

NewsMe: A Case Study for Adaptive News Systems with Open User Model

Chirayu Wongchokprasitti and Peter Brusilovsky
School of Information Sciences, University of Pittsburgh
135 N. Bellefield Avenue
Pittsburgh, PA 15260 USA
chw20@pitt.edu, peterb@pitt.edu

Abstract—Adaptive news systems have become important in recent years. A lot of work has been put into developing these adaptation processes. We describe here an adaptive news system application, which uses an open user model and allow users to manipulate their interest profiles. We also present a study of the system. Our results showed that user profile manipulation should be used with caution.

Index Terms—User profile, personalized news access, open user modeling.

I. INTRODUCTION

The World Wide Web has become a common source that people can access from anywhere at any time. Recently, the continuously updated news content overloads someone trying to keep up with the news. An adaptive web technology would be a good candidate for helping to solve the overload problem.

The main goal of an adaptive news technique is to provide the user access to relevant news content. News personalization systems help the user find relevant news content based on a model of the users interests. The system can recommend or rank the news content, so that relevant content is easier to find. An adaptive news access system, developed by Billsus and Pazzani [1], [2], recommends the most relevant news items for each individual user, coordinated with the users interests and preferences, and has encouraged further work in this area. Their system introduces the use of hybrid user models that consist of separate models for short-term and long-term interests. The evaluation of the system [2] showed impressive results. User modeling and adaptation techniques for personalized news access have been also applied on other systems [3], [4], [5].

Our work brings together research on personalized news access and open user modeling. An adaptive system with an open user model shows the content of the user model to the user, making the adaptive system more transparent to the user. Some systems with open user models also provide a mechanism for users to edit the user model and thus to fine-tune the adaptation process. This gives users more control over the systems performance.

The approach of using open user models has been popular and has provided benefits to educational systems [6], [7], but

very few have been used in the news access area. The obvious difference between the two areas is that education is narrower and each domain is specific to a closed corpus of knowledge. In contrast to this, we find an interesting challenge: adaptive news systems focus broadly on multiple user interests. Also, users often change their interests in current events, while educational systems do not have to follow this fluctuation.

In our work on adaptive news access, we studied the role of an open user model where the system lets users manipulate their own profiles and we explored how user actions would affect system performance.

II. NEWSME: A PERSONALIZED NEWS ACCESS

A. News Presentation

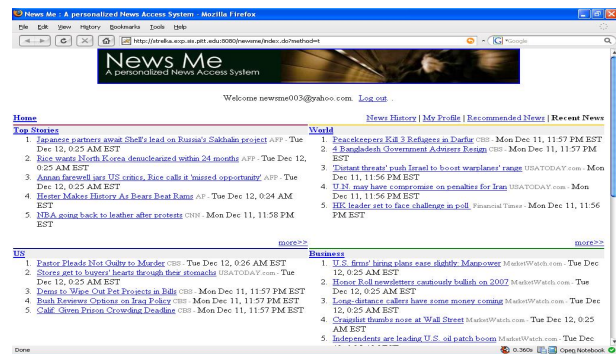


Fig. 1. NewsMe User Interface

NewsMe is a web-based personalized news access system. NewsMe lets users provide feedback about their news-reading interests. Feedback is used to construct user models and to influence how the model recommends relevant news articles to each user. NewsMe retrieves news from 82 RSS news feeds from 21 sources. The news content is categorized into 8 topics and presented to users (as shown in Figure 1). The system maintains separate user models for each topic, which avoid mixing user interests from different areas. The systems crawlers also periodically fetch news article and theirs contents from web news sources. The system then extracts tokens (words or fragments of words) by searching for spaces,

removing stop words and stemming them with the use of the Krovetz stemmer [8] (which produces readable stems unlike more commonly used Porter stemmer [9]). The indexing module creates and stores term frequency vectors.

B. Open user model



Fig. 2. User Feedback Interface

Personalization in NewsMe is made transparent. After reading a news story the user may choose to add it to one of the two lists: "Tracked News" or "Blacklist". Intuitively, the "Tracked News" class label is assigned to news articles that are of special interest to the user. The "Blacklist" class is assigned to news articles rated as the kind of articles users would like to stay away from. Users leave feedback in a frame above the news article (Figure 2). If the feedback is not left, it is assumed that the user has no strong opinion about the article.

