Email Classification Using Back Propagation Technique

Taiwo Ayodele, Shikun Zhou, Rinat Khusainov
Department of Electronics and Computer Engineering
University of Portsmouth, United Kingdom

Abstract

This paper proposes a new email classification model using a teaching process of multi-layer neural network to implement back propagation technique. Email has become one of the fastest and most efficient forms of communication. However, the increase of email users with high volume of email messages could lead to un-structured mail boxes, email congestion, email overload, unprioritised email messages, and resulted in the dramatic increase of email classification management tools during the past few years.
Our contributions are: the use of empirical analysis to select an optimum, novel collection of features of a users' email contents that enable the rapid detection of most important words, phrases in emails and a demonstration of the effectiveness of two equal sets of emails training and testing data.

1. Introduction

Email has been an efficient and popular communication mechanism as the number of Internet users' increases. Therefore, email management has become an important and growing problem for individuals and organizations because it is prone to misuse. One of the problems that are most paramount is *disordered email message, congested and un-structured* emails in mail boxes. It may be very hard to find archived email message, search for previous emails with specified contents or features when the mails are not well structured and organized.

Schuff et al [1] stated that "Emails are widely used to synchronize real-time communication, which is inconsistent with its primary goals". Email messages are designed to be sent, accumulate in repository and be periodically collected and read by receipt, which lends itself to the details of a vacation or a meeting's upcoming agenda. Since most people rely on emails for efficiency and effectiveness of communication, mail boxes may become congested. Messages range from static organization knowledge to conversations with such a broad horizon of messages. Users may find it difficult to prioritize and successfully process the contents of new incoming messages. Also it may be difficult to find a previously archived message in the mail box. Kushmerick [2] stated that "the ubiquity of email and its convenience as knowledge management tools make it unlikely that users' behaviour will change as falling bandwidth and disk storage prices further reduce the incentive to steer away from using email as a document storage system". At this stage new effective method for managing information in email, reducing email overloads is developed by classifying emails based on importance of words in the email messages and deriving the closest classes the mail could belong to either: *critical, urgent, very important, important and not important.*

Email classification presents challenges because of large and various number of features in the dataset and large number of mails. Applicability in email datasets with existing classification techniques was limited because the large number of features makes most mails undistinguishable. In many emails datasets, only a small percentage of the total features may be useful in classifying mails, and using all the features may adversely affect performance. The quality of training dataset decides the performance of both the email classification algorithms and feature selection algorithms. An ideal training dataset for each particular category will include all the important terms and their possible distribution in the category. The classification algorithms such as Neural Network (NN), Support Vector Machine (SVM), and Naïve Bayesian (NB) are currently used in various datasets and showing a good classification result as experimented by Young et al [16]. There have been, however, little studies in applying back propagation techniques (BPT) for email classifications. The main disadvantage of BPT is that they require considerable time for parameter selection and training. On the other hand, previous research has shown that back propagation in neural networks (NNs) can achieve very accurate results, that are sometimes more accurate than those of the symbolic classifiers. NNs have been successfully applied in many real world tasks. In this paper we present BACK PROPAGATION TECHNIQUE a NNs-based system for automatic email classification.

2. Email Classification Challenges

The characteristics of emails differ significantly and as a consequence email classification poses certain challenges, not often encountered in text or document classification. Some of the differences and challenges are:

1. Each users' mailbox is different and is constantly increasing. Email contents vary from time to time as new messages are added and old messages are deleted. A classification scheme that can adapt to varying email characteristics is important.
2. Manual classification of emails is based on personal preferences and hence the criteria used may not be as simple as those used for text classification. This distinction needs to be taken into account by any technique proposed for email classification.
3. The information content of emails vary significantly, and other factors, such as the subject field, sender, CC field, BCC field, the person email is addressed to, play an important role in classification. This is in contrast to documents, which are richer in content resulting in easier identification of topics or context.
4. Emails can be classified into folders and could also be classified into subfolders within a folder. The differences in the emails classified to subfolders may be purely semantic (e.g., meeting with travelling folder, conference expenses within travelling folder and many more).

