Web Page Classification with an Ant Colony Algorithm

Nicholas Holden
Computing Laboratory
University of Kent
Canterbury, Kent
UK, CT2 7NF
nph4@kent.ac.uk

Abstract

As the amount of information available on the internet grows so does the need for more effective data analysis methods. This paper utilizes the Ant Miner algorithm in the field of web content classification and shows that it is more effective than C5.0. It also investigates the benefits and dangers of methods to reduce the large numbers of attributes associated with web content mining such as a naïve WordNet preprocessing stage.

1. Introduction

The amount of information available on the web is huge and growing each year. According to a study conducted by Cyveillance in 2000 there were more than 2.1 billion unique web pages available for public viewing, now in 2004 Google searches more than 4.2 billion. This is great news, the web is becoming ubiquitous and almost any imaginable information is available for consumption. However, the massive amount of information is unprecedented and sifting through it all to find what one is looking for would take lifetimes. Luckily as the web has grown new and more advanced methods of scouring it for the required information have been developed. The ability to mine for specific information has become almost important as the web itself, to the point where the most popular search engine Google has been nominated for inclusion in the Oxford English Dictionary. As popular as Google is it still relies on relatively simple methods that can be improved on with modern techniques and computational power. During this paper it will be demonstrated that by utilizing more intelligent data analysis techniques it is possible to categorize web pages based on their subject. Classification accuracies using different components of a web page will be presented and different methods of processing them will be discussed. It is hoped that by harnessing the Ant Miner classification algorithm it will be possible to show that swarm intelligence can be usefully applied to the field of web content mining. Also that WordNet and other preprocessing stages can help with classification.

The rest of this paper is organized as follows. Section 2 presents a general background in classification and web mining. Section 3 discusses ant colony algorithms, with focus on the Ant-Miner algorithm (an ant colony algorithm for classification). Section 4 reports computational results. Finally, section 5 concludes the paper.

2 Background

While there is a relatively large amount of literature about web mining or about naturally inspired algorithms, the application of swarm intelligence to this field (namely Ant Colony Optimization for web-content mining) is quite unexplored. A notable example is the work by Ajith Abraham and Vitorino Ramos [Abraham et al., 2003] who apply an Ant Colony algorithm with genetic programming to a web usage problem. Another example of the application of machine learning to a web classification problem is the paper by Hwanjo Yu, Kevin Chen-Chuan Chang and Jiawei Han [Yu, et al., 2002].

2.1 Classification

Data mining consists of a set of techniques used to find patterns within a set of data and to express these patterns in a way which can be used for intelligent decision making [Witten et al., 2000]. In this project (as with many other data mining applications) the knowledge is presented in the form of a rule. A rule consists of an antecedent and a consequent. The antecedent is a set of attribute value pairs, for instance, the attribute age with a value of 17. More precisely a rule is of the form:

IF <attribute> = <value> AND <attribute> = <value> AND … THEN <class>

The class part of the rule (consequent) is the class predicted by the rule for the records where the

predictor attributes hold. The class should be an attribute value whose prediction is “useful” for the user as there is no point in discovering knowledge that does not serve any purpose. An example rule might be IF age = 17 THEN occupation is student, which would be useful for a user looking for a relationship between age and occupation. The goal of classification in terms of data mining is to use the patterns, and so rules generated from a set of training data and apply them to a set of data with an unseen class, and hopefully predict the correct class.

2.2 Web Mining

Web mining is the practice of applying data mining techniques to the collection of information known as the web. Web mining can be split into three main categories: content mining, usage mining, and structure mining.

Content mining is a technique that uses the actual information stored on the web to find patterns. It involves the automatic analysis of the text stored in the files (i.e. HTML and email), images and any other media available.

Usage mining [Abraham et al., 2003] deals with the way in which people use the web. It uses access logs from web servers to try and discover the patterns that users make when they use the web.

Structure mining uses the way in which the web is joined together. It analyses the way in which web pages link to each other through hyperlinks, for example, and tries to find useful patterns.