The system also provides the ability to manipulate the open user model for past feedback, through the user profile accessible through "My Profile" link. The profile consists of 2 sections: Your Tracked News (Figure 3, upper half) and Your Blacklist News (Figure 3, lower half). The system lets user update their user profile by either removing any article from any list or moving it to another list. The user can also

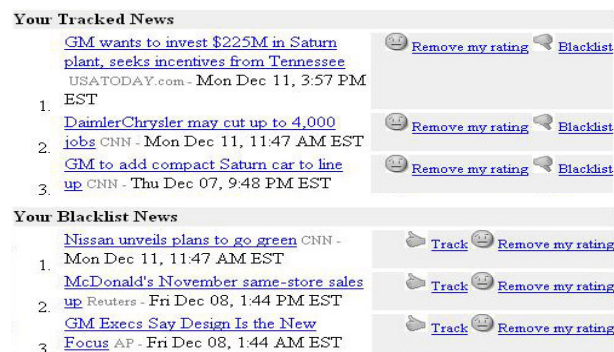


Fig. 3. Tracked News (above) and Blacklist (below) Interface

reconsider earlier decisions using "News History" tab, which lists all the users previously-viewed news stories on a specific topic. Each previously viewed story can be moved to one of the two profiles lists or left unlisted.

C. Learning approach for news access

A classifier built from a large number of training documents that accurately reflect the user's past interests is of limited practical use and may perform substantially worse than a classifier limited to recent data that reflects the user's current

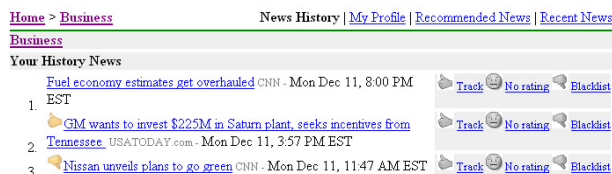


Fig. 4. News History Interface

interests. The model must also be capable of representing the users multiple interests in different topics. And the model must be flexible enough to adapt to a user's changing interests reasonably quickly, even after a previous, long training period. Additionally, users have general news preferences and models can provide good general predictions, in case there's a new story not related to a previously rated event. The user's interests are dynamic; likely to change over time. An assumption of the text classification approach states that more training data leads to improved predictive performance, but this cannot be taken into account. Chiu and Webb [10] studied the utility of including two separate user models in the context of student modeling. Their research showed that data representation and learning algorithms differ significantly from the text classification approach. They assumed that currently collected data would reflect the recent knowledge or preferences of users more accurately than data from previous time periods. However, using only recently collected data models can lead to over specialization, with high precision in the included domains, but poor performance on instances that deviate from the data used to induce the model. They developed a hybrid user model approach that shows significant improvement in prediction rate without significantly affecting prediction accuracy, when compared to the single user model approach. The hybrid model is also applicable to any agent modeling system that constructs models from multiple observations over time. Billsus and Pazzani [1] also used the hybrid model in their adaptive news access system and demonstrated that system performance improved significantly. Other works [11], [12] stated that the use of more than two models contributed no significant advantage over using only two models.

Typically, users tend to track different threads of ongoing recent events, which are tasks requiring short-term information about recent events. The user model should contain information about recently rated events, to identify stories that belong to the same event threads. Also the model should identify stories the users are familiar with. New stories which belong to the same threads of recent information can then be identified. The nearest neighbor algorithm (NN) is selected to achieve this desired functionality. The NN algorithm stores all its training examples in memory. The algorithm compares a new story to all stored instances given the similarity measure and determines the nearest neighbor or the k nearest neighbors. The model converts news stories to term-frequency (TF) vectors, and uses the cosine similarity measure to determine the similarity of two vectors. We also

have two thresholds to classify incoming stories. First, t_{min} is the minimum number used to identify which story is too far from the type of stories the user is interested in. Second, t_{max} is the maximum number used to determine when a story is too close to stories the user has already identified as interesting, which means the new upcoming story is identified as a known story.

In order to classify an unlabeled instance, the algorithm compares it to all stored instances, given some similarity measure, and determines the nearest neighbor or the k nearest neighbors. The main advantage of the NN approach is that only a single topic is needed, in order to allow the algorithm to identify future follow-up news contents from the same thread. In contrast, most other learning algorithms require a large number of training examples to identify a strong pattern. The utility of the NN algorithm has been explored in other text classification applications [13], [14]. To apply the algorithm to natural language text, the system extracts news tokens and converts them to term frequency (TF) vectors, then uses the cosine similarity method to measure the similarity of two vectors [15]. The predicted score is computed by averaging the weighted similarity of a new story with the most recent news stored in the user model, and then multiplying it by the decay function within its own life time (the number of days since the story was retrieved). The general outcome of the experimental comparison between the system used the hybrid modeling and the other one not used is the hybrid model outperforms the other [1].

