Topic Models based Personalized Spam Filter

Sudarsun Santhiappan, *Member IEEE*, Venkatesh Prabhu Gopalan and Valarmathi B

***Abstract*-- Spam filtering poses a critical problem in text categorization as the features of text is continuously changing. Spam evolves continuously and makes it difficult for the filter to classify the evolving and evading new feature patterns. Most practical applications are based on online user feedback, the task calls for fast, incremental and robust learning algorithms. This paper presents a system for automatically detection and filtering of unsolicited electronic messages. In this paper, we have developed a content-based classifier, which uses two topic models LSI and PLSA complemented with a text pattern-matching based natural language approach. By combining these powerful statistical and NLP techniques we obtained a parallel content based Spam filter, which performs the filtration in two stages. In the first stage each model generates its individual predictions, which are combined by a voting mechanism as the second stage.**

Index Terms **–** Dimension Reduction, LSA, N-Gram, PCA, PLSA, Spam Filter, Topic Models, Vectorization

I. Introduction

SPAM is briefly defined by the TREC 2005 Spam Track as "unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by the sender having no current relationship with the recipient". Recent, reliable data indicates that more than 50% of email received currently is Spam. The flow of unsolicited email generates a considerable cost for businesses and for users. Since Spam growth is exponential and prevention is both extremely difficult and rare, the problem must be tackled by developing scientific methods to analyze email traffic in order to identify and reject Spam communications. These methods may be classified into two categories; origin based filtering and content based filtering. Filtering based on origin focuses on the source of the email, recorded by the domain name and the address of the sender device. The goal and analysis of the content-based filters is to review the text contents of emails. Depending on the analysis technique used, several kind of filters can be differentiated [5]: rule based filters that extracts text patterns on rules [2] and assigns a score to each rule based on the occurrence frequency of the rule in Spam and non-Spam emails belonging to a historic corpus of emails; Bayesian Filtering [1] that analysis every word of email and assign Spam and non-Spam probabilities to each word based on statistical measurements; Memory-based filtering that uses email comparison as its basis for analysis [3] and wastes lots of memory in order to achieve an acceptable level of performance; other filtering methods use support vectors [4] or neural networks [6].

Topic models are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words. A topic model is a *generative model* for documents: it specifies a simple probabilistic procedure by which documents can be generated. To make a new document, one chooses a distribution over topics. Then, for each word in that document, one chooses a topic at random according to this distribution, and draws a word from that topic. Standard statistical techniques can be used to invert this process, inferring the set of topics that were responsible for generating a collection of documents.

In this work we used two statistical techniques LSA (Latent Semantic Analysis) and PLSA (Probabilistic Latent Semantic Analysis) and a language model approach, N-Gram for Spam classification. Basically Latent Semantic Analysis (LSA) can establish keyword relevancy. LSA identifies latent structure present among the documents (here mails) using Singular Value Decomposition (SVD) and establishes Word-Word binding [7]. LSI is the modification of the vector-retrieval method that explicitly models the correlation of term usage across documents using a reduced dimension. Probabilistic Latent Semantic Analysis (PLSA) is a novel statistical technique for the analysis of two-mode and co-occurrence data, which has applications in information retrieval and filtering, natural language processing, machine learning from text, and in, related areas [9]. We describe here an N-gram based approach to spam classification that is tolerant of classification errors. The system is based on calculating and comparing profiles of N-gram frequencies. To compute similarity, we basically count the number of n-grams in the query that match n-

Sudarsun Santhiappan, Venkatesh Prabhu Gopalan are with Checktronix India Pvt. Ltd, Chennai 600034 (e-mail: {sudarsun, vprabhu}@burning-glass.com).

Valarmathi B is with SKP Engineering College, Thiruvannamalai 606611, India (e-mail: valar_mathi_99@yahoo.com).

grams in the target. We used PCA (Principal component analysis) for dimension reduction of the vectors generated by topic models. The final results of the LSA and PLSA combined by a combiner (simple neural network). This system is highly effective at classifying email and highly efficient at managing the resources. Incremental machine learning carries out the evolution of the system. The system was developed to identify and filter emails based on the LSA and PLSA statistical classification model with frequent intervals.

