Using Latent Semantic Indexing to Filter Spam

Kevin R. Gee
Dept. of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX 76019
geek@cse.uta.edu

ABSTRACT

Past research has explored the effectiveness of a Naïve Bayesian classifier when filtering unsolicited bulk email (or "spam"). Results have shown that the degree of precision of this approach is generally superior to the degree of recall. This study evaluates the effectiveness of a classifier incorporating Latent Semantic Indexing ("LSI") to filter spam email, using a corpus used in previous studies. Results show that using LSI as the basis for an email classifier to filter out spam enjoys a very high degree of recall as well as a high degree of precision, no matter if the corpus is treated using a stop list or a lemmatizer. While using LSI leads to precision roughly equal to that of using a Naïve Bayesian approach, the LSI technique has a substantially higher recall and is generally more effective under certain conditions.

A conclusion is made that incorporating Latent Semantic Indexing into an anti-spam filter is viable, particularly in implementations when misclassified legitimate messages are not arbitrarily deleted. Other inferences are drawn to the applicability of this method to other text mining tasks.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *language parsing and understanding, text analysis.*

General Terms

Algorithms

Keywords

Latent semantic indexing, LSI, spam, email, text classification.

1. INTRODUCTION

The problem of "spam" email is apparent to any frequent email user: Unwanted, unsolicited messages are emailed en masse to a large number of users indiscriminately, similar to bulk mailings sent through the traditional postal service. While spam is by definition a function of it being sent in an unsolicited manner from an anonymous third party ([11]), it generally employs a distinct tone and language that can be used to identify it ([1]).

There have been several previous attempts to classify spam based on a Naïve Bayesian approach. [9] analyzed a corpus of manually categorized email, using words, phrases, and non-textual characteristics. [1] preprocessed manually-categorized email list postings into four separate corpa using a lemmatizer and a stop list. Both achieved fairly high degrees of precision, but the recall accuracy was slightly less, meaning that both studies found that spam emails were being classified as legitimate. [1] also noted that outright deletions of spam suffer from a relatively high cost, due to the possibility that legitimate emails were being erroneously classified as spam.

This paper analyzes the effectiveness of another machine learning approach, latent semantic indexing, to the problem of filtering spam and legitimate email. Latent semantic indexing is a statistical technique that derives correlations between terms and documents in a corpus and reflects indirect inferential relationships between terms, e.g., like "dog" or "canine" ([7]). Since latent semantic indexing can produce correlations between a document and terms that don't actually appear in the document, it potentially enjoys a much higher degree of recall but a lesser degree of precision ([10], [4]).

Previous studies have used latent semantic indexing on non-textual data ([5]), the training of vocabulary terms ([7], [6]), and the characterization of documents ([8], [7], [6], [4]), among others. This paper will show the results of an email classification test where both the recall and precision measurements are both very high and fall into acceptable levels.

2. LATENT SEMANTIC INDEXING

Latent semantic indexing (LSI) as described by [2] is a statistical technique that derives a statistical correlation between all terms and documents in a corpus, in an attempt to overcome the problems inherent in lexical matching.

At a theoretical level, the LSI process is a modified kernel machine. Specifically (see [2] for specific details), a support vector machine (SVM) is created using as a term-document frequent matrix. Singular value decomposition (SVD) is applied to estimate the term usage across all documents in the corpus, deriving in the process conceptual indices that approximate the underlying word usage structure. Then, to avoid the noisy effects due to excessive variability in the vocabulary usage, the SVD-derived matrices are reduced to an arbitrary k dimensions ([2], [6], [7]). [3] provides an excellent background on kernel machines and how they are not susceptible to local minima like other methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC 2003, Melbourne, Florida, USA
© 2003 ACM 1-58113-624-2/03/03...$5.00

The end result is a condensed vector for each term and document that is a linear combination of data from every other matrix cell ([6]). Retrieval and searching is performed using the database of singular values and vectors obtained from the reduced SVD matrices. Studies have shown that these vectors are robust, effective indicators of meaning and enjoy a higher recall than searching only with individual terms ([4], [6], [7]).