III. THE STUDY DESIGN

In order to evaluate the open user modeling features, we created a version of NewsMe with the user model manipulation features presented in section 2b disabled. This system used as a control condition, provides the same adaptive news functionalities as in the original one but employs *implicit feedback*: it adds every news story the user reads to the "Tracked News" list. The experimental system in our study was the open model system with *explicit feedback* and *profile manipulation* described in section 2b.

The study has two groups of hypotheses:

Group One: Performance hypotheses are:

H1: The open model system, which allows users to manipulate their user profiles, performs better than the control system without this functionality,

H1.1: The open model system with explicit feedback has better performance, and,

H1.2: Users of the explicit feedback system demonstrate higher task performance.

Group Two: User Perspective hypotheses are:

H2: Users prefer the user profile manipulation features in an open model system,

H2.1: Users appreciate more the system with explicit feedback and user profile manipulation, and,

H2.2: Users appreciate the ability to control their profiles.

In the experiment, participants were assumed to be information analysts. They were asked to analyze news articles related to two specific topics, and collect articles reporting recent important events related to each topic. We chose two topics which occurred during November 28th, 2006 and December 12th, 2006. News articles were frozen for the duration of the study. To allow the development of two independent profiles for the topics of interest, one topic was selected for the US tab and one for the Business tab. For the US topic, the "Iraq Civil War" was selected due to the timing of the developing story about US troops' withdrawal. Users were asked to provide a status report on how people reacted to the Iraqi issue, including the relevant details but ignoring reports from the Bush administration. For the Business topic, the "US auto market" was selected as a search topic to determine whether the slow market affected US automakers and how they tried to boost their sales. Reports were to exclude Japanese auto company stories.

We split the search tasks into two sessions that simulated three stages of access to the data collection, at five-day intervals. The first stage provided news items participants would normally have received between November 28th, 2006 and December 2nd, 2006. The second stage and third stage were December 3rd to 7th, 2006 and December 8th to 12th, 2006, respectively. The experiments actually took place on different days, between December 18th and 29th. The reason we split data into three stages was because the first stage was the period of training for the system while the second stage was when users could update their profile. Without the third stage, we could not compare user behavior with and without user profile manipulation.

We used the Google Notebook extension for the Firefox Web Browser (<http://www.google.com/notebook>) as a tool to let users collect links and passages of news, pretending as though these were search tasks given to them by a supervisor. The collection of links and passages represented their tasks, and was assessed for both precision and recall.

Twenty graduate students from the University of Pittsburgh participated in the experiment. They were familiar with the information search task, but none was specifically interested in either of the study topics before the study. Each subject worked with both systems and both topics. Participants were randomly assigned a system for a specific topic in the first session, and switched to another system for the second topic. This design allowed for a direct comparison of the two systems with the same samples. We provided a brief description of the system and tasks at the beginning of each session. Each session consisted of one search task (one on each topic), with two brief post-questionnaires at the end of the second stage and the third stage. Participants had 10 minutes to search and extract relevant information for each stage. Each session took about one hour.

IV. RESULT AND DISCUSSION

A. The Ground Truth

We adopted precision as our measuring tool to evaluate system and user performance. In order to calculate precision and recall, we established the "ground truth" for each topic by manually annotating all news articles. Each article was tagged on a 2-point scale (0=irrelevant, 1= relevant). Table I presents a summary of the annotated news articles. Here, "Relevant" means that the corresponding article matches well with the task description. We considered results statistically significant if they passed the t-test at the 5% level ($p\text{-value} \leq 0.05$).

