the mutation constant which controls the rate of mutation
- Let `Kc` refer to the classification threshold
- Let `Ka` refer to the affinity threshold
- Let `Ksb` refer to the initial stimulation count for naïve B-cells
- Let `Ksm` refer to the initial stimulation count for memory B-cells

Representation

A B-cell receptor holds information that may be extracted from a single e-mail, this is represented as a vector of two parts (see Figure 6). One part holds words present in the subject field of a single e-mail and the second holds words present in the sender (and return address) fields of that particular e-mail. The actual words are stored in the feature vector because once set the vector will not require updating throughout the life of the cell. This can be contrasted to the common practice of using a vector containing binary values as the receptor, each

position in which represents the presence or absence of a word known to the system. As words are continually being added and removed from the system each cell's vector would have to be updated as appropriate when this action occurs. The two sub-vectors are unordered and of variable length. Each B-cell will contain a counter used for aging the cell that is initialized to a constant value on generation and decremented as appropriate. This counter may be re-initialized if the B-cell is added to `BC`.

```
B-cell vector = <subject,sender>
subject = <word 1,word 2,word 3,...,word n>
sender = <word 1,word 2,word 3,...,word m>
```

Figure 6. B-cell structure

Affinity Measure

The *affinity* between two cells measures the proportion of one cell's feature vector also present in the other cell. It is used throughout the algorithm and is guaranteed to return a value between 0 and 1. The matching between words in a feature vector is case insensitive but otherwise requires an exact character-wise match. `bc1` and `bc2` are the cells we wish to determine the affinity between, as shown in Pseudocode 1.

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

Pseudocode 1. Affinity

Algorithms and processes

The AISEC algorithm works over two distinct stages: a training phase followed by a running phase. The running phase is further divided into two tasks, that of classifying new data and intercepting user feedback to allow the system to evolve. An overview of this algorithm is described in Pseudocode 2.

```
PROGRAM AISEC
    train(training set)
    WAIT until (an e-mail arrives or a user action is intercepted)
        ag ← convert e-mail into antigen
        IF(ag requires classification)
            classify(ag)
            IF(ag is classified as uninteresting)
                move ag into user accessible storage
            ELSE
                allow e-mail to pass through
        IF(user is giving feedback on ag)
            update_population(ag)
```

Pseudocode 2. AISEC overview

We now detail each of these three stages in turn, training, classification and the updating of the population based on user feedback. During the training stage the goal is to populate the gene libraries, produce an initial set of memory cells from training examples, and produce some naïve B-cells based on mutated training examples. As the B-cells in the AISEC system represent one class only the entire training set, here called TE, contains only e-mails the user has positively selected to be uninteresting. This is described in Pseudocode 3.

```
PROCEDURE train(TE)
    FOREACH(te ∈ TE)
        process e-mail into a B-cell
        add subject words and sender words to appropriate library
    remove Kt random elements from TE and insert into MC
    FOREACH (mc ∈ MC)
        set mc's stimulation count to Ksm
    FOREACH (te ∈ TE)
        set mc's stimulation count to Ksb
        FOREACH (mc ∈ MC)
```

```
        IF(affinity(mc,te) > Ka)
            clones ← clone_mutate(mc,te)
            FOREACH (clo ∈ clones)
                IF(affinity(clo,bc) >= affinity(mc,te))
                    BC ← BC ∪ {clo}
```

Pseudocode 3. Training

Now the system has been trained it is available to begin two distinct functions. These are the classification of unknown e-mail and population update processes based on user feedback on the correctness of classification attempts. During the running phase the system will wait for either a new mail to be classified or an action from the user indicating feedback. Upon receipt of either of these, the system will invoke the necessary procedure as outlined in either Pseudocode 4 or Pseudocode 5. To classify an e-mail, an antigen, `ag,` is created in the same form as a B-cell, taking its feature vector elements from the information in the e-mail, then assigned a class based on the procedure described by Pseudocode 4.

```
PROCEDURE classify(ag) returns a classification for ag
    FOREACH (bc ∈ (BC ∪ MC))
        IF(affinity(ag,bc)) > Kc)
            classify ag as uninteresting
            RETURN
    classify ag as interesting
    RETURN
```