One issue with LSI is that it does not support the ad-hoc addition of new documents once the semantic set has been generated. Any update to any cell value will change the coefficient in every other word vector, as SVD uses all linear relations in its assigned dimensionality to induce vectors that will predict every text samples in which the word occurs ([6]).

To compare two vectors, the dot product is used to generate a cosine of the angle between the two vectors. [2] provides an exact summary of how this is generated. A cosine of 1 signifies that the two vectors (be they term or document) are considered to be exactly similar (which is different from identical; a cosine of -1 would signify that they are theoretically completely dissimilar. New test documents not previously included in the semantic set can be used for comparison as well, by combining the vectors of their composite terms (Ibid.).

3. PREVIOUS RESULTS

Using the classic definitions for recall and precision, here is the definition for precision (SP and LP) and recall (SR and LR) for both spam and legitimate documents:

Recall: $SR = \frac{S \to S}{S \to S + S \to L}$ $LR = \frac{L \to L}{L \to L + L \to S}$

Precision: $SP = \frac{S \to S}{S \to S + L \to S}$ $LP = \frac{L \to L}{L \to L + S \to L}$

"S→S" and "L→L" are the number of spam and legitimate documents identified correctly, while "L→S" and "S→L" refer to the incorrectly classified legitimate and spam documents.

Table 1 outlines results from [9]; the second, third, and fourth experiments made use of phrases ("be over 21") and non-textual features (whether or not the email included attachments, etc.)

Table 1. Results from [9] (500 attributes).

Attrb	Total Msgs	Test Msgs	Spam	Spam Prec	Spam Recall
Words	1789	251	88.2%	97.1%	94.3%
Words+ Phrases	1789	251	88.2%	97.6%	94.3%
Words+ Phrases+ Non-textual	1789	251	88.2%	100.0%	98.3%
Words+ Phrases+ Non-textual	2815	222	88.2%	92.3%	80.0%

Table 2 shows the output from the experiments conducted by [1]. That study produced a corpus known as Ling-Spam, where 2893 postings to the Linguist mailing list (a moderated list) were hand-categorized. Using a lemmatizer and a stop-list, four separate corpa were created: bare (untreated), lemm (preprocessed using the lemmatizer only), stop (preprocessed using the stop list only), and lemm+stop (preprocessed using both the stop list and lemmatizer). In all, there were 481 spam messages (out of 2893 total), which is about 16%. Note that the Linguist mailing list is moderated, so a 4:1 legitimate-to-spam ratio is not to be unexpected; for an unmoderated list or the average email user's inbox, this would be quite low ([1]).

Table 2. Results from [1] (variable number of attributes).

Filter Config	Num Attrib	Spam Recall	Spam Precision
Bare	50	81.10%	96.85%
Stop	50	82.35%	97.13%
Lemm	100	82.35%	99.02%
Lemm+stop	100	82.78%	99.49%
Bare	200	76.94%	99.46%
Stop	200	76.11%	99.47%
Lemm	100	77.57%	99.45%
Lemm+stop	100	78.41%	99.47%
Bare	200	73.82%	99.43%
Stop	200	73.40%	99.43%
Lemm	300	63.67%	100.00%
Lemm+stop	300	64.05%	100.00%

The precision is similar to that shown by [9], but the recall is substantially less. This is best explained by the probability that many of the spam documents are being identified as legitimate.

4. METHODS

This study sought to compare the results of using a Naïve Bayesian classifier with the results from using an LSI-inspired classifier. For this purpose, the same four corpa from the Ling-Spam collection ([1]) were used, with no additional preprocessing or stop lists. LSI semantic sets for each of the four corpa in the Ling-Spam corpus (bare, lemm, stop, lemm+stop) were created using different LSI dimensions (k=50, 100, 150, and 200).

A classifier was implemented to test the classification of each of the documents, using three different algorithms.