The idea that you can apply a generic rule to an unseen data set and predict the class seems to lend itself well to web page classification. After all in content mining, web pages can be considered just a file containing a set of attributes (words), and it should be possible to assign a class to a given web page based on its constituent words. However web mining should not be treated as a standard data mining application based on the words occurring in that web page. There are important differences and difficulties to take into account.

Firstly the amount of attributes is unusually high in comparison to simpler data mining applications, there are hundreds of thousands of words in the English language and usually, at least, hundreds of different words in each web page. So if our training set was 500 web pages there could be ten thousand attributes. This is quite a subtle problem, it does not seem that drastic at first glance but if we think about the number of potential rules (each attribute can have a value of 0 for it not being there, 1 for it being there) in this case 2^{10000}, we see that each possible rule could never be tested.

Secondly the information we want to mine is the written language within the web pages. Unfortunately the English language (all languages in general) is very complicated. There is no program at the moment that can fully understand the meaning of a given web page. Although progress has been made we can still only hope for a relatively simple interpretation.

There is some hope though, html code gives us clues to help us cut down the number of attributes [Cuter et al., 1999]. The authors of web sites leave summaries or descriptions of the web page in <meta> tags: in <meta NAME="keywords"> the content field gives us a list of keywords the author thinks is suitable for the page, there is also <meta NAME="description"> which gives us hopefully a good overview of the page’s content, of course these tags are optional in the html standard so we cannot always rely on them. Also if some text is in bold or underlined then it is usually important and can also be used as a more condensed attribute source.

A tactic that is often used by web search engines (Google) is to look at the links on a page and the links on pages linking to the current page: Description Text such a link usually contains a very brief description of the page it is linking to. This can be used in two ways. If all the link descriptions of links to a given page could be found, then a concise and accurate description of the given page could be created from them which does not rely on any one person’s perspective or the biased perspective of a page author who includes his own meta data. This is a potentially powerful tool in web mining [Chakrabartia et al., 1998], but to make a compendious description the whole web would have to be searched for links for the given page with obvious drawbacks.

Another tactic that can be used with links is to take data from the linked to page and add it to the record for the current page, possibly with a lower priority; the reasoning behind this is that if a page is linked to from a given page then they probably have related content.

2.3 Wordnet and optimization

Wordnet is an electronic lexicon that contains relationships between words in a tree like structure. It is an attempt to map the human understanding of words and their relationships into an electronic database. Wordnet can be useful in data mining applications that require the analysis of written English data.

Firstly it contains a morphological processor. This is useful as instead of having, for example, the words borrow, borrowing and borrowed we would like to just have the word borrow added to our list of attributes, this is known as stemming. This cuts down the number of attributes, making processing the data faster and also allows us to find patterns more easily. We may not be able to find a pattern with these separate words, but when they are amalgamated together into one attribute, a pattern may emerge.

Secondly as the amount of raw attributes is so high in web mining it may be useful to only use nouns as attributes, as they are usually the subject of a sentence. Hence we trade off the completeness of the information against the ability to find more useful patterns within a given time. Also this has the useful effect of removing stop words, stop words are words that convey little or no useful information in terms of text mining. Examples might be the, and, they. These words are needed for language, but as we do not try and understand language they may be removed. Note that a list of stop words is also used to try and remove them prior to analysis by WordNet.

Thirdly, it can be used to do some intelligent word analysis. To reduce the number of attributes further we would ideally like to capture the idea of a given word in a more generic form and use that instead of the word itself. This should allow us to capture the essence of several words in one. Also if different pages have the same idea behind what they contain then it should allow us to find more trends in the data. For example, if one page contains the words: window, roof, and door, and another web page contains the words chimney, room and brick then we should be able to use Wordnet to find the relationship or root of the tree, the word house. As you can see this would reduce the number of attributes from six to just one. Although this is potentially the most rewarding technique discussed it is also the most risky. If Wordnet finds the wrong relationship between the words we may end up with the wrong root word.

Figure 1 shows the way in which relationships work in WordNet, using the JWNL[1] library it is possible to search for hypernym relationships (Algorithm 1). An example from the diagram might be that brother and sister both have the parent node relative, so the words brother and sister could be replaced with the single word relative.

