Expert Systems with Applications 37 (2010) 18–23

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Automatic thesaurus construction for spam filtering using revised back propagation neural network

Hao Xu [a], Bo Yu [b,*]

[a] College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China
[b] Lab of Knowledge Management and Data Analysis, School of Management and Economics, Beijing Institute of Technology, Beijing 100081, P.R. China

ARTICLE INFO

Keywords:
Spam filtering
Back propagation neural network
Revised back propagation neural network
Automatic thesaurus construction

ABSTRACT

Email has become one of the fastest and most economical forms of communication. Email is also one of the most ubiquitous and pervasive applications used on a daily basis by millions of people worldwide. However, the increase in email users has resulted in a dramatic increase in spam emails during the past few years. This paper proposes a new spam filtering system using revised back propagation (RBP) neural network and automatic thesaurus construction. The conventional back propagation (BP) neural network has slow learning speed and is prone to trap into a local minimum, so it will lead to poor performance and efficiency. The authors present in this paper the RBP neural network to overcome the limitations of the conventional BP neural network. A well constructed thesaurus has been recognized as a valuable tool in the effective operation of text classification, it can also overcome the problems in keyword-based spam filters which ignore the relationship between words. The authors conduct the experiments on Ling-Spam corpus. Experimental results show that the proposed spam filtering system is able to achieve higher performance, especially for the combination of RBP neural network and automatic thesaurus construction.

1. Introduction

In a relatively short timeframe, Internet has become irrevocably and deeply entrenched in our modern society primarily due to the power of its communication substrate linking people and organizations around the globe. Email has become one of the most reliable and economical forms of communication as the number of Internet users has increased, and individuals and organizations rely more and more on the emails to communicate and share information and knowledge. The number of emails has been increasing all the time, however, this explosive growth comes with a variety of problems. The number of unsolicited commercial emails or spam emails has been increasing dramatically over the last few years, and a study has estimated that over 70% of the business emails are spam (Aladdin knowledge systems). These spam emails not only consume users' time and energy in identifying and removing the undesired messages, but also cause many problems such as taking up the limited mailbox space, wasting network bandwidth and engulfing important personal emails; furthermore, they can cause serious damage to personal computers in the form of computer virus. The growing volume of spam emails has resulted in the necessity for more accurate and efficient spam filtering system.

Many spam filtering approaches have been proposed, such as those presented in Blanzieri and Bryl (2008), Fdez-Riverola, Iglesias, Díaz, Méndez, and Corchado (2007) and Jiang (2006). The commonly used machine learning-based techniques include decision trees (Crawford, 2001), memory-based approaches (Sakkis et al., 2003), support vector machine (SVM) (Drucker, Wu, & Vapnik, 1999; Wang, Yu, & Liu, 2005), naive bayes (Androutsopoulos, Koutsias, & Chandrinos, 2000), *k*-nearest neighbor (Crawford, Koprinska, & Patrick, 2004), and artificial immune systems (Bezerra et al., 2006). Among the others, neural network for spam filtering is also studied. The main disadvantage of neural network is that it requires considerable time for parameter selection and network training. On the other hand, previous researches have shown that neural network can achieve very accurate results, sometimes more accurate than those of the symbolic classifiers (Clark, Koprinska, & Poon, 2003). LINGER is a neural network-based system for automatic email classification, the experimental results show that the neural network-based filter achieves better accuracy in the training phase but has unstable portability across different corpora (Clark et al., 2003). Chen et al. compared four algorithms, naïve bayes, decision trees, neural network and boosting, and drew a conclusion that neural network algorithm has higher performance (Chen, Chen, & Ming, 2003). Cárpinteiro et al. proposed a multilayer perceptron model to classify spam emails and non-spam emails, the experiments show that a good neural network-based

* Corresponding author. Tel.: +86 10 6894 9443.
E-mail address: bitboyu@gmail.com (B. Yu).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.02.059

spam filter needs to elaborate on its architecture and learning strategy (Carpinteiro et al., 2006). In Wang, Jones, and Pan (2006), two linear classifiers, perceptron and winnow, are integrated for spam filtering. A hybrid method that combines neural network and genetic algorithms for feature selection is presented for robust detection of spam (Gavrilis, Tsoulos, & Dermatas, 2006). Cui et al. proposed a model based on the neural network to classify personal emails, and the use of principal component analysis (PCA) as a preprocessor of neural network to reduce the data in terms of both dimensionality and size (Cui, Mondal, Shen, Cong, & Tan, 2005). These studies show that neural network can be successfully used for automated email classification and spam filtering. Back propagation (BP) neural network is the most popular among all the neural network applications. It has the advantages of yielding high classification accuracy. However, practical applications are difficult to be satisfied because of the problems of slow learning and the likelihood of being trapped into a local minimum especially when the size of the network is large. These problems are due to the fact that the learning of BP neural network is mechanical and elementary. It does not have the advanced intelligent characteristic to generalize the previous training experience. Many researches have been done to overcome these problems, especially the local convergence (Ma & Ji, 1998; Ooyen & Nienhuis, 1992; Yu, Chen, & Cheng, 1993).