4.1 Nearest Neighbor

Each test document d_{test} was compared against training document dtrain, and for each d_{test}, the document d_{train} deemed most similar (e.g., had the highest cosine) was used to perform the classification.

4.2 Majority Count

To eliminate the risk of the situation where the single most similar document is actually from a different classification, a "majority" test was implemented, where a list is generated of all the training documents with a cosine to dtest that is greater than or equal to an

arbitrary threshold tr. The classification (spam or legitimate) that has the higher count of training documents returned is used to classify dtest (a tie vote classifies the document as legitimate).

The tradeoff here is that if too few documents are returned when doing the majority test, then the algorithm might inadvertently depend on some similar documents of a different classification (e.g., an email is legitimate yet the two most similar documents are both spam). Alternately, if the threshold *tr* is set too low, then many more documents will be returned, skewing the results in favor of the category with more representation (in Ling-Spam, over 80% of the documents are legitimate). The classifier should be able to return an appropriate number of documents (perhaps the closest 1% or so) without returning too few.

4.3 Nearest Neighbor

In part because LSI generally is less precise, but enjoys higher recall, it is conceivable that the single most similar document to a test document dtest is of the correct classification, yet the corpus also contains many other highly similar documents of different categories. In this case, the nearest neighbor approach would succeed while the majority test would fail. The reverse is also true, posing a problem. To attempt to improve the accuracy in situations where the majority classifier disagreed with the nearest neighbor classifier, an ensemble classifier was constructed that compared the majority results (the percentage of documents in the class) with the cosine retrieved by the nearest neighbor query. In general, if one measurement was considered "strong" and the other "weak", the "stronger" measurement prevailed. In all, six tests were designed, each with a vote:

If majority (MAJ) and nearest neighbor (NN) disagree:

1. If MAJ score is > a and NN cosine < b, use MAJ classification
2. If MAJ score is < b and NN cosine > a, use NN classification
3. If NN predicts legitimate with cosine > c, vote as legitimate
4. If NN predicts spam with cosine > c, vote as spam
5. If MAJ test predicts legitimate with score > c, vote as legitimate
6. If MAJ test predicts spam with a score > c, classify as spam

Tests 1 and 2 are mutually exclusive, as are tests 3 and 4, as well as 5 and 6. A document that passes either test 1 or test 2 will then pass, by definition, one of the other tests. Such documents are more heavily skewed toward one classification than the other. Consider the case of a document where the majority test indicated a score of 58% spam (or 42% legitimate) and the cosine to its nearest neighbor (a legitimate document) was 0.85. For example, if a=0.70, b=0.70, and c=0.65, then this document would be classified as legitimate, since it would pass tests 2 and 3.

Any ties between a spam and a legitimate classification at this stage are broken by declaring the document legitimate, since in general, it is worse to delete a legitimate document than let pass a spam mail ([1]).

The final results will show that the ensemble method does provide a slightly better degree of improve with some of the test corpa. In general, it should do no worse, since its purpose is to settle disputes between the nearest neighbor and majority methods.

5. RESULTS

Tables 3, 4, and 5 show the results from the LSI-inspired classifiers when using LSI dimensions of k=50, 100, 150, and 200 (other dimensions produced similar output) on semantic sets derived from the bare, lemm, stop, and lemm+stop corpa from the Ling-Spam corpus.

Table 3 shows the best results for each combination of LSI *k* dimension, corpa, and method. In the interest of brevity, methods that achieved the same results for a particular *k* and corpa are grouped together, where "Maj" refers to the Majority count, "NN", refers to the Nearest Neighbor approach, and "Ens" refers to the Ensemble Voting method.

Table 3. Top dimensions, corpa, and method combinations.