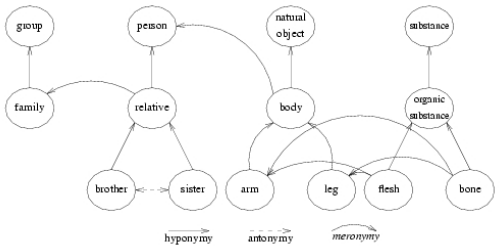

Figure 1: [Miller et al., 1993]

```
WordList = {Words From Text};
ProcessedWordsList = [];
relationShipDepthCutOff = 3;
WHILE (WordListSize > 2)
        BestRelationShip = [];
        CurrentWord = remove WordList element 1
        FOR (i = 0; i<WordListSize)
              Get all relationships between all the
              senses of CurrentWord and WordList
              element i
              Save best as CurrentBest
              IF (BestRelationShipDepth<CurrentBest)
                 Save current relationship as best
        LOOP
        Add the parent node from the
        BestRelationShip to ProcessedWordList
        Remove the other word from the
        BestRelationShip from the WordList
END WHILE
```

Algorithm 1: Finding the best relationships using JWNL

2.4 RSS

RSS is an XML based web content syndication format or Really Simple Syndication. It is used by web sites to show their content in a compact and easily accessible format. It is mainly used by news sites such as BBC[2] to convey information about their latest stories and publications, although it is possible for any site to publish articles in this way.

[1] http://sourceforge.net/projects/jwordnet

[2] http://news.bbc.co.uk/rss/newsonline_uk_edition/front_page/rss091.xml

These RSS feeds contain three basic fields, the title of the document the URL for the document and a short description about it, although many other standards exist with extra fields. These RSS feeds are very attractive to people interested in web classification. They provide concise, accurate and accessible information about many web pages. There are also very useful practical applications for the information contained within the feeds. Users would want to access news, or other information, about a specific subject but would not want to trawl through the entire unclassified web to find them. Google[3] already attempts to classify these news stories based on "interestingness" and subject, but admits itself that "some articles appearing to be out of context" i.e. they are miss-classified. Although RSS feeds will not directly be dealt with due to the current static nature of the Ant Miner algorithm and the dynamic nature of RSS the description and title tags are harvest from a static set of pages. It would be easy to adapt the method described in this paper to get information directly from RSS feeds, and obviously potential future work in adapting the Ant Miner algorithm to a continuous learning scenario, which I believe is viable as Ant Algorithms are by nature good at solving dynamic problems.

3 Ant Colony Algorithms

The pioneer in understand how ant foraging works was Jean-Louis Deneubourg et al. [Jean-Louis Deneubourg et al., 1989]. They proposed that the reason ants are seen creating "highways" to and from their food is because of a chemical pheromone. The way in which this type of intelligent behavior emerges from a group of otherwise unintelligent entities has been labeled as "swarm intelligence". Each ant lays down an amount of pheromone and the other ants are attracted to the strongest scent. This means that the ants tend to converge, which explains the highway pattern they use while foraging for food. The ants do not just converge to a random path, the ants tend to converge to the shortest path, this is because a shorter path is faster to transverse, so if an equal amount of ants follow the long path and the short path, the ants that follow the short path will make more trips to the food and back to the colony. If the ants make more trips when following the shorter path then they will deposit more pheromone over a given distance when compared to the longer path. This is a type of positive feedback and the ants following the longer path will be more likely to change to follow the shorter path, where scent from the pheromone is stronger. The diagram in Figure 1 [Eric Bonabeau et al., 2000] shows a graphical representation of the way in which the shortest path builds up the most pheromone.

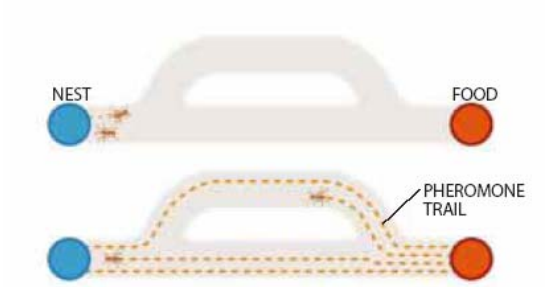

Figure 2: Pheromone trails in natural ant colonies

One problem with the way in which ants find the shortest path naturally involves the addition of a new shorter path after the ants have converged to a longer path. Because pheromone has built up on the older longer path the ants will not take the newer shorter path. This would be a major problem if the algorithm is applied to dynamic problems in computing. However a simple solution was created. If the virtual pheromone evaporates then the longest path will eventually be abandoned for the shortest one. This is because some ants will still take the shorter path by chance (at least in a virtual probabilistic system) and so pheromone will build up and eventually overtake the longer one. This pheromone evaporation does occur in the wild but it is much slower than equivalent computer based implementations.