From the perspective of a computer science researcher, especially a machine learning researcher, spam filtering can be regarded as a binary text classification problem. The classifier must distinguish between legitimate emails and spam emails. However, the sparse and noisy feature space makes it more difficult because an email contains more informal words and sentences as well as more spelling errors. Previous works have attempted to increase classification accuracy using efficient feature selection methods (Kolcz, Chowdhury, & Alspector, 2004) or using more valid features for spam filtering (Crawford et al., 2004). More recently, instance-based reasoning system (Riverola, Iglesias, Díaz, Méndez, & Corchado, 2007) and feature free case-based reasoning system (Delany & Bridge, 2007) have been used for spam filtering and have been proved to have achieved good performance.

In order to perform spam filtering, an important issue is how to represent each email. As in text classification, a typical method is to create document vectors from terms frequently occurring in the documents, known as the bag of words (BOW). However, it ignores the conceptual similarity between terms that can lead to poor performance. Ambiguity in the meaning of terms (for example, the word "car" and "automobile" have the same meaning, the word "bark" has a different meaning when it is used as a verb or as a noun) can prevent the classifier from choosing the categories deterministically, and will directly decrease the classification accuracy. Thesaurus was a promising avenue to address the problem using BOW. Thesaurus is originally proposed as an information-retrieval method for query expansion, it can also work well when it is applied to text classification (Bang, Yang, & Yang, 2006). An automatic thesaurus consists of a set of weighted term associations which are based on the hypothesis that terms have relationship if they co-occur often in the documents (Xu & Croft, 1996). Then the associated terms are used to query expansion. There are many previous works that have researched on automatic thesaurus generation for information retrieval, Qiu and Frei worked on a term by term similarity matrix based on how the terms of the collection are indexed (Qiu & Frei, 1993). Ángel et al. have developed a method using similarity thesauri for Spanish documents (Ángel, Carlos, José, & Emilio, 2005). Pérez-Agüera and Araujo have shown how to use handmade thesauri combining with the statistical method to automatically generate a new thesaurus for a particular knowledge domain (Pérez-Agüera & Araujo, 2006). These approaches obtain promising results when applied to improve information-retrieval process.

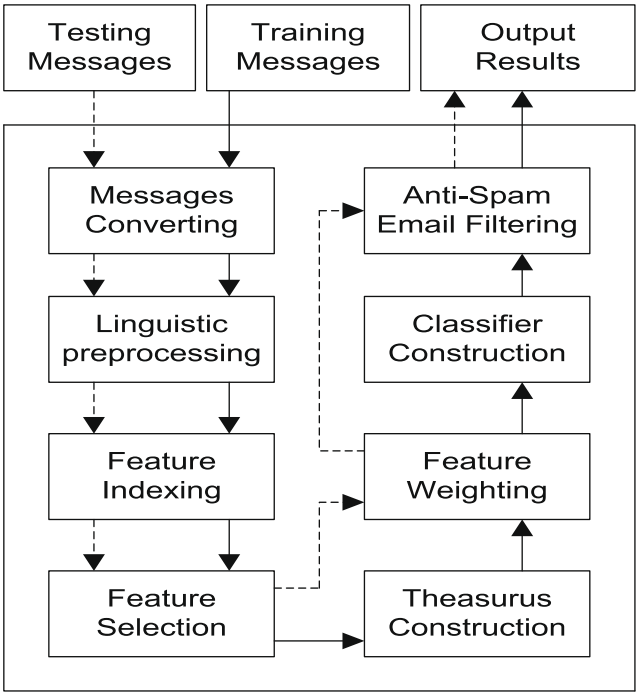

Fig. 1. The architecture of the spam filtering system.

The overview of the system introduced by this paper is shown in Fig. 1, the arrow with dashed line represents the data flow in the testing process, and the arrow with solid line represents the data flow in the training process. The spam filtering system introduced by this paper comprises four key components: data preprocessing, thesaurus construction, classifier construction, and spam filtering. The details of the components are described in the following sections. The remainder of the paper is organized as follows. The conventional BP neural network learning algorithm and the revised back propagation (RBP) neural network learning algorithm are described in Section 2. The process of generating the thesaurus for our spam filtering system is explained in Section 3 . The experiments and results are detailed in Section 4. Finally, the conclusions and future works are given in Section 5.