3. Related Work

There is little exploration into the problems of categorising and grouping emails into folders but less work in classification of emails based on the activities of users (based on what the users do). One of the common existing methods used for email classifications is to archived messages into folders with a view of reducing the number of information objects a user must process at any given time. This is a manual classification solution, however, this is an insufficient solution as folder names are not necessarily a true reflections of their content and their creation and maintenance can impose a significant burden on the user [3]. Yukun et al [4] proposed a new email classification model using a linear neural network trained by Perception Learning algorithm (PLA) and a nonlinear neural network trained by Back Propagation Neural Network (BPNN). A Semantic Feature Space (SFS) method was also introduced in this classification model. Moreover, a rule-based system as explained by Schuff et al [1] can provide straight forward way to semi automate email classification and such system require the users to define a set of instructions for the email application to sort incoming messages into folders and order them by importance. The disadvantages of rule-based system are that they are challenging for non technical users because writing the rules require some level of programming experience. Bifrost [5] an email classifier and a prototype email management system [3] avoids this difficulty by letting user define all filtering rules with a simple graphical interface. Terry et al [6] also proposed a new approach by automatically assessing incoming messages and making recommendations before emails reach the user's inbox, so the priority system classifies each messages as of either high or low importance based on its expected utility to the user. While Yukun et al [4] designed a system "that automatically filter spam emails by using the principal component analysis (PCA) and the Self Organized Feature Map (SOFM). In their schema, each email is represented by a series of textual and non-textual features. To reduce the number of textual features, PCA is used to select the most relevant features. Finally the output of the PCA and the non-textual features should be inputted into a well-trained SOFM to classify (*spam or normal*)" and in summary Boone [5] describe Re: Agent system group similar messages based on existing folder structure provided by the user while it learns concept and decision policies for future message classification based on these folders examples.

Aery et al [17] elaborated further about email classification that "depending upon the mechanism used, email classification schemes can be broadly categorized into: i) Rule based classification, ii) Information Retrieval based classification and iii) Machine Learning based classification techniques.

Rule Based Classification: Rule based classification systems use rules to classify emails into folders. William Cohen [4] uses the *RIPPER* learning algorithm to induce 'keyword spotting rules' for email classification. *i-ems* [6] is a rule based classification system that learns rules based only on sender information and keywords. *Ishmail* [7] is another rule-based classifier integrated with the Emacs mail program Rmail.

Information Retrieval Based Classification: Segal and Kephart [13] have used the TF-IDF classifier as the means for classification in *SwiftFile*, implemented as an add-on to Lotus Notes. It predicts three likely destination folders for every incoming email. The TF-IDF classifier performs well in the absence of a large training set and also when the amount of training data increases, adding to the heterogeneity of a folder.

Machine Learning Based Classification: The *iFile* system by Rennie [9] uses the naive Bayes approach for effective training, providing good classification accuracy, and for performing iterative

learning. *Re:Agent* by Boone [2] first uses the TF-IDF measure to extract useful features from emails and then predicts the actions to be performed using the trained data and a set of keywords. Mail Agent Interface (*Magi*) by Payne and Edwards [10] uses the symbolic rule induction system *CN2* [3] to induce a user-profile from observations of user interactions".