The way in which ants find the shortest path has been used to create the ant colony paradigm, which is commonly used in problems with large search spaces [Marco Dorigo et al, 1996].

3.1 Ant Miner

The Ant Miner algorithm [Parpinelli et al., 2002] takes the ideas from the Ant Colony paradigm and applies them to the field of data mining. Instead of foraging for food the ants in the ant miner algorithm forage for rules and the path they take is described in terms of attribute value pairs. Ant Miner is described in the pseudo-code in Algorithm 2.

TrainingSet = {all training cases};
DiscoveredRuleList = [ ]; /* rule list is initialized with an empty list */
WHILE (TrainingSetSize > MaxUncoveredCases)

[3] http://news.google.com/intl/en_us/about_google_news.html

```
AntI = 1; /* ant index */
NumConverged = 1; /* convergence test
index */
Initialize all trails with the same amount
of pheromone;
WHILE(AntI < MaxAnts AND
NumConverged < MaxConverged)
        AntI starts with an empty rule
        and incrementally constructs a
        classification rule Rt by adding
        one term at a time to the current
        rule;
        Prune rule Rt;
        Update the pheromone of all
        trails by increasing pheromone in
        the trail followed by AntI
        (proportional to the quality of Rt)
        and decreasing pheromone in the
        other trails (simulating
        pheromone evaporation);
        IF (Rt is equal to Rt – 1)
        /* update convergence test */
        THEN
                NumConverged += 1;
        ELSE
                NumConverged = 1;
        END IF
        AntI = AntI + 1;
    END WHILE
Choose the best rule Rbest among all rules Rt
constructed by all the ants;
Add rule Rbest to DiscoveredRuleList;
TrainingSet = TrainingSet - {set of cases correctly
covered by Rbest};
END WHILE
```

Algorithm 2: [Parpinelli et al., 2002]

Firstly an ant starts off with an empty rule. It then iteratively adds attribute value pair (terms) to the rule, using on a probabilistic function that is based on an amount of virtual pheromone and on a heuristic function that measures the information gain of that particular attribute value pair. More precisely, the larger the amount of pheromone and the larger the information gain for an attribute value pair, the more likely that the attribute value is chosen to be added to the current rule. The ant is considered to have completed its rule when there are less than Min_cases_per_rule cases for that rule. The rule is pruned by removing elements that are unnecessary or make the rule worse (in terms of quality). The pheromone matrix is then updated by increasing the amount of pheromone of the attribute values that occur in the rule the ant has just created. For each of the attribute values, pheromone is increased in proportion to the quality of the rule. This matrix can be considered a discrete landscape on which the ants travel, although it is not spacial in the sense that the values stored in the matrix do not map to coordinates. Once this has finished, the next ant creates a new and separate rule based on the pheromone trails of the previous ants. This means that eventually the ants will converge on a good solution as the pheromone for particular "good" attribute value pairs will be much stronger than the rest of them. There is no part of the algorithm that explicitly makes the pheromone evaporate, although the probabilities stored in the matrix are normalized, which has the side affect of making the attribute value pairs that have not been updated with more pheromone decrease relatively.

The Heuristic function (Equation 2) is a measure of information gain which is based on the entropy (Equation 1 [Weiss and Kulikowski, 1991]) associated with an attribute value pair.

$$\text{info}T_{ij} = -\sum_{w=1}^{k}\left(\frac{freqT_{ij}^{w}}{|T_{ij}|}\right) * \log_2\left(\frac{freqT_{ij}^{w}}{|T_{ij}|}\right)$$

Equation 1: the Entropy.