2. Neural network learning algorithms

2.1. BP neural network learning algorithm

BP neural network is a generalization of the delta rule used for training multi-layer feed forward neural network with nonlinear units. It is simply a gradient descent method designed to minimize the total error or mean error of the output computed by the network. The standard architecture of a BP neural network consists of an input layer, an output layer, and one or more than one hidden layer between them. Each layer consists of several nodes of neurons, and there are no connections between neurons in the same layer or in a previous layer. The structure of a typical three-layer back propagation neural network is shown in Fig. 2.

The network functions are as follows: Each node i in the input layer receives a signal x_i as the network's input, multiplied by a weight value between the input layer and the hidden layer. Each node j in the hidden layer receives the signal $In(j)$ according to the following equation:

$$In(j) = b_j + \sum_{i=1}^{n} x_i w_{ij} \quad (1)$$

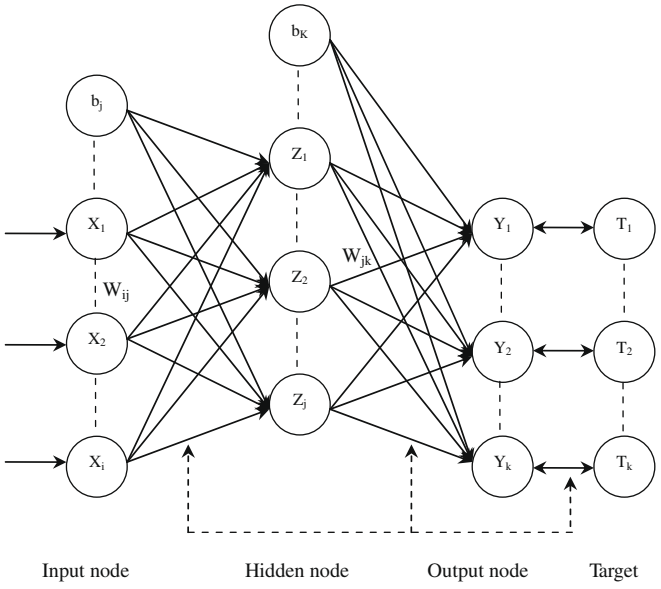

Fig. 2. Typical three layer back propagation neural network.

where w_{ij} is the weight of the connection from the ith neuron in the previous layer to the neuron j. Then the signal $In(j)$ is passed through the bipolar sigmoid activation function

$$f(x) = \frac{2}{1+\exp(-x)} - 1 \tag{2}$$

The output of the activation function $f(In(j))$ then broadcasts all the neurons to the output layer

$$y_k = b_k + \sum_{j=1}^{m} w_{jk} f(In(j)) \tag{3}$$

where b_j and b_k are the bias in the hidden layer and the output layer, respectively.

The output value will be compared with the target; in this paper, we use the mean absolute error as the error function:

$$E_m = \frac{1}{2n} \sum_{k} \sqrt{(T_k - Y_k)^2} \tag{4}$$

where n is the number of training patterns, Y_k and T_k are the output value and the target value, respectively.

The gradient descent method searches for the global optimum of the network weights, and the partial derivatives $\partial E/\partial w$ are computed for each weight in the network. Then the weight is adjusted according to the following equation:

$$w(t+1) = w(t) - \eta \partial E(t)/\partial w(t) \tag{5}$$

where t is the number of epochs, and η is the learning rate.

2.2. The main problems of conventional BP neural network

As mentioned above, the conventional BP neural network learning algorithm has two main problems, viz. its slow training speed and likelihood of entering into a local minimum. The method presented by the authors is designed to overcome these problems.

2.2.1. Slow training speed

In the beginning, the learning process proceeds very quickly, and can make rapid progress, however it slows down in the later stages (Wu, Feng, Li, & Xu, 2005). There are two commonly used methods to improve the speed of training for BP neural network. *(a) Introduce momentum into the network.* Convergence is sometimes faster if a momentum term is added to the weight update formulae. *(b) Using the adaptive learning rate to adjust the learning rate.* The role of the adaptive learning rate is to allow each weight to have its own learning rate, and to let the learning rates vary with time as training progresses. The two methods can accelerate the convergence of the BP neural network learning algorithm to some extent, but they cannot solve the other problems associated with the BP neural network, especially when the size of the network is large.