II. LSI, PLSA AND PCA

Latent Semantic Indexing (LSI) can establish keyword relevancy. LSI identifies latent structure present among the documents using Singular Value Decomposition (SVD) and establishes Word-Word binding. LSI is the modification of the vector-retrieval method that explicitly models the correlation of term usage across documents using a reduced dimension. Latent Semantic Analysis (LSA) [11], [7] is a statistical technique, which describes the underlying structure of texts. It is widely used in author recognition; search engines and computes similarity between texts. LSA addresses the problem of Synonymy.

Many information access tasks rely on comparison of the terms in a document. Often, Latent Semantic Indexing (LSI) is used to better match words that are synonyms and better handle the multiple meaning of a term. LSA uses singular value decomposition to map the high dimensional word-document count matrix to a lower dimensional latent "semantic" space wherein terms and documents that are closely associated are placed near one another.

Probabilistic latent semantic analysis (PLSA) is a statistical latent class model that has been found to provide better results than LSA for term matching in retrieval applications. In PLSA, the conditional probability between documents d and words w is modeled through a latent variable z, which can be loosely thought of as a class or topic. A PLSA model is parameterized by $P(w|z)$ and $P(z|d)$, and the words may belong to more than one class and a document may discuss more than one "topic". It's assumed that the distribution of words given a class, $P(w|z)$ is conditionally independent of the document, i.e., $P(w|z,d)=P(w|z)$. Thus the joint probability of a document d and a word w is represented as:

$$P(w,d) = P(d)\sum_{z} P(w \mid z)P(z \mid d) \qquad (1)$$

The original feature space transformed with the Vector Space Model may contain tens of thousands of different features, and not all classifiers can handle such a high dimension gracefully. Dimension reduction (also called feature pruning or feature selection) is usually employed to reduce the size of the feature space to an acceptable level, typically several orders of magnitude smaller than the original one.

III. PROPOSED SYSTEM & ARCHITECTURE

The proposed system is in the parallel classifiers architecture. Multiple classifiers are performed individual classification of the input mails and the result is combined in the final stage by a simple neural combiner.

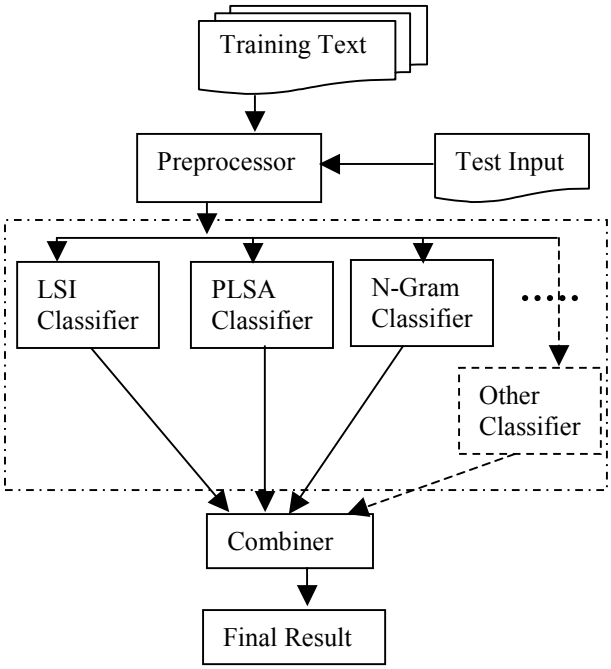

Fig. 1. Overall architecture of the system.

"Fig. 1" shows the proposed architecture. We built two models for LSA and PLSA by using the existing spam corpus. The N-Gram classifier learns the rules on the fly. The detail procedure of the proposed Spam classification is discussed in this section.

A. PLSA Classification

We build a global PLSA model incrementally from the received mails (Spam and Non-Spam). Since the model building procedure is resource crunchy, the model is built every few hundred new emails. The choice of model parameters like aspect count, iterations, weighting function is chosen based on various experiments [10]. Using PCA, P(z|d) aspect model is reduced to P(z'|d), which in turn is used to train [10] the BPN along the tagging provided by an email user. PLSA projects the input document text to a Z-dimensional subspace which is further reduced by PCA to lower Z' dimensional subspace.

