Mining Online Deal Forums for Hot Deals

Wen-tau Yih* Po-hao Chang Wooyoung Kim†
University of Illinois at Urbana-Champaign
201 N. Goodwin Ave, Urbana, IL, 61801 USA
{yih|pchang2|wooyoung}@uiuc.edu

Abstract

Online deal forums are public places where participants share with each other news and information regarding "deals" such as sales promotion events by online stores. The large number of messages in the forums and their inherent uncertainty make it difficult for even seasoned users to identify useful deal *information from the forums. We develop an intelligent deal alert service which assists ordinary Web surfers to find useful deals by mining online deal forums. It periodically crawls relevant deal forums to collect fresh message posts and responses, and evaluate them using a form of probabilistic text classification. Users may be notified of new, "potentially" useful deal messages via emails or they may browse them using their favorite Web browser. We train and evaluate the service using deal posts and responses collected from actual deal forums in the Web. The preliminary evaluation results show that the service is quite effective in reducing the time to find useful deals.*

1. Introduction

"A needle in the haystack" is the phrase that often occurs to us when it comes to information search on the Web. Today's leading search engines usually inundate us with results, leaving us the job of sifting them to find what we really look for. However, there are many cases where this search-and-dump approach may not be good enough. Web forums are one of them.

Web forums are public places in the Web where participants with common special interests meet, discuss and share information. For example, an automania may post a message discussing the off-road performance of an SUV in an automobile forum and initiate a discussion thread, drawing responses from other fellow participants. A response typically contains the author's evaluation of the original post.

Information in such forums not only interests their active participants, but it is useful to other routine Web users (for example, one who has just started shopping for an SUV). The only caveat is that users have to visit these forums and painstakingly comb through the message threads there. Ordinary Web surfers are less likely to do so. Search engines won't help much as blindly searching online forums based on words and phrases does not have sufficient discriminating power to turn up useful messages buried deep inside discussion threads.

What is worse is that not all messages are useful. Since online forums have specific topics, it would look natural at a glance to assume that almost all posts are informative and useful to interested readers. Surprisingly, quite a few posts are not relevant at all. For example, in one forum that we examined, more than 40% of the posts were not useful at all. In some cases, authors uploaded messages onto wrong forums (possibly, by mistake). In other cases, authors provided a piece of information to which a majority of other respondents did not agree. Furthermore, a message being useful is a highly subjective notion; messages useful to some are not to others.

These difficulties warrant development of an intelligent service which automatically sifts through message posts and alerts users with a list of potentially useful messages as they become available. We envision that any such service (or agent) should do the following in one form or another:

- Visit online forums sufficiently frequently and harvest new message posts in a timely fashion.
- Evaluate collected messages for their usefulness and present a user a list of potentially useful ones. The evaluation may be guided by the user's preferences.
- Use the user's feedback to adaptively adjust the message evaluation parameters.
- Allow personalization and customization.

* Supported by NFS grant CAREER IIS-9984168 and an ONR MURI Award.
† Supported in part by the Defense Advanced Research Projects Agency (Contract numbers: F30602-00-2-0586 and F33615-01-C-1097).

This paper reports our experience with implementing the service on deal information forums. The forums, which discuss "hot deals" in the Web, such as sales promotions by online stores, have been chosen for two reasons: one, they are galore, and two, responses found there are well structured and amenable to automated analysis as we explain in the following section. The implementation is based on text classification and machine learning, and has adopted an online learning strategy. Message posts and responses have been collected over a period of time, and used to train the service as well as to evaluate it for its performance.

Related Work Applying machine learning techniques to automated text classification has been extensively studied, for example, in threading email [10], in classifying email into folders [6, 14], in identifying interesting news articles [9], and in developing anti-spam filters [20, 16, 19]. Among these applications, some anti-spam filters (e.g., [16, 19]) share a few properties with our deal finder. For example, they all try to solve a binary text classification problem and are designed to filter out unwanted information for human users.