2.2.2. Local minimum

When training a BP neural network, it is easy to enter into a local minimum, and usually the Genetic Algorithm and simulated annealing algorithms are used to avoid this problem. These algorithms can prevent the BP neural network from entering into a local minimum, but they cannot ensure that the network will not enter into a global minimum, and they are even slower than the conventional BP neural network.

2.3. RBP neural network learning algorithm

The BP neural network with momentum and adaptive learning rate has shown some improvement, but it is not very significant, and it will fluctuate in the long term. The RBP neural network can find the problems during the training of the BP neural network, and the authors have given the formulae to overcome these problems.

During the learning process, neurons face two kinds of problems: neuron overcharge and neuron fatigue. The authors called them morbidity neurons. If we avoid the appearance of morbidity neurons during the learning phase or rectify the problem on time, then the networks can train and evolve effectively.

Firstly, the authors would like to introduce some definitions.

Definition 1 (*Learning phase*). Choosing N iterations (or epochs) as a period, during this period we record some important data and calculate the effect in each learning process. The learning effect will indicate the direction of the network globally and will get rid of the blindness of mechanical and repeated single learning. The next learning phase will adjust the learning model based on the evaluation effect in the previous learning phase. We call this period as learning phase. In our experiments, each learning period includes 50 epochs.

Definition 2 (*Neuron overcharged*). If the input value of the neuron is very big or very small, it will cause the output value to approach to -1 or 1, and will cause the back propagation error to approach to 0. We refer to such a neuron as being overcharged. That is, for the activation function (bipolar sigmoid function)

$$f(net_j + \theta_j) = \frac{2}{1+e^{-\lambda(net_j+\theta)}} - 1 \tag{6}$$

if

$$f(net_j + \theta_j) \geqslant 0.9 \text{ or } f(net_j + \theta_j) \leqslant -0.9 \tag{7}$$

$f(x)$ is the bipolar activation function which values are range from -1 to 1, when Eq. (7) holds, we refer to the neuron j as being overcharged.

Definition 3 (*Neuron fatigue*). If a certain neuron always receives a similar stimulation, then its response to this stimulation will be very similar, so it is difficult to distinguish different stimulations by its response. We refer to such a neuron as being fatigued. That is, during one learning phase, when the neuron j obeys

$$\underset{k}{MAX} f\left(net_j^k + \theta_j^k\right) - \underset{k}{MIN} f\left(net_j^k + \theta_j^k\right) <= 0.2 \tag{8}$$

k is the number of epochs in one learning phase, when Eq. (8) holds, we refer to the neuron j as being fatigued.

It is noted that the overcharged neuron is originated from the derivation of the activation function. In the conventional activation function, λ is 1 or other constants, whereas in our model, λ is an adjustable value. Plagianakos and Vrahatis tried to train the network with threshold activation function by changing the value of λ in their paper. Actually, different combinations of λ correspond to different learning models (Plagianakos & Vrahatis, 2000).

The following formulae are used to rectify the morbidity neurons. For the neuron overcharge problem, we want to limit the maximum and minimum output values to the normal range. In our experiments, the range is from −0.9 to 0.9. In our study, the morbidity neurons were rectified in each learning phase after their evaluation. $MAX_k f\left(net_j^k + \theta_j^k\right)$ and $MIN_k f\left(net_j^k + \theta_j^k\right)$ are the maximum and minimum input values which include the bias θ_j^k during one learning phase, λ is the variable that controls the slope of the activation value. So, the morbidity neuron of overcharged neuron can be rectified by the formula

$$\lambda_j = -\frac{In\left(\frac{2}{1.9} - 1\right)}{MAX_k f\left(net_j^k + \theta_j^k\right) - MIN_k f\left(net_j^k + \theta_j^k\right)} \tag{9}$$

For the neuron fatigue problem, we want to normalize the maximum and minimum input values in the previous phase in order to make them symmetric with respect to the origin. So, the morbidity neuron of neuron fatigue can be rectified by the formula

$$\theta_j = \theta_j - \frac{MAX_k f\left(net_j^k + \theta_j^k\right) + MIN_k f\left(net_j^k + \theta_j^k\right)}{2} \tag{10}$$

3. Automatic thesaurus construction

3.1. Vector space model

The vector space model is implemented by creating the term document matrix and a vector of email documents. In order to create the set of initial feature vectors used for representing the training emails, we need to transform each email into a feature vector. Let the list of relevant terms be numerated from 1 to n. The feature vector of the emails can be represented as:

$$d_k = \langle w_{k,1}, \ldots, w_{k,i}, \ldots, w_{k,n} \rangle \tag{11}$$

where $w_{k,i}$ is the term weight of the ith indexing term in kth email document.