Measure	LSI *k*=	Corpus	Method	Results
Spam precision	50	lemm	Maj, Ens	98.96%
	100	lemm	NN, Ens	98.95%
	150	lemm	Ensemble	98.95%
	200	bare	Ensemble	98.55%
	200	lemm	Ensemble	98.55%
Spam recall	50	bare	Maj, NN, Ens	98.75%
	100	bare	Maj, NN, Ens	98.54%
	100	lemm	Majority	98.54%
	100	stop	Maj, Ens	98.54%
	150	bare	Maj, NN, Ens	99.17%
	200	bare	Nearest	99.38%
	200	lemm+stop	Nearest	99.38%
Legitimate precision	50	bare	Maj, NN, Ens	99.75%
	100	bare	Maj, NN, Ens	99.71%
	100	lemm	Majority	99.71%
	100	stop	Maj, Ens	99.71%
	150	bare	Maj, NN, Ens	99.83%
	200	bare	Nearest	99.88%
	200	stop	Nearest	99.88%
Legitimate recall	50	lemm	Maj, Ens	99.79%
	100	lemm	NN, Ens	99.79%
	150	lemm	Ensemble	99.79%
	200	bare	Ensemble	99.71%
	200	lemm	Ensemble	99.71%

Compared to the results from [9] and [1], LSI methods provide extremely favorable results, particularly when comparing the recall percentage of the spam and legitimate email documents. Note that in no case is the precision for either spam or legitimate documents 100%, where both [9] and [1] achieved a 100% precision for classified spam email under certain conditions.

Since the spam precision measures the accuracy of the documents classified as spam, it is fair to say that LSI has the potential to

aggressively block some legitimate emails. However, based on these results, it is also fair to say that many more spam emails would be detected.

Table 4 summarizes the above table, showing the optimal (and "ties" within 0.01%) method/corpus/*k* triple for each of the four precision/recall measurements.

Table 4. Summary of top combinations.

Email measure	LSI *k*=	Corpus	Method	Results
Spam precision	50	lemm	Maj, Ens	98.96%
	100	lemm	NN, Ens	98.95%
	150	lemm	Ensemble	98.95%
Spam recall	200	bare	Nearest	99.38%
	200	lemm+stop	Nearest	99.38%
Legitimate precision	200	bare	Nearest	99.88%
	200	stop	Nearest	99.88%
Legitimate recall	50	lemm	Maj, Ens	99.79%
	100	lemm	NN, Ens	99.79%
	150	lemm	Ensemble	99.79%

The results in table 4 show which method, corpus, and number of LSI dimensions should be used to achieve optimal precision and recall for spam or legitimate mail. A user's preferences influence which approach is appropriate, given that a typical assumption that marking spam mail as legitimate is not as critical as marking legitimate mail as spam (represented respectively by the legitimate precision and spam precision measurement). However, if the filter were modified to detect pornographic emails, then optimizing legitimate mail precision (and spam recall) would be more important. Since LSI tends to generalize and approximate semantic relationships, it can suffer from degraded precision ([4]). However, the recall capabilities offer a substantial benefit.

If forced to pick a single method, it would appear that the nearest neighbor or ensemble methods are the best ones to use. Note that for achieving a high spam recall percentage, the majority vote scheme does achieve a very slightly higher percentage (about 0.01% in these tests) than the nearest neighbor or ensembles methods. The LSI dimension is seemingly irrelevant; however k=200 does offer slightly higher benefits under certain conditions. As to the corpus used, the clear implication is that a bare or lemmatized corpus is superior.

If the penalty for misclassifying legitimate documents is fairly low (i.e., they are routed to a bulk mail folder similar to Yahoo!'s offering where the user can review them), then it is probably best to go with the fastest implementation, which is the nearest neighbor method and a bare corpus for legitimate precision. In a real-time scenario with a variable email message size, use a "bare" implementation (not treating the messages) is probably best, since the difference in performance is negligible.