Particularly related to our work is classifying reviews based on imbued sentiment. The task is to examine a review and identify its sentiment (or, *tone*) as positive or negative, regardless of its topic (i.e., *non-topic based*). Turney [22] hypothesized that adjectives and adverbs reflect the tone of a review. Two-word phrases which have either an adjective or an adverb as one of the two words were extracted from a review, and semantic orientation was calculated for each of them. The review was labeled positive or negative according to the overall cumulative score. Pang et al. [13] applied supervised machine learning techniques to the task of binary classification of movie reviews. However, the text classification based on machine learning has never been applied to mining forums for messages where sentiment in responses may change abruptly as time passes (e.g., expiration of a deal), to the best of our knowledge.

Overview We begin the rest of the paper with analysis of unique characteristics of responses in the deal information forums. The results of the analysis helped us develop effective response evaluation strategies. Section 3 gives an overview of probabilistic text classification employing machine learning techniques that underpins the implementation of the service. Implementation details of the service are presented in Section 4, which is followed by discussion of the performance evaluation methodology and evaluation results in Section 6. We conclude the paper with the brief summary and the future work in Section 7.

2. Automated Mining of Online Deal Forums

Online stores have sales promotion events from time to time. They release coupon codes for discount or provide mail-in rebates for selected items. Various affiliate programs disseminate such information across deal information Web sites (e.g., Ben's Bargain Center [2]) and online forums (e.g., AnandTech's Hot Deals Forum [1]). We aim to mine these forums for useful deals.

At first glance, the online deal forums appear formidable to mine. Fortunately, they have a few characteristic features which make them amenable to automated analysis. First, even though each has a different look-and-feel, these Web sites are mostly homogeneous content-wise. A typical message post for a "deal" contains a price, a link to the online store, coupon codes and/or rebates; these deal messages usually appear as *message posts*. Regular visitors to the forums often leave *responses* behind, which grow into discussion *threads* over time. A response usually contains the writer's opinion on a post.

More importantly, responses can be labeled with one of four types, namely, agreement/disagreement to a post/response, with a few exceptions [1]. Message posts in the online deal forums sometimes contain verifiable information. For example, a post with incorrect information usually attracts far more disagreement responses than agreement ones. Some deal posts have their validity change over time. A post announcing a sales promotion event usually attracts positive responses until the event expires. Even before the expiration date, when the store runs out of stock on the sale item, complaints that the post is outdated or inaccurate start trickling in. In addition, anecdotal properties [2] pertinent to responses may be exploited to quickly identify the type of a response. For example, agreement messages are usually short; by contrast, disagreement messages tend to be longer as most of them carry argument(s) for disagreement as well.

Nonetheless, automated reasoning about the validity and usefulness of a deal post is by no means easy. First, the service must deduce the related information from the textual content of a message, possibly with some cross checking. This inevitably calls for a form of natural language processing. In order to analyze a deal post, the post and its responses must be accessible to the service. To this end, a crawler needs to periodically visit online forums and download messages. A related practical problem is that well established online deal forums have a large user base which collectively produces a large set of messages. What that implies is that a crawler should be scalable and efficient to guarantee timely analysis while it should judiciously control the frequency of its trip to each site so as not to overwhelm the site.

1 Invalid off-topic responses do appear in a discussion thread, although only occasionally.

2 We observed them while we analyzed responses posted in the online deal forums.

The service we develop uses text classification with machine learning at the core of its deduction engine, which we describe in the following section. The discussion on the crawler implementation is dealt in Section 4.

3. Text Classification with Machine Learning

We resort to probabilistic text classification with online learning to analyze deal posts and responses. A separate decision maker summarizes output predictions and provides customized suggestions based on user profiles.

3.1. Text Classification

At the minimum, text classification involves analyzing a text input and assigning it a predefined category. Spam filters (e.g., refer to [19]) and news clipping services (e.g., [16]) are two instances of well known text classifiers.