3.2. Automatic thesaurus construction

The basic idea of thesaurus is to calculate the similarities between two terms on the basis of their co-occurrence in a document corpus. This approach is based on the association hypothesis that related terms tend to co-occur in documents in the corpus. As a result, this type of automatic thesaurus consists of a set of weighted term associations. For example, in a certain corpus, the top 8 terms are associated with "student" as shown in Fig. 3.

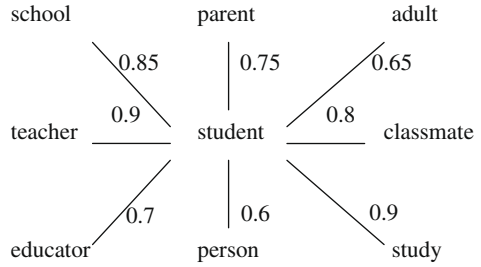

Fig. 3. Example of weighted term associations.

To derive the similarity between terms, term by document matrix is first conducted from the corpus, the term by document matrix is then normalized using Euclidean norm, of which each column is 1. The Euclidean vector norm $\|x\|_2$ is defined by:

$$\|x\|_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{m} x_i^2} \tag{12}$$

where $x = (x_1, x_2, \ldots, x_m)$, m is the number of terms. The relationship between two terms t_i and t_j is then computed as a correlation factor c_{ij}, and then normalized as $sim(t_i \cdot t_j)$. So we can get the relationship between every two terms using Eq. (14),

$$c_{ij} = \vec{t}_i \cdot \vec{t}_{j\cdot} = \sum_{\forall d_k} w_{ik} \cdot w_{jk} \tag{13}$$

$$sim(t_i \cdot t_j) = \frac{\vec{t}_i \cdot \vec{t}_j}{|\vec{t}_i| \cdot |\vec{t}_j|} = \frac{\sum_{\forall d_k} w_{ik} \cdot w_{jk}}{\sqrt{\sum_{\forall d_k} (w_{ik})^2} \cdot \sqrt{\sum_{\forall d_k} (w_{jk})^2}} \tag{14}$$

4. Experiments and results

4.1. Data set

In order to measure the performance of our system, we conducted the experiments on Ling-Spam corpus. Ling-Spam is one of the most popular benchmark corpuses adopted for development and test of emails (Sakkis et al., 2003). It is collected from a moderated mailing list of the profession and science of linguistics. The Ling-Spam corpus consists of 2893 messages, of which 2412 are legitimate messages and 481 are spam messages, approximately 16.6% spam percentage. The legitimate messages were classified as spam or non-spam manually. In our experiments, we use a lemmatized version of this corpus, with the tokenization given in the corpus and with no additional processing, and use a subset of 1000 emails for training and testing our system, which are randomly chosen from the lemmatized version of Ling-Spam corpora. We use 10-fold stratified cross-validation in all of our experiments, nine parts are used for training and the remaining part is used for testing.

4.2. Evaluation

Our experimental results are evaluated by spam precision (SP), spam recall (SR) and accuracy (Acc). According to the paper (Androutsopoulos, Koutsias, Chandrinos, & Spyropoulos, 2000), the SP, SR and Acc are defined as:

$$\text{SP} = \frac{n_{S\to S}}{n_{S\to S} + n_{L\to S}} \tag{15}$$

$$\text{SR} = \frac{n_{S\to S}}{n_{S\to S} + n_{S\to L}} \tag{16}$$

$$\text{Acc} = \frac{n_{L\to L} + n_{S\to S}}{N_L + N_S} \tag{17}$$

where $n_{L\to L}$ and $n_{S\to S}$ are the number of messages that have been correctly classified to the legitimate email and spam email, respectively; $n_{L\to S}$ and $n_{S\to L}$ are the number of legitimate and spam messages that have been misclassified; N_L and N_S are the total number of legitimate and spam messages to be classified.

4.3. Feature selection and final feature weighting

The main difficulty in the application of neural network for spam filtering is the high dimensionality of the input feature space. This is because each unique term in the vocabulary represents one dimension in the feature space, so that the size of the input of the

neural network depends upon the number of stemmed words. The selected features should balance the tradeoff between the classification accuracy and efficiency.

In our experiments, the training data have more than 20,000 features based on the number of words that have occurred in the documents. Fig. 4 shows the classification performance with different features using the BP neural network learning algorithm based on Ling-Spam corpus. From this figure, we can see that the performance increased with an increase in the number of features. When the number of features is up to 2000, it reaches a plateau. Therefore, for minimizing the system cost, we selected 2000 features for the data set by choosing the highest TF*IDF term weights according to the experiment shown in Fig. 4, approximately 90% feature reduction percentage. Thus, the selected features can be represented as

$$d_k = \langle w_{k,1}, w_{k,2}, \ldots, w_{k,2000} \rangle \tag{18}$$

where $w_{k,i}$ is the term weight of the ith indexing term in document k.

