Towards Modeling Threaded Discussions using Induced Ontology Knowledge

Donghui Feng Jihie Kim Erin Shaw Eduard Hovy

Information Sciences Institute
University of Southern California
Marina del Rey, CA, 90292
{donghui, jihie, shaw, hovy}@isi.edu

Abstract

Online discussion boards are a popular form of web-based computer-mediated communication, especially in the areas of distributed education and customer support. Automatic analysis for discussion understanding would enable better information assessment and assistance. This paper describes an extensive study of the relationship between individual messages and full discussion threads. We present a new approach to classifying discussions using a Rocchio-style classifier with little cost for data labeling. In place of a labeled data set, we employ a coarse domain ontology that is automatically induced from a canonical text in a novel way and use it to build discussion topic profiles. We describe a new classify-by-dominance strategy for classifying discussion threads and demonstrate that in the presence of noise it can perform better than the standard classify-as-a-whole approach with an error rate reduction of 16.8%. This analysis of human conversation via online discussions provides a basis for the development of future information extraction and question answering techniques.

Motivation

Within the past decade, the explosive growth of the Web has resulted in an enormous repository of knowledge and created an unprecedented opportunity for data analysis. Online discussion boards have become a popular form of web-based computer-mediated communication, especially in the areas of distributed education and customer support. Discussion board participants use forums to collaborate, exchange information and seek answers to problems. In web-enhanced courses, discussion boards are heavily used for question answering and collaborative problem solving (Soller & Lesgold, 2003; Cakir et al., 2005).

Past approaches to mining information from discussion board text focused mainly on finding answers to questions (Marom & Zukerman, 2005; Feng et al., 2006a). Most of these techniques consider discussion data as a simple corpus of text. However, there are increasing needs for modeling discussion activities more explicitly. In web-enhanced courses, instructors would like to review student discussions so as to understand the kinds of contributions students make and to determine if they are progressing, or need assistance or guidance (Painter et al., 2003). In managing discussion boards, generally, we would like to identify undesirable distractions, such as contributions that are unrelated to the main focus. To support this kind of assessment, we must be able to track topics of discussion and determine if contributions are focused and productive.

To support these capabilities, we have developed several techniques for modeling discussion threads. A discussion thread consists of a set of ordered messages contributed by two or more participants to a correspondence. Thus, we can consider online discussion a special case of human conversation (Levinson, 1983). In the past, we modeled discussion threads as a sequence of speech acts and investigated dependencies among the messages using a set of relational dialogue rules (Feng et al., 2006a; Feng et al., 2006b). In this paper, we systematically analyze the contents of a message, the relations among the messages in a thread, and the relations between each message and the thread to which it belongs. The analysis is based on the focus of a discussion thread, or *topic focus*, which takes into account the topics of the individual messages of the thread and relates them to the topic of the full thread.

Existing efforts in text mining and topic analysis in the natural language and information retrieval communities relate mainly to flat document sets or to news or story streams (e.g. Gabrilovich et al., 2004; Shen et al., 2005); there is limited work on thread-based analysis. In addition, most existing topic identification algorithms require a set of examples to learn dominant lexical features for classifying text sets using supervised learning.

In this paper, we present a novel approach for classifying threaded discussions with little cost for data labeling. In place of a labeled data set, we employ a coarse domain ontology that we automatically induce from a canonical text and use it to build discussion topic profiles. In particular, we analyze discussions from a computer science course using a Rocchio-style classifier (Rocchio, 1971). The course textbook is used as the canonical text.

We also describe a new classify-by-dominance strategy for classifying discussion threads using a classification of individual messages, and compare it with the standard classify-as-a-whole approach where each full thread is

taken as a document. The results show that classify-by-dominance can work better when individual messages contain a lot of noise.

The contributions of this paper are two-fold. First, we study how individual messages in threaded discussions are related to the focus of the overall discussion thread. We propose a novel classification strategy based on the classification of individual messages and evaluate it using a Rocchio-style classifier. In particular, we identify several different situations where the new classification strategy works better than standard approaches. Second, we propose an approach for deriving topic profile vectors from a domain ontology and show how this ontology is automatically derived from a canonical text.