To guarantee some degree of resilience against unseen texts, text classifiers often borrow techniques from the field of machine learning [12]. The techniques generally offer online adaptability to new input data while keeping development costs in check. For example, in supervised learning settings, input data to train a text classifier with are labeled explicitly and "useful" features are extracted. Then, each data item is represented as a vector in the vector space of the extracted features, and various learning algorithms are applied to the items to train the classifier. In the deal alert service we develop, we use the SNoW machine learning architecture to implement the text classifier.

3.2. The SNoW Machine Learning Architecture [17, 5]

SNoW is a multi-class classifier designed specifically for large-scale learning tasks. Target classes are learned as linear functions over a common feature space, and are represented as a sparse network of the functions. SNoW has several properties which make it an ideal classifier for messages in online deal forums. Among them is *feature efficiency* which SNoW inherits from Winnow [11]; Winnow has been shown quite effective in such domains as text processing where feature space is usually very large, but examples have only a few active features of which a small subset is relevant to a target class. Equally important is SNoW being a memoryless online architecture. When a new example is presented, SNoW can be trained with only the new example and does not require the past examples it was trained with. Being memoryless is a necessary condition for a classifier to be truly *adaptive* and online. Evaluating a deal message for its validity and usefulness is inherently probabilistic as a classifier has to operate in an uncertain information domain. Thus, it is desirable for a classifier to report its classification decision as well as a confidence level associated with the decision. SNoW can be configured to generate both outputs.

3.3. Probabilistic Classification

Many applications use SNoW as a "regular" classifier which, at the end of execution, spits only the final judgment using a "the-winner-takes-it-all" strategy. Instead, we opt for *probabilistic classification* and let the classifier return a confidence value along with the final judgment.

Given a feature vector $\mathbf{x}$, SNoW learns a weight vector $\mathbf{w}$ of the same length for a target class. The *raw activation value* of $\mathbf{x}$ is $\mathbf{w}\cdot\mathbf{x}$ where $\cdot$ is the inner product operator. We define the posterior (or, *confidence*) of a judgment for an example, as a function over the set of raw activation values of target classes. Specifically, we compute the posterior p_i of a class i as `softmax` [4] of i's raw activation value. Let act_i be the class i's raw activation value. Then,

$$p_i = \mathrm{softmax}(act_i) = \frac{e^{act_i}}{\sum_{j=1}^{n} e^{act_j}}$$

where n is the the number of target classes.

In the following section we describe the implementation of the deal alert service in great detail. We deliberately keep the discussion general because the overall system design as well as many of the component services can be adapted to other similar Web services.

4. Implementation

The implementation of the service has a modular architecture. The modular design allows the component services to execute concurrently. In particular, all the component services are loosely-coupled around a shared database. Figure 1 shows the interaction among the component services.

4.1. Database

The database serves as a shared repository; the other component services interact with each other using the database. We use MySQL to implement the database which is accessed by other services through ODBC/JDBC. Seven public tables are defined to facilitate interaction among the component services (Table 1).

4.2. Crawler

The primary services that Crawler provides are to download Web pages from online deal forums and to extract specific fields from the pages. In doing so, it should avoid redundant crawling as much as possible. The crawling service is composed of *forum*, *thread* and *post* crawlers. Each

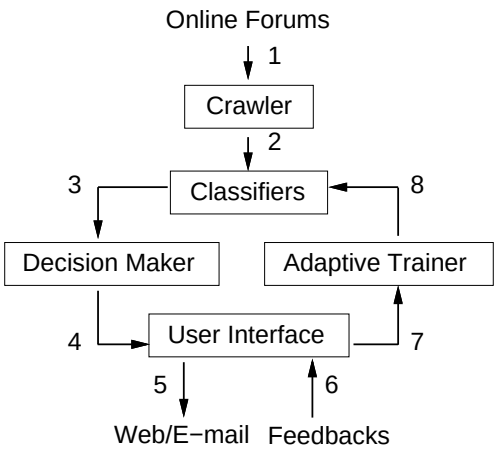

Figure 1. Information flow among the component services: 1. Download posts and responses from online forums. 2. Extract features from messages to classifiers. 3. Analyze posts and responses individually. 4. Combine results from classifiers to make the final decision on a post. 5. Present analyzed information to users. 6. Collect users' feedback. 7. Examine system judgment and users' feedback. 8. Fine-tune classifiers.

crawler has a downloader which fetches HTML pages from Web forums and a parser which parses the pages and extracts relevant information.