Where:

k is the number of classes.
$|T_{ij}|$ is the total number of cases which have attribute i equal to value j.
$freqT_{ij}$ is the number of cases that i is equal to j for class w.

The larger the entropy $info_{ij}$, the lower the predictive power for that term ij, as the term ij is more evenly distributed amongst all classes.

There are two caveats in this function, firstly if the value j does not appear for the attribute i in any of the cases the Entropy is set to $infoT_{ij}$ = log2(number of classes) which corresponds to the lowest power of prediction. Also if the term ij only appears in one class then it is set to $infoT_{ij} = 0$, which is the highest predictive power.

$$\varsigma_{ij} = \frac{\log_2(k) - \text{infoT}_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\log_2(k) - \text{infoT}_{ij}}$$

Equation 2: the Heuristic

Where:

a is the total number of attributes.
b_i is the number of possible values in the domain of attribute i (for example, 2 in the case of yes or no).

The amount of normalized pheromone to be deposited initially on every term ij is given by equation 3.

$$\tau_{ij}(t=0) = \frac{1}{\left(\sum_{i}^{a} b_i\right)}$$

Equation 3: the initial amount of pheromone to be deposited.

Equation 4 gives the rule quality for any rule that the ants might generate, the higher the rule quality ($0 \leq Q \leq 1$) the better the rule.

$$Q = \frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN}$$

Equation 4: rule Quality, Q = sensitivity . specificity.

Where:

True Positives: the number of cases covered by the rule that have the class predicted by the rule.
False Positives: the number of cases covered by the rule that have a class different from the class predicted by the rule.
False Negatives: the number of cases that are not covered by the rule but that have the class predicted by the rule.
True Negatives: the number of cases that are not covered by the rule and that do not have the class predicted by the rule.

Equation 5 gives the way in which the pheromone is updated after an ant has generated its rule. For every attribute value pair in the rule the new pheromone ($\tau_{ij}(t+1)$) is equal to the old pheromone ($\tau_{ij}(t)$) plus more pheromone, derived from the quality of the entire pruned rule multiplied by the old pheromone. The pheromone is normalized after updating which has the side effect of evaporating the pheromone of the terms ij not in the last rule.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) * Q, \forall \ i \mid j \in \text{ to the rule}$$

Equation 5: pheromone updating

Equation 6 gives the probability P_{ij} of choosing any given term ij. If the ij has already been used in the current rule then its probability is set to 0. If the insertion of any term ij into the current rule would mean that it covered fewer cases than Min_cases_per_rule then it is not added and has a new probability of 0.

$$P_{ij} = \frac{\tau_{ij}(t)\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i} \tau_{ij}(t)\eta_{ij}, \ \forall i \in I}$$

Equation 6: the probability of an ant choosing a rule.

Where:

a is the total number of attributes.
b_i is the total number of values on i domain (two for yes and no).
I are the attributes i not yet used by the ant.

When the rule generation loop finishes (i.e., the condition AntI < MaxAnts AND NumConverged < MaxConverged is not satisfied), the best rule is selected from the generated rules (based on rule quality) and added to the best rule list. The rule needs a class to be useful, and is given a class based on which class when assigned to the rule gives the best rule quality.

After each rule is generated it is pruned. The quality of the rule is measured after each term ij is removed from it. The rule which when removed increases the rule quality the most is then removed. Note that the class of the rule may be changed after each term ij is removed. This is continued until no term ij can be removed that would increase the quality of the rule.

To test the rules generated they are applied to the test data set in the order they were created. To be considered a correct attempt at classifying the unseen data the antecedent must match the attribute value pairs stored in the test set. The consequent (class) must also match the one predicted by the rule. If there are cases that are not covered by any rule generated then a default rule is used. This default rule simply classifies all the cases left as the majority class in the training set.

In the case that there is only one class left equation 4 breaks down. It returns divide by zero errors as there are no possible true negatives. The Ant Miner algorithm as it is lacks a capacity to deal with these circumstances. In these cases it has been decided to return just the sensitivity.