The paper begins with our methodology for modeling and analyzing online discussions, including how we build a Rocchio-style classifier and induce an ontology from a canonical text and the details of the two classifying strategies, classify-by-dominance vs. classify-as-a-whole. We next report results from an empirical analysis of discussion threads, and highlight the differences between the two approaches. We conclude with our contributions and directions for future work.

Modeling Approach

As a first step towards modeling discussion threads, we want to identify topics discussed in threaded discussions and assess whether the topics shift or remain focused within the threads. Most machine learning approaches to topic classification use supervised learning techniques that require a set of manually labeled data, and the classifiers are trained with a selected learning algorithm, such as Naïve Bayes or SVM (Support Vector Machine). In most cases, manually labeling data is time consuming and expensive. Although there is research that proposes bootstrapping from limited data or explores the use of unlabeled data (e.g. Nigam et al., 2000; Raskutti et al., 2002), the need for a sufficient number of label examples remains a major bottleneck.

Furthermore, in an online discussion forum, the cost of labeling data may be higher for several reasons. First, the total number of topics and the volume of messages are usually large, and the annotation of training examples for each topic is difficult and can easily become ad hoc. This results in inconsistent annotations and noisy training examples. Second, messages in online forums are typically posted in chronological order, so it is not guaranteed that positive training examples for all topics exist in the corpus at the time of the annotation, and training for topics with sparse data is not possible. To overcome the lack of labeled data and reduce human annotation cost, we apply a Rocchio-style classifier and derive topic profile vectors using automatic ontology induction, as described below.

Rocchio-Style Classification

Rocchio-style classifications have been used in the field of Information Retrieval (IR) for topic detection and tracking (e.g. Allan, 1996; Callan, 1998; Fiscus & Wheatley. 2004). We adopt the idea of using relevance feedback from ad hoc information filtering research. Rocchio-style classifiers compute a *profile* vector for each text category, or topic, as a weighted average of positive and negative training examples, and can work well when the training set is small and a large data set is unavailable. An adaptive version of the Rocchio algorithm is described in Equation 1

$$\vec{T} = \alpha\vec{T} + \beta\frac{\sum_{\vec{d}\in R}\vec{d}}{|R|} - \gamma\frac{\sum_{\vec{d}\in NR}\vec{d}}{|NR|} \quad (1)$$

where $\vec{T}$ is the profile vector of each topic, $\vec{d}$ represents the new message vector, *R* refers to the positive example set (relevant), and *NR* refers to the negative example set (not-relevant). The parameters (alpha, beta, gamma) are typically set to 1.

Theoretically, a human judge might check a new message and provide relevance feedback. If the feedback is positive, the new document is added to the relevant set, *R*, otherwise, it is added to the not-relevant set, *NR*, and the topic profile vector is updated. When relevance feedback is not available, an assumption is made that the top *M* classification results can be viewed as positive examples. This is called pseudo-relevance feedback and is typically conducted via a threshold-learning module. If the classification score of the new message, based on cosine similarity, exceeds a particular threshold, it is considered a positive example and added to the relevant message set, *R*. Learning classification parameters using this approach require a separate validation set with correct labels, which is often expensive. To minimize the cost, we simplify the construction of the classifier by using the basic version of Rocchio without adaptation.

In practice, the basic classification decision is made based on the cosine similarity between each topic profile vector and the new message vector. Suppose $\vec{T}$ is the profile vector of each topic and $\vec{m}$ represents the new message vector. Then classification after normalization can be determined using Equation 2, where *M* is the total number of unique words appearing in the corpus.

$$\begin{aligned}\hat{T} &= \arg\max_{T}(\cos\_sim(\vec{T},\vec{m})) \\ &= \arg\max_{T}(\sum_{k=1}^{M} w_{T_k} * w_{m_k})\end{aligned} \quad (2)$$

Each message vector contains a set of vector elements based on TF or TF*IDF transformations. Each topic can have one representative vector. The classification of each message can be decided by calculating the lexical similarity score between topic vectors and message vectors using Equation 2.

A key challenge for most Rocchio algorithms is constructing an initial profile vector before starting to classify new messages. Proposed techniques typically derive such a vector from a seed training set with positive

and/or negative examples. We describe how we derive a topic profile vector in the next section.