The forum crawler identifies newly added and updated threads in a forum since its last visit, and imports them into the `Threads` table (see Table 1) as fresh threads, indicating that they need to be further crawled. The thread crawler looks for newly added/updated messages for each thread listed in the `Threads` table and marks those that require downloading in the `Posts` table. The post crawler looks up the `Posts` table and downloads individual deal messages and responses. The first two crawlers store only the most recent downloaded document. Unlike its siblings, the post crawler keeps all the past downloads of deal posts and responses, since messages may get modified after being posted. These crawlers mostly work independently and concurrently with each other; when necessary, they communicate through the database tables, albeit occasionally.

4.3. Classifiers

Our deal alert service uses *topic* and *sentiment* classifiers. The former classifies posts into `useful` and `useless` classes and the latter classifies responses into `agree` and `disagree` classes. How the classifiers do their feature extraction and classification is further discussed in Section 5.

Name	Content Description
Forums	information related to the target online forums (forum URLs, update rates, required crawlers).
Threads	properties of downloaded threads (number of replies, the last update time, originators).
Posts	individual posts (original posts, following responses). Multiple records for a post may exist when its content is changed.
Evaluation	the predictions generated by the classifiers and judgment by the decision maker.
Feedback	users' feedback on system judgment. Used to train and fine-tune the classifiers.
Classifiers	specific parameters for classifiers. May be modified to alter the classifiers' behaviors.
Users	the profiles of registered users. May be modified to customize the Web service.

Table 1. Database tables for data exchange.

4.4. Decision Maker

The decision making process consists of opinion pooling followed by final judgment. In the opinion pooling, a consensus class distribution of all the responses in a discussion thread is sought. We use `LinOp`, a linear combination of empirical probability distributions [8] (in our case, estimated probabilities from the sentiment classifier), to aggregate opinions. Then, the aggregated opinion is combined with outputs from the topic classifier to produce the final judgment. The output probabilities from the topic classifier and the opinion pooling process are assumed to be independent, and the normalized product of the two values is returned as the final confidence.

4.5. User Interface

The User Interface service serves dual purposes – displaying evaluation results generated by the Decision Maker service and collecting users' feedback on the results. We use PHP to implement the service in two active pages, one for each task. Porting to other platforms, such as JSP and ASP, is straightforward as long as ODBC is supported.

4.6. Adaptive Trainer

A salient feature of our service is its ability to continuously adjust the evaluation parameters as users' feedback comes in, without causing any disruption to the service. The choice of SNoW as the learning algorithm has enabled this dynamic online adaptation. SNoW uses linear functions over a feature space as classifiers. The use of linear functions allows us to represent classifiers as a set of weight values, which are updated according to users' feedback using simple multiplication by a predefined consant.

Note that user feedback should be used differently in training the topic classifier and in training the sentiment classifier. Since the feedback is given for a thread as a whole and for a deal post in particular, we regard it as a reliable response to the post. By contrast, responses do not get any feedback. Currently, we use feedback to a post to reinforce its responses if it is positive one. We naturally expect most responses to a truly useful post would be "agree" messages. Although this assumption sounds too simple-minded, the reinforcement has been quite effective in our experiments.

5. Message Classification and Feature Extraction

5.1. Topic Classifier

Evaluating deal messages for their potential usefulness to a user and classifying them accordingly amount to a traditional binary topic classification. We have built a topic classifier which uses a technique akin to one commonly found in machine learning-based spam filters to categorize deal messages into two classes, *useful* and *useless*.