4.4. Final feature weighting

As mentioned before, document k is represented in terms of a vector d_k in Eq. (18), $w_{k,i}$ is the term weight of the term t_i when it is contained in document k, if t_i does not appear in document k, $w_{k,i}$ becomes 0.

The similarity between document k and term t_j can be defined as Eq. (19)

$$sim(d_k, t_j) = \sum_{t_i \in d_k}^{n} (w_i \cdot sim(t_i \cdot t_j)) \tag{19}$$

where $sim(t_i \cdot t_j)$ is the similarity between two terms that is calculated using Eq. (14). With respect to document k, all the terms in a collection can be ranked according to their $sim(d_k, t_j)$. The top n terms are used as expansion terms and are assigned the appropriate weight as:

$$e_w(d_k, t_j) = sim(d_k, t_j) / \sum_{t_i \in d_k} w_i \tag{20}$$

Then the expanded terms are added to the original vector to get the final term weight,

$$d_{ek} = (w_{k,1} + w'_1, \ldots, w_{k,n} + w'_n) \tag{21}$$

where w'_n is equal to $e_w(d_k, t_j)$ if t_j is in the top n ranked terms, otherwise it becomes 0.

Once all the training and test examples are represented by this way, the new document vectors are performed by the neural network classifier, and the neural network algorithm will decide which categories the test samples should be assigned into.

4.5. Error reduction

In order to distinguish the improvement of the authors' proposed revised BP neural network, we compared the mean absolute error using four different methods. The mean absolute error function is defined as Eq. (4). The first method is the conventional BP neural network (BP), the second method is the conventional BP neural network combining with the thesaurus (BP+ thesaurus), the third method is our proposed revised BP neural network (RBP), and the last one is the RBP neural network combining with the thesaurus (RBP+ thesaurus).

All the authors' experiments were conducted on a Pentium personal computer. The algorithms are written in C++ and all the programs are compiled in VC++ 6.0. All the figures in the experiments section are drawn by Matlab 6.5 using the data generated by the program.

From Fig. 5, in the case of the BP, we can see that the mean absolute error is reduced when the number of epochs increased.

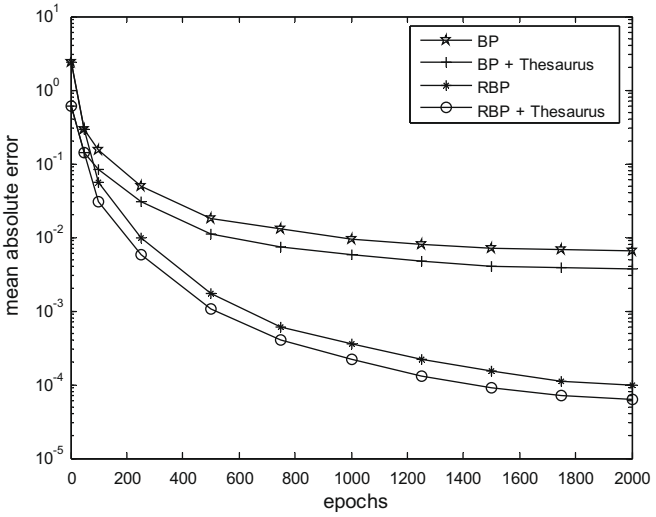

Fig. 5. Mean absolute error reduced when the number of epochs increased.

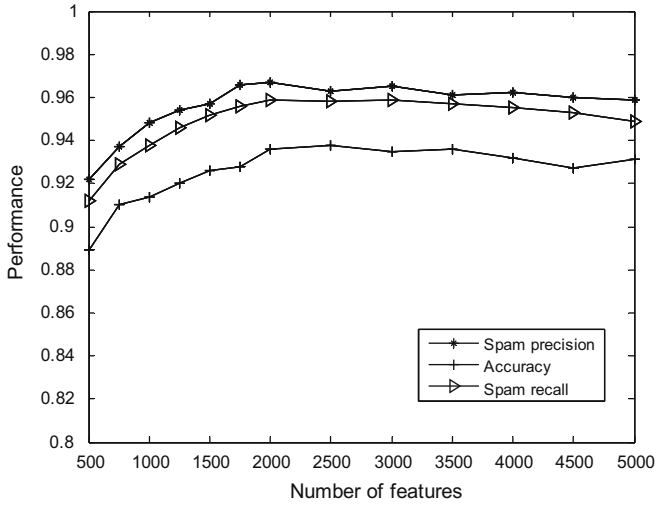

Fig. 4. Performance with different number of features.