4. Email Classification

Classification of text in an email message is an example of supervised learning that seeks to build a probabilistic model of a function that maps emails to classes. In supervised learning of text in email messages, where an entire email dataset represents one example of emails to be classified, a learning algorithm is presented with a set of already classified, or labelled, examples. This set is called the *training* set. A number of classified emails from the training set are removed prior to model building to be used for testing the model's performance. This set is known as the *testing* set. To better measure the classification accuracy of our model, several models are built from different partitions of the examples to training and testing sets. The classification error is then averaged over each model. This process is called n-times cross validation where "n" is the number of times the example set is partitioned. We produce 1000 models for evaluation using this process and we obtained 1000-times cross validation. Now our model has been constructed, it was used to predict the classification of future email messages. The accuracy of our models are largely dependent on:

- The performance of our back propagation algorithm
- The important word selection using information retrieval
- The "representativeness" of the training data with respect to newly acquired email data to be classified.

The more representative, the training data, the better the performance. A larger number of training examples is often better, because a larger sample is likely to be more reflective of the actual distribution of the data as a whole.

4.1. Email Message Transformation

Classification machine learning algorithms operate on numerical quantities as inputs. A labelled example is often a vector of numeric attribute values with one or more attached labels. For some methods, like naïve Bayesian, integer counts of the attribute values are all that is required. Attributes in these cases can be nominal types but they are still ultimately converted to numeric counts before processing. The conversion of text records to vectors of numeric attributes is a multi-staged process of feature construction which employs several key transformations. There are many approaches to this process – one of the most common is to treat each email content as a "bag of words." Each unique word constitutes an attribute (a position in our example vector). The number of occurrences of a word in a n email (frequency of occurrence) is the attribute's value for that email message. Email messages are therefore represented as vectors of numeric attributes where each attribute value is the frequency of occurrence of a distinct term. This set of email message vectors is often referred to as a vector space. Algorithms that operate on such representations are said to be using vector space models of the data. Furthermore, some types of words, or series of words, may be preferable for learning. For example, often nouns or noun phrases are preferred. Part-of-speech identification algorithms and lexical/semantic dictionaries are typically used to provide additional information about terms. Also, very common words like "and" and "the" are often filtered out using stop word to improve performance. All of these transformations are performed before any learning takes place. The number of steps involved in this pre-processing can be quite numerous and often constitutes the bulk of the overall model building process [9].

4.2. Applications of Message Classification

Our email message classification has many direct applications such as:

- Classification of email according to the content to (Critical, Urgent, very important, Important, and Not important) or
- Classification of streams of emails, with various important words, phrases, to identify particular items of interest.

We implemented supervised learning for the email word extractions and also support higher-level objectives like:

- Information extraction: the process of extracting bits of specific information from unstructured email messages
- Classification methods: These are often used to identify the parts of message content that qualify for extraction. For example, extracting the date, time, location, important words-meeting, interview, surgery appointment, flight booking, conference booking and venue, credit card deadline etc.

5. New Neural Network Approach

The approach that we implemented for our email classification into categories is a *supervised learning*. Our sample categories are: Critical, *Urgent, Very important, and Others*. Our solution

is based on heuristic technique and on the facts that if an email is about:

- Loss of life, vital incident, accident, etc, then our classifier should indicate that such a mail is *critical*
- Meeting deadlines, reminder of vital appointments, interview appointment, visa embassy appointment. In summary, if such a mail is about *time* and *deadline*, then such a mail could be categorised as *Urgent*
- Conference invites, paper presentations, reminder of events, meeting reminder, tasks to perform daily etc such a mail could be categorised as *very important*
- If there is not timing and deadline in such a mail, if it is not about loss of life, illness, reminder of meeting, messages from friend and family, and the likes, then such a mail will be categorised as *others*

We embedded a technique by Ramos [6] who stated that "Term Frequency Inverse Document Frequency (TF-IDF) to determine what words in a corpus of email messages might be more favourable to use in a query. As the term implies, TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. Words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that if that words were to appear in a query, the document could be of interest to the user". We provide evidence that this simple algorithm efficiently categorizes relevant words that can enhance query retrieval in email messages. With our proposed solution, we define informally, that query of word retrieval can be described as the task of searching a collection of email dataset for specific instances of the content.