The BPN based neural network is trained on Z' principal component inputs and one output which is Spam or non-Spam. Alternatively, P(z|w) projection of the input mail text on the global PLSA model may also be used to train [10] the BPN network. In this case, P(z|w) projection is reduced to P(z'|w) by PCA and fed to the BPN input nodes. We used the training corpus used of PLSA model building as the training corpus for neural network as well. The steps involves; 1) Project the training document (mail) on the PLSA model. 2) Use the vector generated by PLSA model as input to the PCA for dimensionality reduction. 3) Use this vector for the neural network training. We repeated the above steps for a subset of the training corpus of PLSA model covering all the mails to be learned by neural net. By this process, the BPN learns about the classification based on the vectors generated by the PLSA model and PCA, which was found to improve the classification to a greater extent. While testing, 1) a test document (incoming mail) is fed to PLSA for projection and vectorization. 2) The vector (of aspects) is fed to the PCA. 3) The output of the PCA is given to the BPN network. 3) The BPN network emits its prediction with a confidence score. This score will be given as an input to the final combiner.

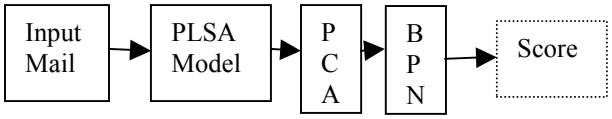

Fig. 2. PLSA classification procedure.

B. LSI Classification

This method is very similar to PLSA method. With LSI, i) rank of the SVD is analogous to number of aspects in PLSA, ii) left singular vector matrix (U) is analogous to P(z|w) aspect model, iii) right singular vector matrix (V) is analogous to P(z|d) aspect model, iv) singular values vector is analogous to P(z) aspect vector. Like PLSA model building, global LSI model is built from all the available mails incrementally after few hundred new mail messages. The model parameters like rank, weighting functions are chosen based on various experiments described in [8]. The LSI vector for each training document is of 'k' dimension, where 'k' is the best rank of LSA model. The dimensionality of this matrix of projection vectors is reduced to 'r' dimensions using Principal Component Analysis (PCA). This reduced matrix is fed in to BPN based Neural Network, which has 'r' number of input nodes and one output node which takes the value tagged by an email user as Spam or non-Spam. The training documents for LSI model building and Neural Networks are same. Steps involved; 1) Vectorize the training documents using LSI model. 2) Reduce the dimensionality of the matrix of pseudo-vectors of training documents using PCA. 3) Feed the reduced matrix into Neural Network system for learning. By this process, the BPN learns about the classification based on the vectors generated by the LSI model, which would improve the classification rate considerably. Testing phase includes 1) test document is fed to LSA for vectorization. 2) Vector is reduced using PCA model. 3) Reduced vector is fed into the BPN neural network. 4) The BPN network emits its prediction with a confidence score. This score will be given as an input to the final combiner.

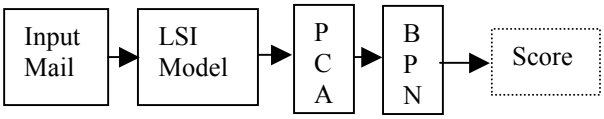

Fig. 3. LSI classification procedure.

C. N-Gram based Approach

An N-Gram is a string, typically a phrase of length N words. N-Gram approach exploits the fact that Spam mails possess repeating phrases or string patterns. The previous two methods targeted on the context of the Spam mails and used that as a feature to differentiate against legitimate mails. N-Grams on the contrary, looks for repeating patterns, which are pre-dominant in Spam mails and legitimate mails. N-Grams are superior than words based filters like Bayes in the sense, N-Grams with a min N=2 show more confidence in judging patterns.

The proposed system constructs a tree whose leaves are training documents (Spam or non-Spam) and the intermediate nodes are N-Grams extracted from its children. The weightage of a N-Grams is defined by its number of children. When this tree is expanded, it yields a list of N-Grams together with its weights and it's frequency. Using IWF weighting [8], it becomes easy to get rid of noisy N-Grams as IWF applies a very small weight to a very frequent occurrence. The advantage of such a system is that, the entire training in unsupervised.