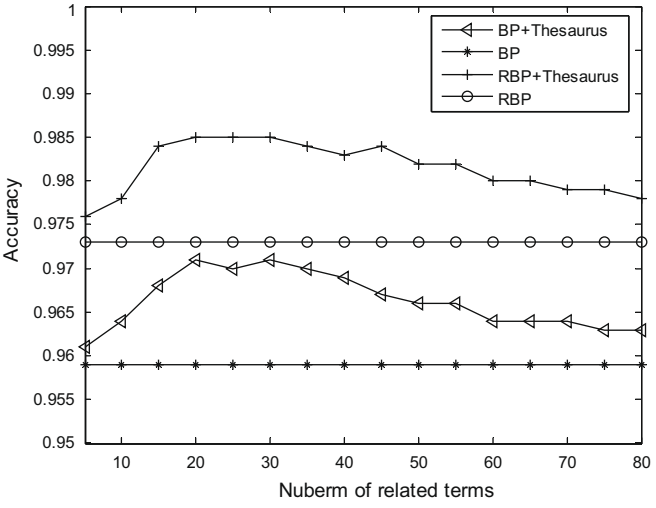

Fig. 6. Classification performance using four kinds methods.

At the beginning of the training phase, the error is reduced very rapidly. But this error reduction slows down after a certain number of epochs, and then levels off. However, in the case of our RBP, no morbidity neurons are produced in the first learning phase, with the result that the next learning phase is very similar to that of the BP. However, from the third learning phase, the RBP progresses more rapidly than the BP, it also has a good tendency in the later part of the training. The error reduction tendency of BP combining with the thesaurus (or RBP combining with the thesaurus) is similar to the BP (or RBP), but it has a smaller initial error.

4.6. Results

The experiments have been performed using the four methods mentioned above, and the results are given in Fig. 6. This figure also shows the effect of the number of related terms added to the original document, the performance is measured by the accuracy which is defined in Eq. (17). In our experiments, the number of related terms is ranked from 5, 10, 15,...,75 to 80. In the beginning, the accuracy increases when the number of related terms is increased, when the terms are up to 20, it reaches a plateau, then the accuracy begins to decrease when the related terms are more than 35. However, it still performs better than the benchmark method (BP or RBP). The results in Fig. 6 also indicate that the best performance is obtained with 15–35 top-ranked related terms.

The performance of the RBP is much better than that of the BP learning algorithms, and RBP in combination with thesaurus achieves the best performance among the four methods.

5. Conclusions and future works

In this paper, we proposed two methods to improve the performance of spam filtering system: one is automatic thesaurus construction, and another is the RBP neural network algorithm. Both the methods showed improvements when compared with the benchmark method, conventional BP neural network learning algorithm. The combination of the two methods achieved very promising results. In this paper, we have conducted the experiments using only the neural networks classifier, even it obtains good results, but we have not compared it with the other classification approaches such as SVM. More experiments will be done in the future to compare the performance of neural network and of the other classifiers for spam filtering using different feature selection approaches.

Acknowledgment

The research work in this paper is supported by the National Natural Science Foundation of China, under Grant No. 70031010.

References

Aladdin knowledge systems. Anti-spam white paper. <http://www.eAladdin.com>.

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., et al. (2000). An evaluation of naive bayesian anti-spam filtering. In *Proceedings of workshop on machine learning in the new information age, Barcelona* (pp. 9–17).

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 160–167).

Ángel, F. Z., Carlos, G. F., José, L. A. B., & Emilio, R. (2005). Reformulation of queries using similarity thesauri. *Information Processing and Management, 41*(5), 1163–1173.

Bang, S. L., Yang, J. D., & Yang, H. J. (2006). Hierarchical document categorization with *k*-NN and concept-based thesauri. *Information Processing and Management, 42*(2), 387–406.

Bezerra, G. B., Barra, T. V., Ferreira, H. M., Knidel, H., de Castro, L. N., & Von Zuben, F. J. (2006). An immunological filter for spam. In *Proceedings of the international conference on artificial immune systems, Oeiras, Portugal* (pp. 446–458).

Blanzieri, E., & Bryl, A. (2008). *A survey of learning-based techniques of email spam filtering.* Technical report DIT-06-056 (updated version), University of Trento, Italy.