5.1. Our Method

We implement a neural network [14] based system for automated email classification into user defined "word classes". This is an experiment based on the content of subject field in a mail box and email message content. The classes are words with meaning (Critical, Urgent, Very important and others). The experiments show that it is more accurate than several other techniques. We also investigate the effects of various feature selection, weighting and normalization methods, and also the use of back propagation for the email datasets.

We implement search for collections of important words in email corpus [10] from Enron, our refined problem then becomes the task of searching this corpus for email datasets that the query retrieval system considers relevant to what the mail user entered as the query. We have a set of emails E, with the user entering a query $q = w_1, w_2, ...w_n$, for a sequence of words w_i. Then we wish to return a subset E^* of E such that for each $e \in E^*$, we maximize the following probability:

$$P(e \mid q, E) \qquad (1)$$

[11]. As the above notation suggests, numerous approaches to this problem involve probability and statistics, while others propose vector based models to enhance the retrieval.

Given an email collection E, a word w, and an individual email $e \in E$, we calculate

$$we = fw, e * \log(|E|/fw, E) \qquad (2)$$

Where fw, e equals the number of times w appears in e, $|E|$ is the size of the corpus, and fw, E equals the number of email messages in which w appears in E. There are a few different situations that can occur here for each word, depending on the values of fw, e, $|E|$, and fw, E, the most prominent of which was well examined. Assume that $|E| \sim fw, E$, i.e. the size of the corpus is approximately equal to the frequency of w over E. If $1 < \log(|E|/ fw, E) < c$ for some very small constant c, then we will be smaller than fw, e but still positive. This implies that w is relatively common over the entire corpus but still holds some importance throughout E. For example, this could be the case if TF-IDF would examine the word 'critical illnesses over the doctor's email conversations. This is more relevant to surgeons as well as the family of individual; this result would be brought to the attention of our classifier as *critical* because such a mail is about life and death. This is also the case for extremely common words such as articles, pronouns, and prepositions, which by themselves hold no relevant meaning in a query (unless the user explicitly wants email messages containing such common words). Such common words thus receive a very low TF-IDF score, rendering them essentially negligible in the search. This is where our search for important words in email messages is implemented and seems to be very effective in email messages.

5.2. The learning process

Our learning technique is an associative learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the neural network (self-supervised). And this learning process described above is implemented in this work. Figure 1 shows the sample learning process.

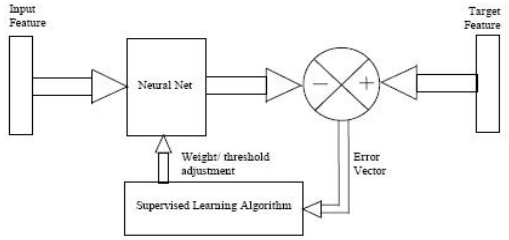

Figure 1. A sample Supervised Learning Process

We implemented neural network (NN) which has the ability to learn by example with back propagation techniques. Back propagation according to Habra [15] is a popular type of network that can be trained to recognize different patterns including images, signals, and text. Once the user has created the data, this data is used to run by a multilayer neural network. The inputs of the NN are the word importance in email messages and the output is the importance.

Each word in an email represents an input node in the neural network. Therefore the number of neurons in the first layer equals to the number of words in the input vector. For the output layer, there are four nodes. The nodes are filled binary. If the email is urgent, the first output node gets value 1, and the rest 0, if the email is very important the second output node gets value 1 and the rest 0, and so on. It is required to have input-output data and the data is found from the email messages (words in the email content). With the word extraction from email messages, we based our approach on information retrieval method as described by Ramos [6]. The most important words are selected and using formula tf-idf (a proven approach in IR). Most important words in an email message get bigger values. **Output (Critical, Urgent, Very important, Others)**: In order to obtain the output, we created the NN data model in Figure 2:

The purpose of this module is to label each mail data as one of the classes. This is how the output is formed. If the user does not do this step properly, the data will be corrupted and the learning algorithm will not learn anything. All procedures must be followed in order to obtain accurate categories.