Pseudocode 4. Classification

To purge the system of cells which may match interesting e-mails, the AISEC algorithm uses the two signal approach as outlined in the background section of this chapter. Since signal one has occurred, that is, the instance has already stimulated a B-cell and been classified. Signal two comes from the user, in the form of interpreting the user's reaction to classified e-mail. It is during this stage that useful cells are stimulated and unstimulated cells are removed from the system. antigen `ag` is the e-mail on which feedback has been given.

```
PROCEDURE update_population(ag)
    IF(classification was correct)
        FOREACH(bc ∈ BC)
            IF(affinity(ag,bc) > Ka)
                increment bc's stimulation count
        bc_best ← element of BC with highest affinity to ag
        BC ← BC ∪ clone_mutate(bc_best,ag)
        bc_best ← element of BC with highest affinity to ag
        mc_best ← element of MC with highest affinity to ag
        IF(affinity(bc_Best,ag)> affinity(mc_best,ag))
            BC ← BC \{bc_best}
            bc_best's stimulation count ← Ksm
            MC ← MC ∪ {bc_best}
            FOREACH(mc ∈ MC)
                IF(affinity(bc_best,mc) > Ka)
                    decrement mc stimulation count
                    add words from ag's feature vector to gene libraries
    ELSE
        FOREACH(bc ∈ (MC ∪ BC))
            IF(affinity(bc,ag) > Ka)
                remove all words in bc's feature vector from gene libraries
                delete bc from system
    FOREACH(bc ∈ BC)
        decrement bc's stimulation count
    FOREACH(bc ∈ (MC ∪ BC))
        IF(bc's stimulation count = 0)
            delete bc from system
```

Pseudocode 5. Update B-cell population

The process of *cloning and mutation* which has been used throughout this section is detailed in Pseudocode 6. B-cell `bc1` is to be cloned based on its affinity with B-cell `bc2`. Constants `Kl` and `Km` are used to control the rate of cloning and mutation. The symbol $\lfloor x \rfloor$ denotes the "floor" of x. That is, the greatest integer smaller than or equal to the real-valued number x and is necessary because `num_clones` and `num_mutates` must be integers.

```
PROCEDURE clone_mutate(bc1,bc2) returns set of B-cells
    aff ← affinity(bc1,bc2)
    clones ← ∅
    num_clones ← ⌊ aff * Kl ⌋
    num_mutate ← ⌊ (1-aff) * bc's feature vector length * Km ⌋
    DO(num_clones) TIMES
        bcx ← a copy of bc1
        DO(num_mutate) TIMES
            p ← a random point in bcx's feature vector
            w ← a random word from the appropriate gene library
            replace word in bcx's feature vector at point p with w
        bcx's stimulation level ← Ksb
        clones ← clones ∪ {bcx}
    RETURN clones
```

Pseudocode 6. Cloning and mutation

RESULTS

To determine the relative performance of AISEC, it was necessary to test it against another continuous learning system. The naïve Bayesian classifier explained previously, was chosen as a suitable comparison algorithm. Even though the fundamental assumption of naïve Bayes, that all attributes are independent, is violated in this situation Mitchell (1997) states "*probabilistic approaches such as the one described here [naïve Bayesian] are among the most effective currently known to classify text documents*" (p. 180). An implementation of the naïve Bayesian classifier was implemented by the first author that was adapted to intercept input relating to

classification accuracy in the same way as the AISEC system. This was done according to Equation 1, where the set V = {uninteresting, interesting}, $P(v_j)$ is the probability of mail belonging to class V_j and calculated based on the frequency of occurrence of class V_j. The term $P(a_i|v_j)$ is the probability of the e-mail containing word a_i given the e-mail belongs to class V_j. These probabilities are calculated using observed word frequencies over the data the system has been exposed to and so frequencies may be updated based on user input much as in AISEC. The default probability assigned to an unknown word was $1/k$ where k is the total number of words known to the system.