4 Results

To test the web page classification system fully it is useful to vary the system setup so that it can be ascertained how each factor contributes to the end accuracy. Also, for comparative purposes the performance of the Ant Miner algorithm is tested against the more established C5.0 in certain cases. Comparisons between the C4.5 algorithm and Ant Miner have been carried out before for more general data mining applications [Parpinelli et al., 2002].

4.1 Experimental Setup

A set of 127 web pages in three different classes were harvested from the BBC web site (Education, Technology and Sport). The BBC web site was chosen for analysis because it is arranged in a rigid standard way, which means that all pages have standard tags which can be used for mining. The standard of writing is also high, making it possible to draw relationships between the content and so class of the page and the information saved in the Meta fields. Unfortunately it does not utilize hyperlinks as standard to relate the content of the page to other pages. This makes it impossible to mine information relating to more structural aspects of the web. Also some pages published by the BBC are released in more than one class, so a page that appears in the politics section may also appear in the education section, in these cases the page in question is removed totally from the collected set. To gauge the accuracy of the rules generated five fold cross validation was used. These same sets of training and test data were used to evaluate using the C5 algorithm with Clementine. To make it fair Clementine's own cross validation was not used as to keep the same sets of test and training data in both cases. The C5 algorithm was run with standard settings with no expect noise using a rule set.

For WordNet optimization the Java WordNet Library (JWNL) under BSD license was used. This provides a method to find a relationship between any sense of any word. An algorithm was used to find the best relationships between any two words that were extracted from a given web page. Relationships that were too vague were discarded as they are more likely to be erroneous in the context of their meaning on the page. A maximum relationship depth of 3 was used. All words were stemmed, stop words were removed as well as punctuation.

Although not originally intended, it became necessary to remove attributes with only one occurrence in the data set. These attributes convey only very specific information about the page that they are contained in. If these attributes were left in then the amount of overall attributes increased significantly. This lead to results that were unnecessarily inaccurate and large experimental times.

In the experiments only involving nouns any words that could possibly be classified as nouns by WordNet were used. In both cases words that were not recognized by WordNet were presumed to be proper nouns, these proper nouns were left in as they usually contain important and relevant names.

The "standard" Ant Miner Settings were used:
MaxUncoveredCases = 10
MaxConverged = 20
MaxAnts = 3000
Min_cases_per_rule = 10

Note that MaxConverged was increased from 10 (default value of Ant Miner) to try and stop premature convergence to worse rules. The other parameter values were the same as the default values in [Parpinelli et al., 2002].

4.2 Classification Accuracy

The following tests give the accuracy with the standard deviation, along with the average number of rules and the number of terms in each case. WN is with WordNet analysis, S is just with stemming. Title is where the words are harvested from the title field in the documents, Des is where the words are taken from the description field and U or Union is the union of the two sets. Note that the average number of terms is the average number of terms in all rules for a single fold of cross validation. Also the rule count does *not* include the default rule for Ant Miner of C5.

It might be possible that "black box" type classifiers may be able to produce higher accuracies given this same data set. However one

of the aims of this project was to produce comprehensible knowledge that could be interpreted by the user in the form of rules. It can be argued that such knowledge is not available from "black box" classifiers such as neural networks and so no such comparison is presented here.

Test Setup	Accuracy	No. Rules	No. Terms
WN-Title	59.46±17.02	2.8±0.45	19.57±11.86
WN-Des.	65.54±12.04	3.0±0.0	19.13±13.54
WN-Union	71.03±7.50	3.0±0.0	11.47±7.70
S-Title	55.77±14.83	3.0±0.0	5.13±5.57
S-Des.	72.88±12.87	3.2±0.45	9.25±6.95
S-Union	68.62±9.87	3.0±0.0	11.00±6.71

Table 1: Ant Miner Results – using only nouns

Test Setup	Accuracy	No. Rules	No. Terms
WN-Title	72.33±4.24	3.0±0.0	10.13±7.84
WN-Des.	63.82±12.30	3.0±0.0	12.46±8.18
WN-Union	66.98±9.50	3.0±0.0	11.87±8.63
S-Title	61.86±7.40	3.0±0.0	21.87±10.45
S-Des.	60.90±7.05	3.0±0.0	15.73±10.45
S-Union	66.93±10.42	3.0±0.0	12.53±8.15