Ontology Induction

A predefined common vocabulary benefits knowledge sharing and reuse among AI systems. An ontology refers to a specification of a representational vocabulary for a shared domain of discourse-definitions of classes, relations, functions, and other objects (Gruber, 1993). Although it has become clear that it is almost impossible to build an adequate ontology, people have come up with several approaches for the reliable construction of ontological knowledge (Hovy, 2005). Domain ontologies are used in IR and NLP (Natural Language Processing) for question answering (Hermjakob et al., 2002), assessing document similarity (Murray et al., 2005), and topic identification (Lin 1997). However, manual construction of typological knowledge is expensive and error-prone (Denny, 2002).

To reduce the cost of manual knowledge acquisition, a canonical text of the domain can be used. For modeling student discussions in a university course, the textbook for the course can be used as the canonical text. We automatically induce a domain ontology by processing a portion of the textbook and deriving the topic profile vectors from the induced ontology. An ontology is typically represented as a hierarchy of concepts and relations that exist among the objects in the domain. The idea is to induce a coarse hierarchy of domain topics and represent each topic with a vector using the terms that represent the topic.

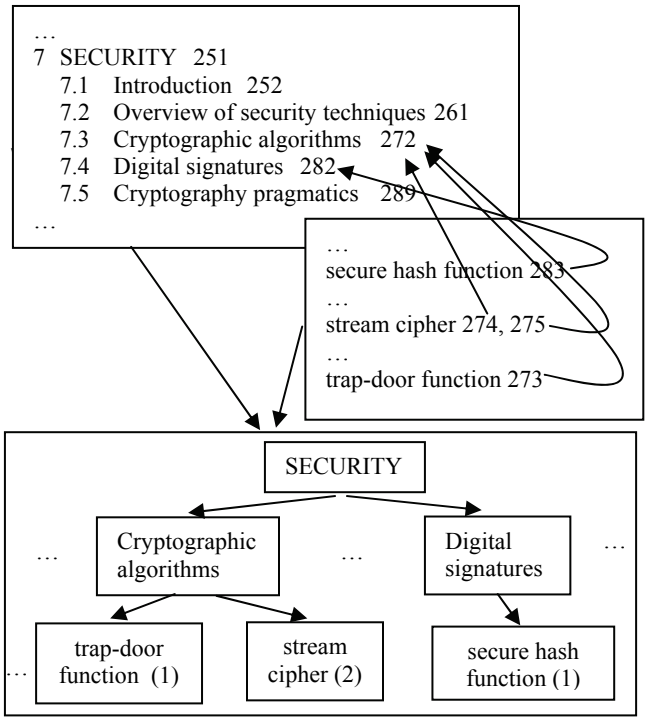

Figure 1. An example of ontology induction.

The system learns and induces ontology knowledge by processing the table of contents and the index of the textbook. The table of contents contains the main sections and sub-sections of the book. The index contains a list of terms and the hierarchical relations among the terms (i.e., some terms represent general categories while other terms are specific to a category). The index terms can be related to the main section topics by page number. The table of contents and the textbook index are automatically scanned and processed. The relationships between section topics and terms, and relationships within terms (via the index hierarchy) are preserved during the procedure. From this, we build a coarse domain ontology with a hierarchical tree structure of terms. Unlike other domain ontology, each term in our ontology is associated with a frequency score, which represents the total number of references to it in the text. Using this, we compute the TF value for the terms. That is, we turn the whole textbook into a training set.

Figure 1 gives an example of how the ontology is induced with this procedure. We used a textbook from a graduate-level operating systems course. From the table of contents, we extracted 131 topics or categories, consisting of 961 different words. From the index, we induced a total of 1427 technical concepts and 2183 different words.

Using the main themes of the course syllabus, we identified six main topics for the course. Each topic may include one or more sections in the textbook and therefore corresponds to one or more sub-trees in the induced ontology. The sub-trees are then transformed into vectors that represent the topic profiles. Our classifier uses the resulting topic profile vectors.

To identify the topic focus of a discussion thread that consists of a sequence of messages, we investigate classification strategies that make use of relations between individual messages and the full thread. Next, we propose a new classify-by-dominance strategy and compare it with the standard classify-as-a-whole strategy.