Experimental Setup

Experiments were performed with 2268 genuine e-mails, of which 742 (32.7%) the first author manually classified as uninteresting and the remaining 1526 (67.3%) were considered of some interest. Due to the unsuitability of the few publicly accessible e-mail datasets which are traditionally used for single shot learning, unlike the continuous learning scenario discussed in this chapter, we were unable to test the system on a set of benchmark e-mails. All e-mails used were received by the author between October 2002 and March 2003, and their date ordering was preserved. This temporal ordering is reflected in the order in which the e-mails are presented and should allow both systems to adapt to any drifting concepts and changing e-mail text. When processed, the sender information also included the return address, as this may be different from the information in the sender field. These fields were tokenized using spaces and the characters ".", ",", "(", ")", "!", "@", "<", ">" as delimiters. During the runs of the AISEC algorithm, the same values for all parameters were used. These values were arrived at by trial and error during testing and tend to work well over this dataset (see Table 1). The naïve Bayesian system was trained on the oldest 25 e-mails as both classes are required for training, the AISEC system was trained on the oldest 25 uninteresting examples only with the remainder of both used as a test set.

Kc (classification threshold)	0.2
Ka (affinity threshold)	0.5
Kl (clone constant)	7.0
Km (mutation constant)	0.7
Ksb (Naïve B-cell stimulation level)	125
Ksm (Memory cell stimulation level)	25
Kt (initial number of memory cells)	20

Table 1. Parameter values

Unlike traditional single shot learning, where there is a fixed test set, we address continuous learning where the system is continually receiving e-mails to be classified. Each time a new e-mail is classified the system can use the result of this classification (the information about whether or not the class assigned was correct) to update its

internal representation. This continuous learning scenario calls for a slightly different measure of accuracy to that which is normally applied. Conceptually, as there is no fixed "test set" the system keeps track of its performance over the past 100 classification attempts. As each e-mail is classified an average accuracy over these previous attempts is reported. The final classification accuracy is determined by taking the mean of all these values. As AISEC is non-deterministic the result presented in Table 1 is the average of ten runs using a different random seed each time. The value after the "±" symbol represents the standard deviation. The result for the naïve Bayesian algorithm has no standard deviation associated with it as, since it is a deterministic algorithm, just a single run was performed.

Algorithm	**Mean Classification Accuracy**
Bayesian	88.05%
AISEC	89.09% ± 0.965

Table 2. Results for continuous learning task

From Table 2 we can see that the AISEC algorithm can classify the e-mails in the given continuous test set with a slightly higher accuracy compared with the Bayesian approach, although we do not claim it classifies with higher accuracy in general. Instead, based on these results, we think it is reasonable to conclude that our algorithm performs with accuracy comparable to that of the Bayesian algorithm but with dynamics very different to that algorithm. We also undertook an experiment that assessed the performance of the algorithm when run in a traditional one-shot learning scenario. In this case the evolution of the system was stopped after the initial training e-mails and no feedback mechanisms were able to evolve the sets of B-cells from that point onwards. These results suggested the performance was surprisingly good with mean predictive accuracies just 5% lower than with the user feedback mechanism. From this we suggest that the user feedback mechanisms are useful for the continued accuracy of the system, but not essential for this AIS to function well. This has been previously demonstrated by the AIS-based classifier, AIRS (Watkins & Timmis, 2002).