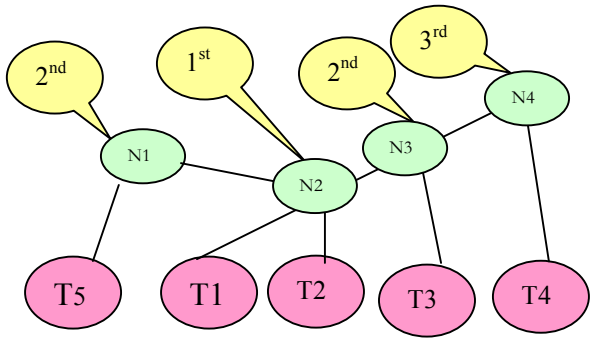

Fig. 4. An Example N-Gram Tree.

"Fig. 4" shows the arrangement of N-Grams for an example case where T1 through T5 are the test documents. And N1 through N4 is the N-grams found with various combinations of the documents. Out of the identified 4 N-Grams, N4 takes more weightage because it is common across four documents, whereas N2 is a 1st order N-Gram existing between only a pair of documents. When the above tree is expanded, we get four N-Grams with 3 different weights. When the tree is bigger, N-Grams of a particular order say 'P' can exist amongst multiple sets of 'P+1' documents, which yields the frequency component for computation of weights.

The weight is derived based on the order of a N-Gram (number of documents that share it) and the frequency of the N-Gram (number of set of documents that share it). We have not explicitly differentiated between a Spam text and a legitimate text so far in the extraction of N-Grams. For instance, if we consider an N-Gram with order 'P' and frequency 'F', the weight 'W' is diminished based on the ratio of Spam and legitimate mails in the set of size 'P'.

$$W \leftarrow W * \frac{S}{S+L} \qquad (2)$$

Where P: Size of the document set sharing the N-Gram
S: Number of Spam in P documents
L: Number of Legitimate in P documents

When the number of Spam text is less in the set of documents that constituted an N-Gram, the weight of that N-gram is greatly diminished. In N-Gram language models, each word depends probabilistically on the n-1 preceding words:

$$P(w_1 \ldots w_n) = \prod P(w_i | w_{i-n+1} \ldots w_{i-1}) \qquad (3)$$

Once the model (M-order N-Gram Tree) is built with our training corpus, any test document can be searched for our N-Gram collection. Using the weights of the matching N-Grams, we may predict whether the test document is indeed a Spam mail or a legitimate mail. It is done by thresholding. The threshold value can also be derived in an unsupervised fashion by passing our training document one after the other to get matching N-Grams along with their weights. The average weight computed for the Spam collection in training set could be used as the threshold.

The simplest algorithm to extract N-Grams from a pair of documents is the windowing method where a window of constant width say 10 words (from document 1) is chosen and glided over 10 words of document 2. When there is an exact match, the next 10 words starting from the 11th word is taken in document 1 and the procedure is repeated. When there are no matching in the first iteration, the window size is reduced to 9 and the procedure is repeated until a match is found in document 2. When all the words in first window are elapsed, the window is moved to the next word starting from 2nd word to 11th word and same procedure is repeated. This is a brute force method. A generic algorithm for the described method is presented below.

Algorithm 1: N-Gram Detection

```
Let N1, N2 be the number of tokens in
documents D1 and D2
Let the initial window size be W
Initialize Set NG tp
For I in 0, N1-W
  For P in W, 1 Step -1
    NG1←Tok(I)..Tok(I+P)
    Set match = false
    For J in 0, N2-P
      NG2←Tok(J)..Tok(J+P)
      If (NG1==NG2)
        I ← I+P
        NG ← NG+NG1
        match = true
        break
      Endif
    EndFor
    If (match == true)
      Break
    Endif
  EndFor
EndFor
```

The training process in time CPU intensive and hence it could be scheduled when the processor is idling. But the testing process in indeed faster, which can be easily integrated with the mail, fetching process of any mail client. The primary advantage of this system is that it is completely unsupervised other than marking a received mail as Spam or not-Spam.

D. The Combiner

This combiner is a simple BPN network, which accepts the scores from the LSI, PLSA models, and N-Gram predictor on its input nodes and gives the final classification result. Basically this combiner acts like a binary classifier classifying the document into either of two classes (Spam, Not Spam). One can use a simple voting combiner, which just goes by which technique yielded the maximum confidence score.