Table 2: Ant Miner Results - all words

Test Setup	Alg	Accuracy	No. Rules	No. Terms
WN-U-N	AM	71.03±7.50	3.0±0.0	11.47±7.70
	C5	67.99±4.67	10.6±1.67	23.0±3.39
WN-T-A	AM	72.33±4.24	3.0±0.0	10.13±7.84
	C5	66.42±7.42	11.8±3.19	24.6±3.97
S-Des-N	AM	72.88±12.87	3.2±0.45	9.25±6.95
	C5	72.21±8.09	9.0±2.35	19.4±5.59

Table 3: Comparison between Ant Miner and C5.0

Table 1 shows the accuracies from the different setups. In general the title produces the lowest accuracies with description and then union leading. In general the addition of WordNet optimization increases the accuracy of the tests. The reason the accuracies increase from title to union is because of the amount of information that is available to the Ant Miner algorithm and so to find patterns in increases in each case. Title is the shortest field in general, so conveying the least information with union conveying the most. The WordNet optimization seems to have had the desired effect, however the standard deviation of the accuracy is in general higher than without the optimization. It is probable that this is because using WordNet is risky, it sometimes can be very beneficial but other times it can produce very bad relationships and so accuracies. It should also be noted that the accuracy goes down with just stemming when using the union of the sets. This is likely to be because the number of attributes becomes very large when both description and title are used, this increases the search space massively. This means that it is very hard for the Ant Miner algorithm to produce good rules and so good accuracies. However when just the description with stemming is used a good number of attributes is reached. The number of attributes is not too great as to overwhelm the algorithm but the important information is still left in its raw form. WordNet reduces the number of attributes and so it is possible to utilize more information and still obtain good results.

Table 2 shows the accuracies are in general worse with the inclusion of words that are not nouns. This may be because the amount of attributes becomes overwhelming and the search space too large to find good patterns. The test which had the biggest number of attributes was union with stemming which had 234 attributes. However, it seems that too much information is better than not enough as just using the title only produced 61 attributes and was 5% worse. Also nouns and proper nouns tend to hold the most "information" in a sentence. If words that hold little meaning are included they may make the algorithm produce rules based on them when in fact there is no real pattern just noise. Interestingly the title field with WordNet optimization receives one of the best accuracies out of any test. This may be because although the title contains the least information it also has the least attributes. With WordNet optimization this is the test with the least amount of attributes (60) while still retaining all the data. This makes it easier for the Ant Miner algorithm to find the best rules based on all the information. Also the title tends to be a more compact description in only one sentence, possibly leading to fewer WordNet confusions. The standard deviation of this test is

also the lowest which may mean that WordNet has accurately guessed the relationships more often.

Table 3 shows a comparison between the three best results obtained using the Ant Miner algorithm (with respect to the accuracy) when the same set with the same cross validation was analyzed by the C5 algorithm in Clementine. From top to bottom the first test uses WordNet analysis on the union of the title and description with nouns only. Then WordNet analysis on all the words from the title, and finally the nouns from the description after stemming. The Ant Miner algorithm is comparable or beats the C5.0 algorithm in terms of accuracy. If we compare the average number of terms and average number of rules the Ant Miner and C5.0 algorithm produce we can see there is a big gap. The Ant Miner algorithm produces at worst a third of the number of rules with half the number of terms when compared to C5.0. This means the Ant Miner algorithm excels in terms of comprehensibility in comparison to the C5.0 algorithm. A user would find it much easier to understand and possibly use the knowledge discovered by the Ant Algorithm.

In general it seems that a naïve implementation of WordNet optimization as described in this paper can be beneficial. However the errors and misinterpretations it produces when dealing with more complex and longer sentences can nullify the advantages described. In the scenarios in this paper WordNet optimization is most beneficial when the sentence is short with a simple meaning such as in the title field. I believe that simply stemming the nouns would be more effective on the more complex sentences if the number of attributes did not rise so much, overwhelming the Ant Miner algorithm. This idea that the Ant Algorithm has trouble with a high number of attributes is supported by the C5 comparison, as its accuracies are much closer when the number of attributes rises when the union and description sets are used. Also there is a notable increase in the computational time needed to generate rules from sets with higher numbers of attributes when compared to the C5 algorithm.