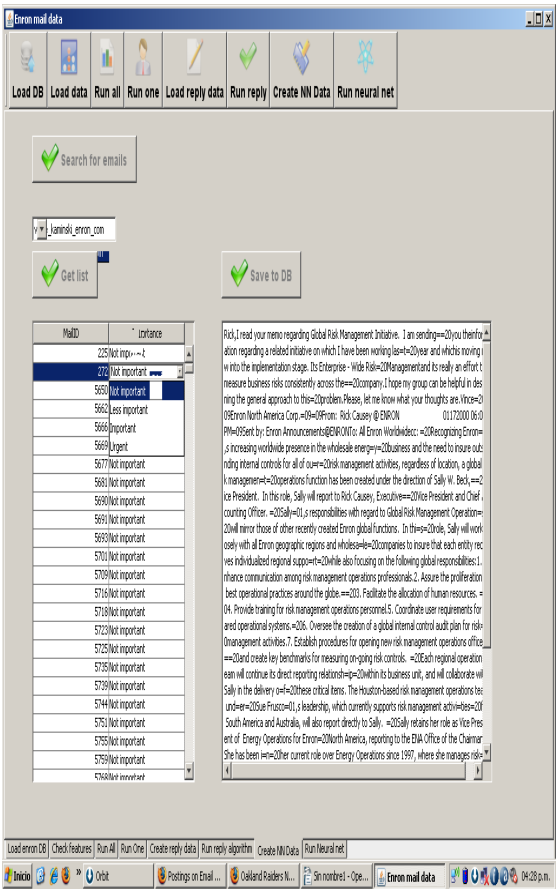

Figure 2. New Classifier with options (Urgent, Very Important, etc.)

6. Evaluations Performance

Over 10000 email conversations from the Enron email dataset [18] as the test bed and BPT algorithm was implemented several times on the email datasets. To evaluate the performance of our email classifier, we measure the number of good classification- correct and bad classifications – incorrect in comparison to human judged classifications. The rate of success is the average of correct classification over all data. All emails from a given user are separated in two equal sets training and test. The training data is used to train the neural network (NN) and our NN knows the patterns that make emails to be categorised as *critical*, *urgent*, *very important, Others* etc. Our learning solution shows that NN learns when the training error (the errors that commit the NN while learning) is decreasing.

In order to check that the NN learned, the test data is used, because the test data was not used to train, it is an unbiased estimate of the performance. In this step, the test data is presented and the results are shown in table 1. If the category equals the right value, it is a correct category that such a mail

is categorised to otherwise, it is an incorrect category. The numbers shown as accuracy are the percentage of correctly categorised mails and incorrect categorised mails over all test data as shown in Table 1:

Table 1. Email classification BPT results

No of Runs for Neural Network	Correct Classified Mails- Hits	Incorrect Classified Mails- Misses	Total Numbers of Mails	% of Successful Classified Mails
1	223	519	742	30.05%
2	451	209	660	68.33%
3	443	134	577	76.77%
4	403	92	495	81.41%
5	361	51	412	87.62%
6	295	35	330	89.39%
7	222	26	248	89.51%
8	149	16	165	90.30%
9	80	3	83	96.38%
10	82	1	83	98.80%

The Table 1 shows the experimental results of our proposed neural network solution for email classification into meaningful words. We calculate:

$$Email\_Categories = \frac{New\_NN\_Correct}{Total\_Num\_Mail}$$

We are able to achieve 98% success with the email categories and when our new neural network algorithm is compared with human participants our algorithm's performance seems to work well and better than the existing approach. Figure 3 shows more graphical results of our experiments.