TABLE I
SUMMARY OF ANNOTATED NEWS ITEMS IN THE STUDY

	US		Business	
	Relevant	News	Relevant	News
Stage 1	34	611	28	933
Stage 2	44	569	15	902
Stage 3	42	591	16	862
Total	120	1771	59	2697

B. System Performance Analysis

In this section we present the results of comparing the performance of the experimental and control versions of the NewsMe system. We calculated precision by comparing the news items in the recommended news tab with each category in Table I. We took all relevant news stories for each topic into account for calculating recall. We did not show the system performance on the stage 1 because it was the training stage and the list of news items were chronologically sorted but not ranked. Since web users typically pay the most attention to the top ranking results, adaptive news systems should place their most relevant items at the top of the recommended list, especially in the first screen (the top 20, approximately). To measure the recommendation performance of NewsMe, we calculated precision for the top ranking 60.

The recommended news list in the NewsMe system contains not only the recent interests of the users but also shows the known news list, so that we can take known news into account when calculating precision for the first screen. The top 60 news items in the recommended news and the top 60 in the known news list (in case the system found new items that were too similar to previous news items) were collected and verified with Table I. Figure 5 and 6 show the precision of the top ranked 60 news articles for both systems on the US and Business topics, respectively. The solid blue line represents precision of the open model system with explicit feedback and the red dashed line indicates the precision of the open model system with implicit feedback. In Figure 5 and 6, neither system seems to outperform the other one ($p\text{-value} < 0.05$).

Since the main difference between the two systems is the ability to manipulate tracked and blacklist items in the user model, we investigated the news items manipulation frequency with the open user model and compared it with

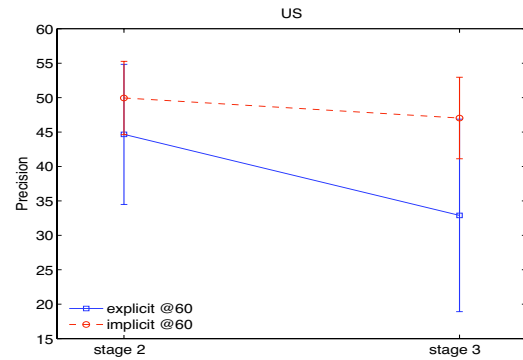


Fig. 5. Precision @ top 60 on US topic.

individually determined system precision. We found that news item manipulation frequency has some correlation with system performance. Only four users did manipulation in stage 2, one on the US topic and three on the Business topic (Table II). Out of the only two made considerable changes (User 1 in US topic and User 2 in Business). In both cases user model manipulation resulted in the worst system performance for the topic.

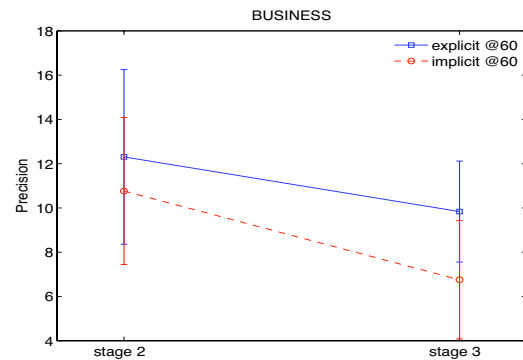


Fig. 6. Precision @ top 60 on Business topic.

In Table III, we also found the same results for stage 3 where only 3 users manipulated their user models, 1 on the US topic and 2 on the Business topic. In the US topic, the system precision of User 1, who made many changes, was the lowest when compared with other users' who did not do manipulation. Also on the Business topic, system performance of User 2 was worse than the performance for other users. The system performance for User 3, who only one manipulations, was about the same as the others. We found that negative relationships existed for both topics.

Figures 7 and 8 hints that user performance (measured as the precision of user selection) is generally higher for the open model system with explicit feedback. However, t-test found no significant difference between precision values of both systems ($p\text{-value} > 0.05$).

TABLE II
USER MANIPULATION VS. SYSTEM PERFORMANCE AT STAGE 2

Topic	User	Chages	Precision
US	17	0	51.62
	13	0	50.55
	9	0	43.14
	4	0	41.88
	16	0	41.67
	12	0	39.74
	8	0	38.96
	20	0	38.57
	5	0	31.00
	1	5	10.45
Business	3	1	16.50
	11	0	16.00
	7	0	13.17
	10	0	13.00
	6	1	12.00
	15	0	11.67
	19	0	10.00
	14	0	9.00
	18	0	5.50
	2	8	4.61