Carpinteiro, O. A. S., Lima, I., Assis, J. M. C., de Souza, A. C. Z., Moreira, E. M., & Pinheiro, C. A. M. (2006). A neural model in anti-spam systems. *Lecture notes in computer science* (Vol. 4132, pp. 847–855). Berlin: Springer.

Chen, D. H., Chen, T. J., & Ming, H. (2003). Spare email filter using naive bayesian, decision tree, neural network and adaboost. <http://www.cs.iastate.edu/~tongjie/spamfilter/paper.pdf>.

Clark, J., Koprinska, I., & Poon, J. (2003). A neural network based approach to automated email classification. In *Proceedings of IEEE/WIC international conference on web intelligence, Halifax, Canada* (pp. 702–705).

Crawford, E., Kay, J., & McCreath, E. (2001). Automatic induction of rules for email classification. In *Proceedings of the 6th Australasian document computing symposium, Coffs Harbour, Australia* (pp. 13–20).

Crawford, E., Koprinska, I., & Patrick, J. (2004). Phrases and feature selection in email classification. In *Proceedings of the 9th Australasian document computing symposium, Melbourne, Australia.*

Cui, B., Mondal, A., Shen, J., Cong, G., & Tan, K. L. (2005). On effective e-mail classification via neural networks. In *Proceedings of the 16th international conference on database and expert systems applications (DEXA05), Copenhagen, Denmark* (pp. 85–94).

Delany, S. J., & Bridge, D. (2007). Caching the drift: Using feature free case-based reasoning for spam filtering. In *Proceedings of the 7th international conference on case based reasoning* (pp. 314–328).

Drucker, H., Wu, D., & Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10*(5), 1048–1054.

Fdez-Riverola, F., Iglesias, E. L., Díaz, F., Méndez, J. R., & Corchado, J. M. (2007). Applying lazy learning algorithms to tackle concept drift in spam filtering. *Expert Systems with Application, 33*(1), 36–48.

Gavrilis, D., Tsoulos, I. G., & Dermatas, E. (2006). Neural recognition and genetic features selection for robust detection of e-mail spam. In *Proceedings of the 4th Helenic conference on AI, Heraklion, Crete, Greece. Lecture notes in computer science* (Vol. 3955, pp. 498–501). Berlin: Springer.

Jiang, E. (2006). Learning to semantically classify email messages. In *Proceedings of the international conference on intelligent computing, Kunming, China* (pp. 700–711).

Kolcz, A., Chowdhury, A., & Alspector, J. (2004). The impact of feature selection on signature-driven spam detection. In *Proceedings of the first conference on email and anti-spam.*

Ma, S., & Ji, C. Y. (1998). A unified approach on fast training of feedforward and recurrent networks using EM algorithm. *IEEE Transaction on Signal Processing, 46*(8), 2270–2274.

Ooyen, A., & Nienhuis, B. (1992). Improving the convergence of the back propagation algorithm. *Neural Networks, 5*(3), 465–471.

Pérez-Agüera, J. R., & Araujo, L. (2006). Query expansion with an automatically generated thesaurus. In *Proceedings of the intelligent data engineering and automated learning. Lecture notes in computer science* (Vol. 4224, pp. 771–778).

Plagianakos, V. P., & Vrahatis, M. N. (2000). Training neural networks with threshold activation functions and constrained integer weights. In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks (IJCNN'00)* (Vol. 5, pp. 51–61).

Qiu, Y. G., & Frei, H. P. (1993). Applying a similarity thesaurus to a large collection for information retrieval. Available from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.8211&rep=rep1&type=pdf.

Riverola, F., Iglesias, E. L., Díaz, F., Méndez, J. M., & Corchado, J. M. (2007). SpamHunting: An instance-based reasoning system for spam labeling and filtering. *Decision Support Systems, 43*(3), 722–736.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval, 6*(1), 49–73.

Wang, B., Jones, G. J. F., & Pan, W. (2006). Using online linear classifiers to filter spam emails. *Pattern Analysis and Applications, 9*, 339–351.

Wang, H. B., Yu, Y., & Liu, Z. (2005). SVM classifier incorporating feature selection using GA for spam detection. In *Proceedings of the 2005 international conference on embedded and ubiquitous computing, Nagasaki, Japan* (pp. 1147–1154).

Wu, W., Feng, G. R., Li, Z. X., & Xu, Y. S. (2005). Deterministic convergence of an online gradient method for BP neural networks. *IEEE Transactions on Neural Networks, 16*(3), 533–540.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 4–11).

Yu, X. H., Chen, G. A., & Cheng, S. X. (1993). Acceleration of back propagation learning using optimized learning rate and momentum. *Electronics Letters, 29*(14), 1288–1289.