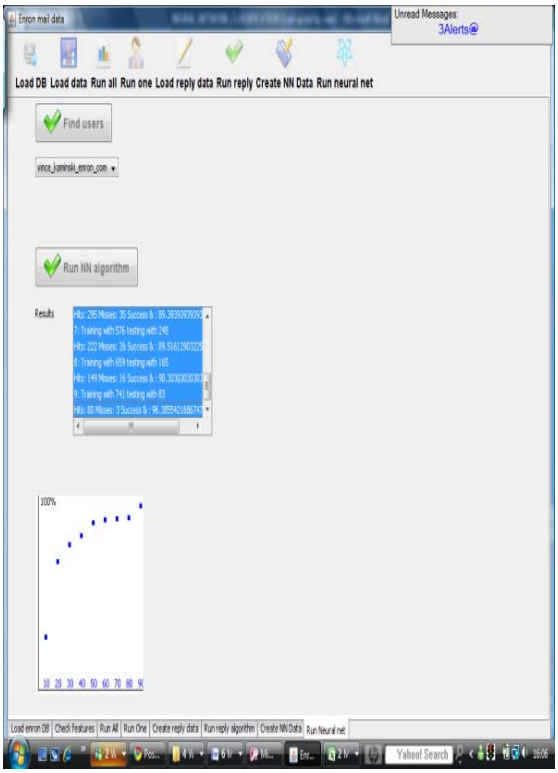

Figure 3. Our new neural network experiment

Figure 3 shows how each email is being trained and how the test data are analysed. It also shows the output results of the percentage accuracy of the email classification categories. It is deduced from figure 3 that after experimenting over 10,0000 of email messages from thousands of users from IT, Banking, Academia, Researchers, students, private and public businesses, health area, constructions, tourism and transportation sector and many more, the training classifier was trained to analyse email messages from the aforementioned sectors which covers most areas where is the most used communication tools for communication. Figure 3 shows the novel training classifier becomes more intelligent and learn faster as error decreases and was able to categorise email messages into meaningful words: *Critical, Urgent, Very important, and Others* based on training and examples as it has been thought.

Further testing and evaluations were implemented on thousands of email messages to determine the categories they belong to using back propagation technique (BPT). Table 2 shows that the more email messages the user have the harder it becomes to determine the categories as the percentage of accuracy decreases if more email messages are to be grouped in bulk.

Table 2. Comparisons of Human-Judged Classification and BPT Classification

Email Counter.	Human Judged Classifications	Back Propagation Technique Classifications	% of Classification Accuracy
1	1000	986	98.6%
2	2000	1884	94.4%
3	3000	2790	93.0%
4	4000	3699	92.4%
5	5000	4600	92.0%
6	6000	5461	91.0%
7	7000	6358	90.8%
8	8000	7146	89.3%
9	9000	8005	88.9%
10	10000	8710	87.1%

In order to compare different approaches of email classifications, a gold standard is needed. In practice, for comparing extractive classifier, we need to know what sentences a human classifier would extract as most important from a target email corpus to determine the categories of email messages. Notice that having such a gold standard may also allow us to verify our assumptions on what information and algorithms our BPT should rely on.

Table 2 indicated that if BPT is applied to a small numbers of incoming email messages into the

mail box, it performs very well and BPT can achieve 98% accuracy in comparison to human-judged classifications. But when BPT is applied to huge numbers of emails above 6000 messages at the same time, the propose technique's accuracy starts decreasing and this is as a result of the back propagation's inability to performs thorough error check as compared to when it has less mails to check and calculate better. With over 10000 email messages, we are able to achieve 87% classification accuracy with the BPT.

7. Conclusion

In this paper, we study how to generate accurate email categories. We analyse the characters of emails and study the email conversation structure, which we argue have not been sufficiently investigated in previous research on email classification using back propagation technique. We build a novel structure: Our classification is based on heuristic technique with the used of Term Frequency Inverse Document Frequency (TF-IDF) to determine what words in a corpus of email messages might be more favourable to use in a query, We also implement a neural network based system for automated email classification into user defined "word classes" and our BPT implemented was able to learn technique in an associative learning approach, in which the network is trained by providing it with input and matching output patterns.