The line chart Figure 7 details the classification accuracy after the classification of each mail. This uses the accuracy measure described above and details the results for the entire test set apart from the first 100 e-mails. It can be seen that both algorithms are closely matched in general but there are certain areas where the changing data causes them to behave differently. Of interest are the areas between 1,000 and 1,250 and again between 1,900 and 2,100 e-mails classified. In both situations AISEC exhibits an increase in accuracy while there is a decrease in accuracy from the Bayesian algorithm. Even after manual inspection of the data the reasons for this were undetermined. We are currently considering a more rigorous and lengthy analysis of the test data to try to

explain this interesting phenomenon. One suggestion would be that AISEC is faster to react to sudden changes. Consider, for example, a word that is very common among uninteresting e-mail. The AISEC system will represent this as the presence of this word in a number of B-cells. The Bayesian system will represent this as a high frequency of occurrence in this class compared to the frequency of it appearing in the other class. Consider now this word begins to be used in interesting e-mail. The AISEC system will react quickly by deleting any cells containing this word that would result in a misclassification. By contrast the Bayesian system will react by only incrementing the frequency count of this word in the interesting class. Given the word has been common in uninteresting e-mail for some time the frequency of occurrence in this class will still be large compared with frequency of occurrence in the interesting class and so will have a negligible effect on the final calculated class probability. Only after this word has been used many times in confirmed interesting e-mail the differences in the frequencies of usage may even out, and the difference in the probabilities this word being used in each class significantly decrease.

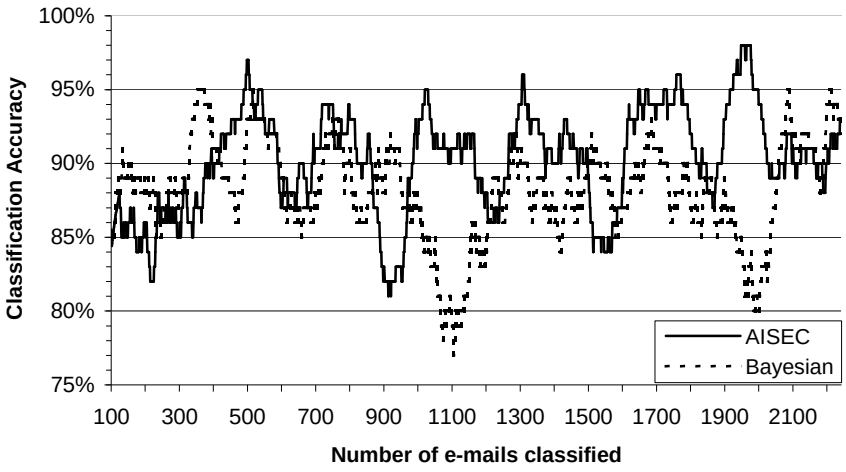

Figure 7. Change in classification accuracy over time

CONCLUSION AND FUTURE WORK

As a first step towards a danger model based artificial immune system for web mining, we have described a novel immune inspired system for classification of e-mail. We have shown that an immune inspired algorithm written especially with text mining in mind may yield classification accuracy comparable to a Bayesian approach in this continuous learning scenario. We have gone some way to showing an immune inspired system is capable of the specialized task of document classification using a text-mining approach. The results presented were generally encouraging but it is clear there is still work to be done to optimize such a system. An increase in

accuracy may be achieved by a change in the kind of features stored in the B-cell's feature vector, such as a measure of the relative importance of words and coupled with the necessary change in affinity function. An improvement in accuracy may also be made by the use of body text from the e-mail, stopword removal, stemming or words or perhaps the use of training data to optimize the algorithm's parameters.

We feel that the AISEC algorithm has shown an AIS based algorithm can perform text-based classification with accuracy comparable with a naïve Bayesian classifier. We now wish to push forward with a more complex system. We would like to continue this project by investigating the use of Danger theory. This next step will be to extend AISEC to work in a danger-based scenario. In this scenario the concept of interestingness of an e-mail is more dynamic because it depends not only on the contents of the e-mail (as in this chapter) but also on the current status of the mailbox. In particular when the mailbox is nearing capacity this may be interpreted as a danger signal and appropriate action taken. The ultimate goal of this work is to develop a web mining system based on the danger model. The AISEC algorithm is one step in that direction and it is hoped that continued investigation will lead us further towards our goal.

REFERENCES

Aickelin, U., & Cayzer, S. (2002). *The Danger Theory and Its Application to Artificial Immune Systems.* Paper presented at the First International Conference on Artificial Immune Systems (ICARIS 2002) (pp.141-148), Canterbury, UK.