TABLE III
USER MANIPULATION VS. SYSTEM PERFORMANCE AT STAGE 3

Topic	User	Chages	Precision
US	12	0	46.19
	9	0	41.52
	13	0	38.74
	8	0	37.98
	4	0	37.61
	20	0	33.69
	16	0	31.52
	17	0	6.01
	5	0	4.92
	1	3	4.36
Business	7	0	14.33
	6	0	9.83
	10	0	9.33
	15	0	8.50
	3	1	8.50
	11	0	8.50
	18	0	7.67
	14	0	7.17
	19	0	5.83
	2	5	5.50

C. User feedback analysis

Participants were given a post-questionnaire to investigate their satisfaction with the system (Table IV **) after each search task, except for stage 1, which was the training stage. For all questions, they were asked to rate their level of agreement from 1 (Not at all) to 5 (Extremely). For both systems, they were given questions to rank topic familiarity, sufficiency of news, trust of system, control of system, and overall satisfaction. For only the open model system, users were asked to rate the utility of the user model controls for adding news to the tracked news list, blacklist list, and the displaying of terms.

A two-way ANOVA was performed on the questionnaire data, to examine significant differences in user answers by system and by stage. Tables 5 and 6 show the mean responses for each topic, by stage and system, respectively, with overall means reported. Pairs of data with significant differences are indicated by an asterisk. As shown in Table V, Question 7, subjects indicated that the My Profile tab helps them to

**indicates open model with explicit feedback only

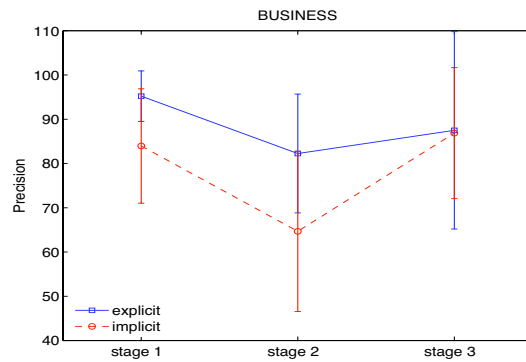


Fig. 7. User Performance on Business topic.

TABLE IV
POST-STAGE 2 AND 3 QUESTIONNAIRE QUESTIONS

1. Were you familiar with this topic before the search?
2. Did the system provide you with sufficient news for your task?
3. Were you confident in the system's ability to find useful information on this topic?
** 4. Did you feel you had enough control over how the system recommended news items?
** 5. Did you find that adding news to Tracked News was useful in helping the system find useful news items for this topic?
** 6. Did you find that adding news to Blacklist was useful in helping the system find useful news items for this topic?
** 7. Does displaying news in my profile help you understand how the system finds useful news items for this topic?
8. Does displaying news in news history help you understand how the system finds useful news items for this topic?
9. Does displaying news in recommended news help you understand how the system finds useful news items for this topic?
10. Did news reports in the known news list in recommended news reflect that you have known those news reports for this topic before? (Skip this question if you did not see known news list in the recommended news)
11. Overall, did you have a positive experience with this system?

understand how the system finds useful news items for the US topic versus the Business topic, with an overall p-value of 0.013.

As shown in Table 6, Question 3, subjects indicated they trusted more in the system's ability to find useful information for the US topic versus the Business topic (p-value 0.017).

TABLE V
MEAN POST-QUESTIONNAIRE RESPONSE TO QUESTIONS WITH
SIGNIFICANT DIFFERENCES BETWEEN TOPICS AND STAGES (*P ≤ 0.05)

Question	Stage	US	Business
4	2	3.40	3.50
	3	3.40	3.20
	Overall	3.40	3.35
5	2	4.00	3.40
	3	3.90	3.20
	Overall	3.95	3.30
6	2	3.60	3.20
	3	3.20	3.00
	Overall	3.40	3.10
7	2	4.00	3.60
	3	4.20	3.20
	Overall	*4.10	*3.40

TABLE VI
MEAN POST-QUESTIONNAIRE RESPONSE TO QUESTIONS WITH
SIGNIFICANT BETWEEN TOPICS AND SYSTEMS (*P ≤ 0.05)