5 Conclusion

Attempting to classify information available on the web is a challenging task. During this paper methods and different approaches that may be used to try and achieve this task have been outlined. It has been shown that using the information that might be stored in an RSS feed or in a normal web page it is possible to classify unseen examples with reasonable accuracy within this data set. It has also been shown that a naïve WordNet preprocessing stage can be useful in certain situations. The Ant Miner algorithm has proved again to be a powerful classification tool and to produce accuracies that are at worst comparable to the more established C5.0 algorithm. Not only this but it presents knowledge in a much more compact and comprehensible way.

The web is becoming a more widely recognized way of obtaining up to date content. With this greater recognition so the amount of data available for consumption increases. This is why it is more important now to be able to extract the wanted content from the unwanted content. One way of doing this is via classification. The system that has been described in this paper brings us a step closer to this goal although further research is required. It has already been proven that ant based algorithms are good at problems involving continuous learning in the field of networking [Ruud Schoonderwoerd et al., 1997]. It hopefully would be possible to adapt the Ant Miner algorithm to continuous learning applications as the content available on the web is dynamic by nature.

Another alteration that would be useful for the Ant Miner algorithm for use in the web mining field is the ability to tolerate larger amounts of attributes. This is obviously a very hard problem and methods to navigate around it have been demonstrated, such as deleting attributes that only occur once, using only nouns, WordNet preprocessing and the use of smaller fields such as the title. This is the largest problem facing the system described which is why so much effort has been put into trying to solve it, with only limited success. It seems that the more information available to the classifier the more accuracy that can be achieved, but with this higher amount of information comes the cost of more attributes. Experimentation with a relatively un-optimized system put a practice limit on the number of attributes at about 500. This is a very low number when a much larger more realistic data set is considered and also the amount of extra words available in the main content of a page. The amount of attributes would reach a plateau as the number of web pages increased. If the Ant Miner algorithm can be optimized to cope with this number then it would be more applicable to real world web mining applications.

Bibliography

[Abraham et al., 2003] Ajith Abraham and Vitorino Ramos. Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming, 2003.

[Cuter et al., 1999] M. Cutler, H. Deng, S. S. Maniccam and W. Meng, A New Study Using HTML Structures to Improve Retrieval, 1999.

[Witten et al., 2000] Ian H. Witten, Eibe Frank, Data Mining Practical Machine Learning Tools with Java Implementations, Morgan Kaufmann Publications, 2000.

[Parpinelli et al., 2002] R.S. Parpinelli, H.S. Lopes and A.A. Freitas. Data Mining with an Ant Colony Optimization Algorithm. IEEE Trans. on Evolutionary Computation, special issue on Ant Colony algorithms, 6(4), pp. 321-332, Aug. 2002.

[Eric Bonabeau et al., 2000] Eric Bonabeau, Guy Théraulaz, Swarm Smarts, Scientific American, , pp 73-79, March 2000.

[Jean-Louis Deneubourg et al., 1989] Beckers, R., S. Goss, Jean-Louis Deneubourg and J. M. Pasteels. 1989. Colony size, communication and ant foraging strategy . PSYCHE (CAMBRIDGE) 96(3-4) 1989: 239-256.

[Miller et al., 1993] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to WordNet: An On-line Lexical Database, 1993.

[Chakrabartia et al., 1998]Soumen Chakrabartia, Byron Doma, Prabhakar Raghavana, Sridhar Rajagopalana, David Gibsonb, and Jon KleinbergcAutomatic. Automatic Resource compilation by analyzing hyperlink structure and associated text, 1998.

[Marco Dorigo et al, 1996] Marco Dorigo, Luca Maria Gambardella, Ant colonies for the traveling salesman problem, 1996.

[Ruud Schoonderwoerd et al., 1997] Ruud Schoonderwoerd, Owen Holland, Janet Bruten, Ant-like agents for load balancing in telecommunications networks, 1997.

[Yu, et al., 2002] Hwanjo Yu, Kevin Chen-Chuan Chang, Jiawei Han, Heterogeneous Learner for Web Page Classification, 2002.