Obviously, judging the potential usefulness of a post by just looking at the content increases chances of producing incorrect results even for a domain expert. Consulting other reliable information sources is a prudent way to improve the odds for correct classification. Consider a post claiming that a $350 Sony digital camcorder is a deal. A conscious user would visit, for example, DealTime [7] or PriceScan [15]) and cross-check the claim to get more confident on her judgment.

The reliability, however, does not come for free. Performing cross-checking in an automated deal alert service requires more elaborated text processing (e.g., information extraction [18]) and additional semantic analysis. Without them, it is virtually impossible to distinguish, for example, the regular product price from the other numeric values in a message. (We believe the development of the semantic Web [3] should help in this regard.)

Another, somewhat doable, strategy is to get help from domain experts – those who participate in online deal forums and leave opinions on deal posts – by reading follow-up responses to the post. We incorporate this strategy into our service by building a *sentiment classifier* which analyzes responses to deal posts.

5.2. Sentiment Classifier

The sentiment classifier concentrates on deducing the respondent's opinion imbued in her response: *agree* or *disagree*. The finer the resolution gets, the more information the classifier provides, but only with increasing difficulty to implement. In principle, the topic and sentiment classifiers share the same implementation methodology. However, the sentiment classifier is more difficult to develop because it should be able to catch subtle emotional differences from response statements which are, in most instances, short and broken.

5.3. Feature Extraction and Training

A message in an online forum (either a post or a response) has a unique set of fields. `Title`, `author`, `posting time`, and message `body` are among them. Information useful to train classifiers is buried in the fields; it is discovered from the fields in the form of *features* to train the topic and sentiment classifiers. To facilitate feature extraction, words in the `title` and `body` fields are tagged with their corresponding part-of-speech and the message content is preprocessed into a sentence list.

The amount of information harvested from a field varies, depending on the field's data size. For example, the `title` field usually contains a single line composed of only a few words. Thus, we extract trigrams, bigrams, and unigrams of the words and their part-of-speech tags as features. Because the `author` field consists of a word or two, it is used as a feature as a whole. For the `body` field whose size ranges from one to several sentences, we use bigrams and unigrams and part-of-speech tags as features, but not trigrams. An interesting observation we have made in developing the deal alert service is that deal message posts often have a URL pointing to an online store where the item can be purchased. Therefore, we extract whether a message body has URL links as a binary feature, and the corresponding word lists that the URL links demarcate as additional features.

In addition to the above contextual features, we have found that anecdotal "popularity" can be exploited. In general, a useful post draws attention from readers and attracts some responses for some time (e.g., until the deal expires), though no clear correlation can be found between the number of responses and usefulness (Figure 2). However, if a message thread keeps attracting attention, it is highly likely that it is an off-topic thread, such as FAQ and administrative information. Based on this observation, the number of responses is also considered as a feature in training the topic classifier.

In training the sentiment classifier, we use the almost same set of fields to extract features from with a couple of exceptions. First, features from the `title` field are ignored since the messages in a thread have the same title. Second, whether a response has a URL link is ignored, too.

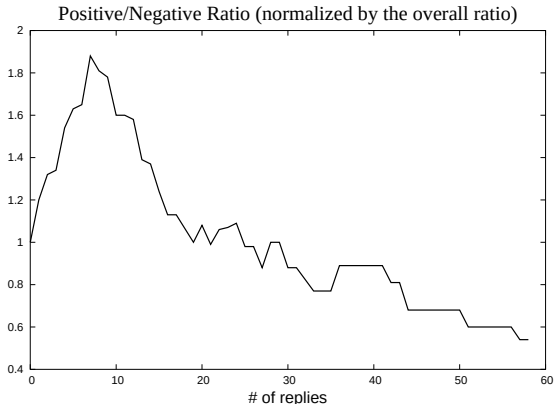

Figure 2. Axis-X is the number of replies; axis-Y is the ratio of positive / negative, normalized by all threads