We have shown that neural networks using back propagation technique can be successfully used for semi-automated email classification into meaningful words. The back propagation is based on learning by example and outperforms several other algorithms in terms of classification performance. We explored the effects of various features selection, hidden weight calculations techniques. Frequency and tf-idf weighting with mailbox level normalization which produced the best results in email classification. Further experiments will be carried out in the future.

8. References

[1] Schuff, D., O. Turetke, D. Croson, F 2007, 'Managing Email Overload: Solutions and Future Challenges', *IEEE Computer Society, vol. 40, No. 2, pp.* 31-36.

[2] Kushmerick, N., Lau, T. 2005, '*Automated Email Activity Management: An Unsupervised learning Approach', Proceedings of 10th International Conference on Intelligent User Interfaces*, ACM Press, pp. 67-74.

[3] Helfman, J., Isbell, C. 1995, 'Ishmail: Immediate Identification of Important Information', AT&T Labs.

[4] Boone, G. 1998, 'Concept Features in Re: Agent, An Intelligent Email Agent', *Proceedings of 2nd International Conference on autonomous agents,* ACM Press, pp.141-148.

[5] Balter, O., Sidner, C. *Bifrost Inbox Organizer: Giving Users Control over the Inbox*. in *In Proceedings of the Second Nordic Conference on Human-Computer interaction*. 2002. Aarhus, Denmark: ACM Press.

[6] Ramos, J. (2002). *Using TF-IDF to Determine Word Relevance in Document Queries,* Department of Computer Science, Rutgers University, Piscataway, NJ, 08855.

[7] Yun, F.Y., Cheng, H.L., Wei, S. (2008). *Email Classification Using Semantic Feature Space, Proceedings* of the 2008 International Conference on Advanced Language Processing and Web Information Technology, IEEE Computer Society Washington, DC, USA, pp.32-37.

[8] Yukun, C., Xiaofeng, L., Yunfeng, L. (2007). *An E-mail Filtering Approach Using Neural Network*, Springer Berlin, pp. 688-694.

[9] United States. The Board of Trustees of the University of Illinois. (2003). *D2K™ Data to Knowledge™ Text Mining: Email Classification.*

[10] Bryan Klimt and Yiming Yang. (2004). The Enron corpus: A new dataset for email classification research. In European Conference on Machine Learning.

[11] Berger, A., Lafferty, J. (1999). Information Retrieval as Statistical Translation. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval* (SIGIR.99), 222-229.

[12] Clark, J., Koprinska, I, Poon, J. (2003). *A Neural Network Based Approach to Automated E-mail Classification.* In the proceedings of Web intelligence (WI 2003). IEEE/WIC International Conference, pp.702-705.

[13] Bousquet O., Boucheron, S., Lugosi, G. (2004). Introduction *to statistical learning theory*. In Advanced Lectures in Machine Learning, Springer, pp.169-207

[14] Artificial Neural Networks. 2008. *Neural Networks*. [Online] Artificial Intelligence Technologies tutorial. Available at: http://www.learnartificialneuralnetworks.com/#Intro [Accessed 10 May 2009].

[15] Habra, A., 2005. *Neural Networks - An Introduction.* [Online] Technology Exponent. Available at: http://www.tek271.com/?about=docs/neuralNet/IntoToNeuralNets.html [Accessed 20 April 2009].

[16] Youn, S. a. (2006). A Comparative Study for Email Classification. *JOURNAL OF SOFTWARE , 2* (3), 1-13.

[17] Aery, M. a. (2005). eMailSift: Email Classification Based on Structure and Content. *In Proceedings of the Fifth IEEE international Conference on Data Mining* (pp. 18-25). Washington, DC: IEEE Computer Society.