Anderson, C. C., & Matzinger, P. (2000). Danger: The view from the bottom of the cliff. *Seminars in Immunology, 12*(3), 231-238.

Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., & Stamatopoulos, P. (2000). *Learning to filter spam e-mail: A comparison of a naïve Bayesian and a memory-based approach.* Paper presented at the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lyon.

Blake, C. L., & Merz, C. J. (1998). *UCI Repository of machine learning databases*. Retrieved 20 May 2003, from http://www.ics.uci.edu/~mlearn/MLRepository.html

Brutlag, J. D., & Meek, C. (2000). *Challenges of the Email Domain for Text Classification.* Paper presented at the Seventeenth International Conference on Machine Learning (ICML 2000) (pp 103-110), USA.

Chakrabarti, S. (2003). *Mining the web (Discovering Knowledge from Hypertext Data)*: Morgan Kaufmann.

Crawford, E., Kay, J., & McCreath, E. (2001). *Automatic Induction of Rules for e-mail Classification.* Paper presented at the Australian Document Computing Symposium (ADCS 2001) (pp.13-20), Australia.

Cunningham, P., Nowlan, N., Delany, S. J., & Haahr, M. (2003). *A Case-Based Approach to Spam Filtering that Can Track Concept Drift* (Technical report TCD-CS-2003-16). Dublin: Trinity College.

Dasgupta, D. (1999). An overview of AIS. In D. Dasgupta (Ed.), *Artificial Immune Systems and Their Applications* (pp. 3-21): Springer.

deCastro, L. N., & Timmis, J. (2002). *Artificial Immune Systems: A New Computational Intelligence Approach*: Springer.

Diao, Y., Lu, H., & Wu, D. (2000). *A comparative study of classification based personal e-mail filtering.* Paper presented at the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2000).

Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994). *Self-Nonself Discrimination in a Computer.* Paper presented at the IEEE Symposium on Research in Security and Privacy (pp 202-212), Los Alamitos, USA.

Friedman, N., & Kohavi, R. (2002). Bayesian Classification. In W. Klosgen & J. M. Zytkow (Eds.), *Handbook of Data Mining and Knowledge Discovery* (pp. 282-288): Oxford University Press.

Gallucci, S., & Matzinger, P. (2001). Danger signals: SOS to the immune system. *Current Opinion in Immunology, 13*(1), 114-119.

Google. (2003). *Google homepage*. Retrieved 23 May 2003, from www.google.com

Graham, P. (2003). *A Plan for Spam*. Retrieved 23 April, 2003, from http://www.paulgraham.com/spam.html

Hunt, J. E., & Cooke, D. E. (1996). Learning using an artificial immune system. *Journal of Network and Computer Applications, 19*(2), 189-212.

Kim, J., & Bentley, P. J. (2001). *An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection.* Paper presented at the Genetic and Evolutionary Computation Conference 2001 (GECCO 2001) (pp. 1330-1337), San Francisco USA.

Linoff, G. S., & Berry, M. J. A. (2001). *Mining the web (Transforming Customer Data into Customer Value)*: Wiley.

Liu, B., Ma, Y., & Yu, P. S. (2001). *Discovering unexpected information from your competitors' web sites.* Paper presented at the Seventh International Conference on Knowledge Discovery and Data Mining (KDD 2001) (pp. 144-153), San Francisco, USA.

Matzinger, P. (2002a). The Danger Model: A Renewed Sense of Self. *Science, 296*, 301-305.

Matzinger, P. (2002b). *The Real Function of The Immune System or Tolerance and The Four D's*. Retrieved 30/10/2002, from http://cmmg.biosci.wayne.edu/asg/polly.html

Mitchell, T. M. (1997). Bayesian Learning. In C. L. Liu & A. B. Tucker (Eds.), *Machine Learning* (pp. 154-200): McGraw-Hill.

Rennie, J. D. M. (2000). *ifile: An Application of Machine Learning to Mail Filtering.* In proceedings of the KDD-2000 Workshop on Text Mining, Boston, USA.