IV. CONCLUSION

The proposed system is intended to filter mail messages only based on the preference of an individual. The performance of the proposed system increases with more received mails. The advantage of all 3 systems is that they don't need any initial learning. The features are

extracted automatically from the training and testing set. Since, the models are rebuilt and N-Grams are updated continuously, the proposed system could handle evolving Spam more elegantly in an unsupervised fashion. The only disadvantage is the CPU-intensive model building process and N-Gram tree building process, which need to be scheduled only when the CPU is idling. But the bigger advantage of the system is the fast prediction process.

V. REFERENCES

[1] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. "An Evaluation of Naïve Bayesian Anti-Spam Filtering", *Proc. of the workshop on Machine Learning in the New Information Age, 2000.*

[2] W. Cohen, "Learning rules that classify e-mail", *AAAI Spring Symposium on Machine Learning in Information Access, 1996.*

[3] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch, "TiMBL: *Tilburg Memory-Based Learner - version 4.0 Reference Guide", 2001.*

[4] H. Drucker, D. Wu, and V. N. Vapnik., "Support Vector Machines for Spam Categorization", *IEEE Trans. on Neural networks*, 1999.

[5] D. Mertz, "Spam Filtering Techniques. Six approaches to eliminating unwanted e-mail.", Gnosis Software Inc., September, 2002. Ciencias Físicas, Universidad de Valencia, 1992.

[6] M. Vinther, "Junk Detection using neural networks", MeeSoft Technical Report, June 2002. Available: http://logicnet.dk/reports/JunkDetection/JunkDetection.htm.

[7] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. "Indexing By Latent Semantic Analysis", *Journal of the American Society For Information Science*, 41, 391-407. (1990)

[8] Sudarsun Santhiappan, Venkatesh Prabhu Gopalan, and Sathish Kumar Veeraswamy,"Role of Weighting on TDM in Improvising Performance of LSA on Text Data", *Proceedings of IEEE INDICON 2006*.

[9] Thomas Hofmann, "Probabilistic Latent Semantic Indexing," *Proc. 22 Int'l SIGIR Conf. on Research and Development in Information Retrieval, 1999*

[10] Sudarsun Santhiappan, Dalou Kalaivendhan and Venkateswarlu Malapatti **."**Unsupervised Contextual Keyword Relevance Learning and Measurement using PLSA", *Proceedings of IEEE INDICON 2006.*

[11] Landauer, T. K., Foltz, P. W., & Laham, D. "Introduction to Latent Semantic Analysis", DiscourseProcesses, 25, 259-284. (1998).

[12] G. Furnas, S. Deerwester, S. Dumais, T. Landauer, R. Harshman, L. Streeter and K. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure," *in The 11th International Conference on Research and Development in Information Retrieval, Grenoble, France: ACM Press*, pp. 465--480. (1988)

[13] Jiawei Han, Micheline Kamber, "*Data mining -- Concepts and Techniques", Morgan Kaufmann Publishers, 2001.*

[14] Thorsten Brants, Francine chen, Loannis Tsochantarids, Topic Based Document Segmentation Using Probabilistic Latent Semantic Analysis.

[15] Damashek, M. Gauging , "Similarity via N-Grams: Language-Independant Sorting, Categorization and Retrieval of Text". *Science*, *267*. 843-848.

[16] Sholomo Hershkop, Salvatore J.Stolfo , "Combining Email models for False Positive Reduction", *KDD'05, August 2005.*

VI. BIOGRAPHIES

Sudarsun S (M' 2002) is the Director – R & D at Checktronix India Pvt Ltd, Chennai. He holds an M.Tech Computer Science from IIT Madras. He received a BE degree in Electronics and Instrumentation Engineering from Madras University with a Gold Medal. His research experience includes Statistical Natural Language Processing, Machine Learning and Distributed Computing.

Venkatesh Prabhu G is a Research Associate at Checktronix India Pvt Ltd, Chennai. He holds an ME in Computer Science from Anna University, Chennai. He completed his BE in Computer Science at MKU, Madurai. His research interests include AI, Data Mining, Pattern Classification, Knowledge Based Neural Networks, Machine Learning, Committee Machines and Hybrid Techniques in Soft Computing.

Valarmathi B is a Professor and Head of Computer Science and Engineering at SKP Engineering College, Thiruvannamalai. She holds an M.Tech Computer Science from IIT Madras. She received her BE degree in Electronics and Communication Engineering from Bharathidasan University. She is currently pursuing her PhD at Anna University, Chennai. Her research areas include Natural Language Processing and Software Quality.