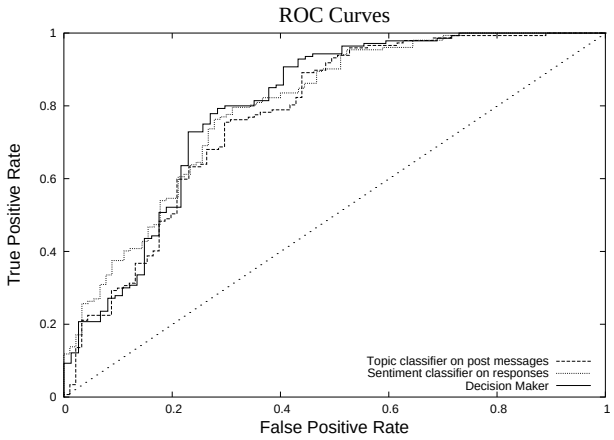

Figure 3. ROC Curves

6. Evaluation

6.1. Methodology

We collected messages from AnandTech's Hot Deals Forum [1] from December 14, 2003 to March 31, 2004 to evaluate our deal alert service. We manually labeled 1220 threads based on deal posts and their responses. A thread was labeled positive when its anchor message provided useful deal information; otherwise, it was labeled negative. Of the 1220 threads, 733 were positive and the rest negative.

The threads were arbitrarily grouped into five sets of an equal size. We conducted experiments by using four sets for training and the remaining one for evaluation. Note that a thread consists of a single anchor post and several trailing responses. Thus, it is reasonable to expect that the label of a thread is the same as that of its anchor post. We used the thread labels in training the topic classifier.

However, the number of responses was too large to manually examine each response and annotate it with its *tone*. Therefore, we assumed the label of a response in a thread was the same as that of the thread. Although the assumption was inherently noisy, we expected most of the training examples would be labeled correctly using this strategy and the overall classification performance be still acceptable.

6.2. Performance Metric

The most common metric to use to evaluate a classifier for performance is *accuracy* or *0/1 loss*. Only when a classifier makes a mistake, it is penalized by 1. Thus, the smaller the total score is, the better a classifier is. Although very simple, the 0/1 loss metric fails to present an accurate picture of a classifier's performance for deal messages. For example, the metric may fool us very easily when an imbalanced data set is given for testing. A detection device achieves 99% accuracy by always predicting that an instance is negative when a test set contains only 1% of positive instances. It also becomes less reliable when the costs of making false positive and false negative predictions are not equal. A user who wants to find as many good deals as possible is ready to accept some false positives, but would be upset with any false negatives. For these reasons, we decided to use the *receiver operating characteristic* (ROC) analysis [21, 23] which has become popular for its robustness in measuring performance of a binary classifier.

The ROC analysis represents the trade-off between the true positive (TP) and false positive (FP) rates of a classifier. The TP rate is the number of positive predictions made divided by the total number of positive instances given. The FP rate is the number of positive predictions made divided by the total number of negative instances given.

In our evaluation, we categorize messages according to a cutoff threshold; a message is labeled positive if the estimated probability is greater than the threshold. When a useful thread is classified as positive using the message classification, it is TP. If a thread is incorrectly labeled positive, it is FP. Consequently, a different threshold will result in a different pair of TP and FP rates (or an operating point). An ROC curve plots these operating points on a 2-D space.

6.3. Evaluation Results

Figure 3 shows ROC curves of the topic and sentiment classifiers and the Decision Maker. Notice that the curves overlap. The overlap indicates that there is no definite winner among the three. However, we observe that the sentiment classifier is slightly better than the topic classifier.

Classifier	Recall	Precision	F_1
Topic classifier	0.959	0.746	0.839
Sentiment classifier	0.954	0.755	0.843
Decision Maker	0.943	0.795	0.863

Table 2. Recall, Precision and F_1; thresholds are selected to maximize F_1's

Also, the Decision Maker performs better in most of the region as it combines the results of the two classifiers.