Question	Stage	US	Business
1	Explicit FB.	2.90	2.20
	Implicit FB.	2.00	2.40
	Overall	2.45	2.30
2	Explicit FB.	3.65	3.35
	Implicit FB.	3.70	3.55
	Overall	3.68	3.45
3	Explicit FB.	3.75	3.30
	Implicit FB.	3.80	3.30
	Overall	*3.78	*3.30
4	Explicit FB.	3.40	3.35
	Implicit FB.	N/A	N/A
	Overall	N/A	N/A
5	Explicit FB.	3.95	3.30
	Implicit FB.	N/A	N/A
	Overall	N/A	N/A
6	Explicit FB.	3.40	3.10
	Implicit FB.	N/A	N/A
	Overall	N/A	N/A
7	Explicit FB.	*4.10	*3.40
	Implicit FB.	N/A	N/A
	Overall	N/A	N/A
8	Explicit FB.	3.80	3.50
	Implicit FB.	3.30	3.20
	Overall	3.55	3.35
9	Explicit FB.	4.00	3.80
	Implicit FB.	3.90	3.85
	Overall	3.95	3.83
10	Explicit FB.	3.80	3.50
	Implicit FB.	3.30	3.20
	Overall	3.55	3.35
11	Explicit FB.	3.85	3.40
	Implicit FB.	3.75	3.70
	Overall	3.80	3.55

V. DISCUSSION AND FUTURE WORK

At the beginning of this study, we expected that the explicit feedback with user profile manipulation approach would outperform the implicit feedback approach. But from our experimental results, we found that the experimental system did not perform significantly better than the implicit feedback system. This is, in fact, good. It indicates that the implicit feedback system is efficient enough to match the explicit feedback system, because users tend to read relevant news stories and avoid irrelevant ones.

With extensive features providing the ability to manipulate the user profile, we found the results to be different than what we expected. We noticed that user model manipulation affected system performance. System performance for users who made few changes to the user profile was similar to that for users

who made no changes to the user profile. But when many changes were made to the user profile, system performance became much worse. This result indicates that without caution, user model manipulation can lower performance.

REFERENCES

- [1] D. Billsus and M. Pazzani, *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Germany: Springer-Verlag, 2007.
- [2] —, "User Modeling for Adaptive News Access," *User Modeling and User-Adapted Interaction*, vol. 10, no. 2-3, pp. 147-180, June 2000.
- [3] L. Ardissono, L. Console, and I. Torre, "An Adaptive System for The Personalized Access to News," *AI Communications*, vol. 14, pp. 129-147, November 2001.
- [4] A. Diaz and P. Gervas, "Personalization in News Delivery Systems: Item Summarization and Multi-tier Item Selection using Relevance Feedback," *Web Intelligence and Agent Systems*, vol. 3, no. 3, pp. 135-154, November 2005.
- [5] B. Magnini and C. Strapparava, "User Modeling for News Web Sites with Word Sense Based Techniques," *User Modeling and User-Adapted Interaction*, vol. 14, no. 2-3, pp. 239-257, June 2004.
- [6] S. Bull, "Supporting Learning with Open Learner Models," in *Proc. Hellenic Conference on Information and Communication Technologies in Education*, Athens, Greece, 2004, pp. 47-61.
- [7] R. Denaux, V. Dimitrova, and L. Aroyo, "Integrating Open User Modeling and Learning Content Management for The Semantic Web," in *Proc. User Modeling 2005*, August 2005, pp. 9-18.
- [8] R. Krovetz, "Viewing Morphology as an Inference Process," in *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 1993, pp. 191-202.
- [9] M. Porter, "An Algorithm for Suffix Stripping," *Program*, vol. 14, no. 3, pp. 130-137, July 1980.
- [10] B. C. Chiu and G. I. Webb, "Using Decision Trees for Agent Modeling: Improving Prediction Performance," *User Modeling and User-Adapted Interaction*, vol. 8, no. 1-2, pp. 131-152, March 1998.
- [11] W. Cohen and H. Hirsh, "Joins That Generalize: Text Classification using WHIRL," in *Proc. International Conference on Knowledge Discovery and Data Mining*, New York, NY, 1998, pp. 169-173.
- [12] K. M. Ting and B. T. Low, "Model Combination in the Multiple-Dataset Scenario," in *ECML '97: Proceedings of the 9th European Conference on Machine Learning*. London, UK: Springer-Verlag, 1997, pp. 250-265.
- [13] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic Detection and Tracking Pilot Study Final Report," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, 1998, pp. 194-218.
- [14] P. K. Chan and S. J. Stolfo, "A Comparative Evaluation of Voting and Meta-Learning on Partitioned Data," in *Proc. International Conference on Machine Learning*, 1995, pp. 90-98.
- [15] R. Belew, *Finding Out About*. Cambridge, UK: Cambridge University Press, 2000.