It should be noted that a review of the erroneously classified documents showed a few legitimate and spam emails that were consistently misclassified, no matter the LSI dimension, the classifier (nearest neighbor, majority, or ensemble) or the corpus (bare, lemmatized, stoplist, or lemmatized+stoplist). This could possibly signify that these emails were misclassified in the beginning or appear to be so similar to the other category that it's really a "tossup". Here is an example of one of the legitimate emails from [1], that is consistently considered by the LSI algorithms to be spam:

> Subject: translators needed for women for women !
>
> i am posting the following message for my friends who are not on the list : * would you like to use your language skills to help women survivors of the war in bosnia and croatia ? women for women , a u . s . based , nonprofit sponsorship program sending letters and money each month to the region, is desperately seeking volunteer translators . we translate letters both from and to english . even if you can manage only a handful of letters each month , it would lighten the load for the few translators we have now . for more information , call our office at (703) 519-1730 , and leave a message for zainab or robin . thank you ! * you may also send an e-mail message to me mima @ seur . voa . gov and i will forward it to * women for women : .

While the hand classification performed by [1] has marked this as legitimate, it certainly has a spam-sounding tone, even for a linguistics-based mailing list. Here is one mail consistently marked by the LSI classifiers as legitimate, even though it was hand-classified as spam:

> Subject:
>
> b a r g a i n a i r f a r e s your 1 - stop travel supplier air , hotels , cars, trains , tours , packages * * * call 1-888 - 5-bargain or 202-898 - 7887 for reservations * * * * * * receive a $ 10 discount by referrering to this email * * * roundtrip international airfares : athens frnkfrt london madrid milan munich nice paris rome vienna atlanta $ 819 $ 750 $ 410 $ 745 $ 778 $ 700 $ 730 $ 699 $ 820 $ 845 boston $ 840 $ 760 $ 410 $ 770 $ 749 $ 675 $ 705 $ 620 $ 799 $ 799 chicago $ 935 $ 808 $ 520 $ 720 $ 829 $ 735 $ 766 $ 720 $ 850 $ 820 cincinati $ 999 $ 799 $ 510 $ 745 $ 850 $ 725 $ 765 $ 700 $ 820 $ 810 new york $ 820 $ 760 $ 360 $ 710 $ 799 $ 675 $ 715 $ 385 $ 730 $ 799 philadelphia $ 800 $ 730 $ 410 $ 670 $ 799 $ 658 $ 711 $ 600 $ 789 $ 699 washington $ 800 $ 750 $ 410 $ 695 $ 788 $ 640 $ 699 $ 620 $ 799 $ 740 discounted fares available from every us city to every city world wide ! ! exclusive domestic fares : washington to losangeles . .. . . $ 289 r / t atlanta to seattle . . . . . $ 299 r / t newyork to losangeles . . . . . $ 269 r / t philadelphia to denver . . . . . $ 289 r /t hotel exclusives - daily breakfast and taxes all included : vienna $ 59 frankfurt $ 75 london $ 85 prague $ 75 munich $ 79 manchester $ 95 nice $ 75 athens $65 budapest $ 69 naples $ 75 amsterdam $ 79 warsaw $ 89 geneva $ 79 paris $ 75 dublin $ 99 brussels $ 79 berlin $ 79 florence $ 79 venice $ 85 zurich $ 85 milan $ 79 lisbon $ 69 barcelona $ 75 madrid $ 75 over 8 , 000 hotels available from economy to 5 star deluxe at tremendous savings * * * call 1-888 - 5-bargain or 202-898 - 7887 for reservations * * * * * * receive a $ 10 discount by referrering to this email * * *

Deleted: and

Formatted: Space After: 4 pt

Formatted: Indent: First line: 0", Space After: 4 pt

Deleted: ¶

Formatted: Space After: 4 pt

This is most certainly spam. An analysis of the other emails in the corpus suggests that the large number of cities mentioned in the email probably legitimizes it. However, there are a number of clues that would be detected upon manual classification, such as the empty subject line, the first word broken up by spaces ("b a r g a i n"), and the high proportion of punctuation marks to terms. Additional preprocessing or classification methods could be added to the ensemble to detect these. Note that [9] implements classification based on non-textual characteristics; such characteristics could be included in an LSI implementation.