Table 2 summarizes *Recall* and *Precision* values of the three, which are standard evaluation metrics in the information retrieval community. *Recall* is the number of true positives divided by the number of true positives and false negatives, and *Precision* is the number of true positives divided by the number of all positives predicted by a classifier. F_1 is the harmonic mean of recall and precision values; it is generally accepted that having greater F_1 corresponds to better quality. We again observe the same result that the sentiment classifier is slightly better than the topic classifier and the Decision Maker performs best, although the differences are not that significant. The preliminary experimental results are encouraging as they show the feasibility of our approach.

7. Conclusion

An intelligent Web service which mines online deal forums and notifies users of potentially useful deal messages is presented. The service employs binary text classification based on adaptive online machine learning to label text messages `useful` and `not useful`. We observed in the preliminary experimental results that the service reduced the deal message lookup time for ordinary users considerably. Three avenues for further improvement are being pursued. First, we are exploring ways to incorporate more advanced text classification schemes into the classifiers and the decision maker. It will lead to development of a scheme which uses limited user feedback more effectively. Ingenious heuristics to unearth and exploit domain-specific knowledge will be developed through collection, observation and analysis of actual deal post messages. Lastly, a user-friendly interface to gather more detailed user feedback would help refine personalization of the service.

References

[1] AnandTech Inc. Hot Deals Forum. http://forums.anandtech.com/categories.cfm?catid=40.

[2] Ben's Bargain Center. http://bensbargains.net.

[3] T. Berners-Lee, J. Hendler, and O. Lassilal. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[4] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[5] A. Carleson, C. Cumby, J. Rosen, and D. Roth. The SNoW Learning Architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May 1999.

[6] W. Cohen. Learning Rules that Classify E-Mail. In *Proc. of the AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[7] Deal Time. http://www.dealtime.com/.

[8] A. Garg, T. S. Jayram, S. Vaithyanathan, and H. Zhu. Generalized Opinion Pooling. In *Proceedings of the 8th Intl. Symp. on Artificial Intelligence and Mathematics*, pages 79–86, 2004.

[9] K. Lang. Newsweeder: Learning to Filter Netnews. In *Proc. of the 12 th International Conference on Machine Learning*, pages 331–339, 1995.

[10] D. Lewis and K. A. Knowles. Threading Electronic Mail: a Preliminary Study. *Inf. Process. Manage.*, 33(2):209–217, 1997.

[11] N. Littlestone. Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning*, 2:285–318, 1988.

[12] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[13] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[14] T. Payne and P. Edwards. Interface Agents that Learn: An Investigation of Learning Issues in a Mail Agent Interface. *Applied Artificial Intelligence*, 11(1):1–32, 1997.

[15] Price Scan. http://www.pricescan.com/.

[16] J. Rennie. ifile: An Application of Machine Learning to Mail Filtering. In *Proceedings of the KDD-2000 Workshop on Text Mining*, 2000.

[17] D. Roth. Learning to Resolve Natural Language Ambiguities: A Unified Approach. In *Proc. of AAAI*, pages 806–813, 1998.

[18] D. Roth and W. Yih. Relational Learning via Propositional Algorithms: An Information Extraction Case Study. In *IJCAI*, pages 1257–1263, 2001.

[19] K. Schneider. A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering. In *Proc. 10th Conf. of the European Chapter of the Association for Computational Linguistics*, pages 307–314, 2003.

[20] E. Spertus. Smokey: Automatic Recognition of Hostile Messages. In *Proc. of the 14th National Conference on AI and the 9th Conference on Innovative Applications of AI*, pages 1058–1065, 1997.

[21] J. Swets and R. Pickett. *Evaluation of Diagnostic Systems : Methods from Signal Detection Theory*. Academic Press, 1982.

[22] P. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics*, pages 417 – 424, 2002.

[23] K. Woods and K. Bowyer. Generating ROC Curves for Artificial Neural Networks. *IEEE Transactions on Medical Imaging*, 16(3):329–337, Jun 1997.