Secker, A., Freitas, A. A., & Timmis, J. (2003). A Danger Theory Inspired Approach to Web Mining. In J. Timmis, P. Bentley & E. Hart (Eds.), *Lecture Notes in Computer Science volume 2787 (The Proceedings of the Second International Conference on Artificial Immune Systems)* (pp. 156-167): Springer.

Segal, R. B., & Kephart, J. O. (1999, May 1999). *MailCat: An Intelligent Assistant for Organizing E-Mail.* Paper presented at the Third International Conference on Autonomous Agents (pp. 276-282).

Sompayrac, L. (1999). *How the Immune System Works*: Blackwell Science.

Timmis, J., & Knight, T. (2002). Artificial Immune Systems: Using The Immune System as Inspiration for Data Mining. In H. A. Abbass, R. A. Sarker & C. S. Newton (Eds.), *Data Mining: A Heuristic Approach* (pp. 209-230): Idea Group Publishing.

Timmis, J., & Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge Based Systems, 14*(3-4), 121-130.

Twycross, J. (2002). *An Immune System Approach to Document Classification* (Technical Report HPL-2002-288): HP Labs, Bristol, UK.

Watkins, A., & Timmis, J. (2002). *Artificial Immune Recognition System (AIRS): Revisions and Refinements.* In proceedings of The First International Conference on Artificial Immune Systems (ICARIS 2002) (pp. 173-181), Canterbury, UK.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer Systems that Learn*: Morgan Kaufmann.

Williamson, M. M. (2002). *Biologically Inspired Approaches to Computer Security* (Technical Report HPL-2002-131): HP Labs, Bristol, UK.

Yang, J., & Park, S.-Y. (2002). Email Categorization Using Fast Machine Learning Algorithms. *Discovery Science 2002*, 316-323.

Authors' Biographies

Andrew Secker received a first class B.Sc. with honors in Computer Science from the University of Kent, UK, in 2002. He is currently a Ph.D. student at the University of Kent working under the supervision or Dr Alex Freitas and Dr Jon Timmis. Andrew's research interests are in population based and biologically inspired systems. The working title for his Ph.D. is "An Artificial Immune System for Web Mining".

Dr Alex A. Freitas received the B.Sc. degree in Computer Science from the "Faculdade de Tecnologia de Sao Paulo", Brazil, in 1989; the M.Sc. degree in Computer Science from the "Universidade Federal de Sao Carlos", Brazil, in 1993; and the Ph.D. degree in Computer Science from the University of Essex, UK, in 1997. He was a visiting lecturer at the "Centro Federal de Educacao Tecnologica", in Curitiba, Brazil, from 1997 to 1998; and a lecturer at the "Pontificia Universidade Catolica", also in Curitiba, Brazil, from 1999 to 2002. Since 2002 he is a lecturer at the University of Kent, in Canterbury, UK. His publications include two books on data mining and more than 60 refereed research papers published in journals, books, conferences or workshops. He has organized two international workshops on data mining with evolutionary algorithms, and delivered tutorials on this theme in several international conferences. He is a member of the Editorial Board of the Intelligent Data Analysis - an international journal. At present his main research interests are data mining and evolutionary algorithms.

Dr. Jon Timmis is a Lecturer in Computer Science at the University of Kent and is head of the Applied and Interdisciplinary Informatics Research Group. He received his PhD in Computer Science from the University of Wales, Aberystwyth, where he worked as a Research Associate investigating the use of immune system metaphors for machine learning. He is principle investigator for a number of industrial and government funded research projects. He has served on several program committees for artificial immune systems at international conferences and has given a number of invited talks on artificial immune systems at UK and international universities. He has published over 35 papers on artificial immune system related research and is the co-author of the first book on artificial immune systems. He was the conference co-chair with Dr. Peter Bentley for the two international conferences on artificial immune systems (ICARIS) and continues to be the co-chair for the 3rd ICARIS in 2004. He is a member of the IEEE and a member of ISGEC (International Society of Genetic and Evolutionary Computation).