When comparing the results from each individual corpa (tables 3, 4, 5), it is seen that the majority classifier works better with the spam emails and the nearest neighbor approach works better with legitimate emails. Since there are many, many more legitimate documents (about a 4:1 ratio), the odds are great that a spam document might "appear" to be legitimate when compared with just the single closest document, and the majority count measurement forces the inclusion of more possibilities. Whether or not this holds true for a corpus that is predominately spam is an issue for further study.

As mentioned above, no matter which LSI *k* dimension is used, better results were achieved classifying the spam emails using the bare corpus than with any of the three preprocessed corpa. This lends credence to assertions made by [6] and [7] that morphologic stemming or substitutions are unnecessary when working with latent semantic indexing. However, the lemmatized corpus performed better at isolating the legitimate emails, suggesting that some stemming would be appropriate, as the results with the spam emails in the lemmatized corpus are still excellent.

6. CONCLUSIONS AND FUTURE WORK

Given the experimental data that shows a significant improvement with respect to the classification recall of spam documents, it can be concluded that using latent semantic indexing is a viable method for classifying spam. With the corpus used, using a lemmatizer would prevent some legitimate emails from being misclassified; in general, there is little difference across the board between the preprocessed and "untreated" versions of the corpus.

One obvious drawback to this approach is due to how LSI's design - no new documents can be added to the semantic set without forcing the set to be reconstructed. This is fine if there is a relatively small number of test documents compared to the number of training documents, or if the training set is comprehensive enough to accurately represent the semantic correlations for all of the spam documents that could ever be generated. In a real-world implementation, the LSI semantic set should be flexible enough to handle the ever-increasing set of newly classified (and manually verified) data, and it should also be scalable to handle heavy email traffic. The example of the spam document continually being characterized as legitimate is a prime example of the negative impact a incomplete semantic set can have on documents that have a high proportion of previously undiscovered terms. Previous studies have derived semantic sets using an encyclopedia ([7]), which might serve to be general enough to handle almost any type of document.

Future work in this area should include work to address these issues. Other work could include the adaptation of term-based comparison filtering using LSI.

7. REFERENCES

[1] Androutsopoulos, I. Koutsias, J., Chandrinos, K.V., Paliouras, G., and Spyropoulos, C.D. An Evaluation of Naïve Bayesian Anti-Spam Filtering. In Proceedings of Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, Barcelona, 2000.

[2] Cristiani, N. and Scholkopf, B. "Support Vector Machines and Kernel Methods: The New Generation of Learning Machines". AI Magazine. Fall 2002 (Vol 23, no. 3), pp. 31-41, 2002.

[3] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., K., and Harshman, R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41:391-407,1990.

[4] Gee, K. Text Classification Using Latent Semantic Indexing. Master's thesis. The University of Texas at Arlington, 2001.

[5] Jiang, J. Using Latent Semantic Indexing for Data Mining. Department of Computer Science, University of Tennessee, 1997.

[6] Landauer, T. K. and Dumais, S. T. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240, 1997.

[7] Landauer, T. K., Foltz, P. W., Laham, D. An Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284, 1998.

[8] Laham, D. Latent Semantic Analysis approaches to categorization. In Proceedings of the 19th annual meeting of the Cognitive Science Society, 1997.

[9] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. A Bayesian Approach to Filtering Junk E-Mail. In Learning for Text Categorization - Papers from the AAAI Workshop, pp. 55-62, Madison ,Wisconsin. AAAI Technical Report WS-98-05, 1998.

[10] Sebastiani, F. Machine Learning in Automated Text Categorisation. Technical Report B4-31, Istituto di Elaborazione dell'Informatione, Consiglio Nazionale delle Ricerche, Pisa, 1999. http://faure.iei.pi.cnr.ir/~fabrizio.

[11] The American Heritage® Dictionary of the English Language, Fourth Edition. Houghton Mifflin, 2000. http://www.dictionary.com/search?q=spam