Classify-by-Dominance vs. Classify-as-a-Whole

Traditional topic classifications of documents may use a news or story stream to perform topic detection and information filtering where an individual news item or story is considered a classification unit. There has been limited research on the automatic classification of threaded discussions. A full discussion thread is like a human conversation in that it consists of a set of messages with inherent relationships. This makes the task of classifying a discussion thread unique. We investigate the relationship between individual messages in a thread and the full thread by classifying messages and threads with pre-selected topics.

For message classification, we take the standard bag-of-words approach to transform each message into a vector representation, and follow the Rocchio classification. For thread classification, we consider several strategies. The first one is to simply view all the messages in a thread as in a whole document and transform all of them into one vector. The Rocchio classifier is then applied to this vector.

We refer to this classification strategy as *classify-as-a-whole*. Another strategy is to classify the full thread based on the dominant topics in the thread. We refer to this as *classify-by-dominance*. The dominance can be defined with several different criteria, for example, a) the maximum number of times a topic is mentioned in all messages, or b) the nearest topic based on the voting from individual messages by their classification scores or c) the nearest topic based on the message with the largest classification score. Or it can be defined by whether the topic of a thread is dominated by d) the first message that initiates the whole thread, or e) the closing message of the whole thread, or f) the longest continual sequence of the whole thread. We call a classification based on any of the above criteria a classify-by-dominance approach.

The classification of a discussion thread contains two phases. Initially, all of the messages in a thread are classified with Rocchio but the classification of the thread is determined by the dominant criteria. Suppose in a thread S we have n messages, $m_1, m_2, \ldots, m_n$. We apply Rocchio to each message and get the classifications $T_1, T_2, \ldots, T_n$ with the classification scores $TS_1, TS_2, \ldots, TS_n$. The classification decision, $\hat{T}$, of the thread can be made with the following formulas, each one corresponding to one of the dominance criteria:

$$\hat{T}(S) = \arg\max_T count(T_i = T) \quad (3)$$

$$\hat{T}(S) = \arg\max_T sum(TS_i = T) \quad (4)$$

$$\hat{T}(S) = \arg\max_T (TS_i) \quad (5)$$

$$\hat{T}(S) = T_1 \quad (6)$$

$$\hat{T}(S) = T_n \quad (7)$$

$$\hat{T}(S) = \arg\max_T LCS(TS_i = T) \quad (8)$$

We compare the classification results with different settings and investigate how messages and threads interact within a thread. For simplicity, we represent the criteria as D_COUNT, D_SUM, D_NEAREST, D_INIT, D_CLOSING, and D_LCS respectively. The results are reported in the following section.

Empirical Results

Before presenting our classification results, we will describe the discussion corpus in greater detail, including the degree of coherence of each thread. The corpus consists of discussions from one semester of a USC graduate course in Advanced Operating Systems.

Thread length	1	2	3	4	5	6	8	9	10	12	16
Number of threads	5	17	7	5	2	7	1	2	2	1	1

Table 1. Number of threads per length in corpus.

The data set contains 206 messages and 50 threads. The average length of a thread is 4.12 messages. Table 1 shows the number of threads by length in the corpus.

As described previously, no training data is required for classification. However, in this particular course, the topics of the discussion forum followed the syllabus and the students had to choose one of the categories within which to initiate a discussion thread. We use these topic choices as the gold standard for our analysis. The six topics are 1) Communication Models, 2) Distributed Concurrency, 3) Naming and Binding, 4) Security, 5) File Systems, and 6) Case Studies. The topics correspond to one or more sub-trees in our ontology. The distribution of messages over the topics according to our best classifier (as described below) is shown in Figure 2. Most of the messages are classified into topics 1 and 5, with relatively few in topic 6.

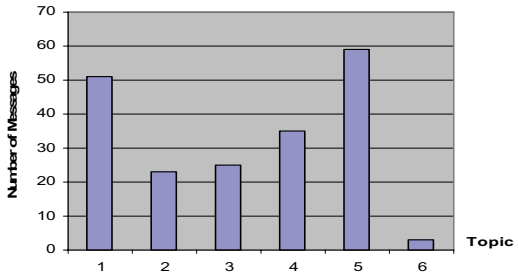

Figure 2. Number of messages per topic.

Figure 3 shows topic distribution changes over time. Each time period in the x axis represents a bi-week. The changes in the topic focus closely match the syllabus. When the instructor begins a new topic, discussions on that topic dominate the discussion forum. Contributions include discussions on technical concepts, projects and assignments.

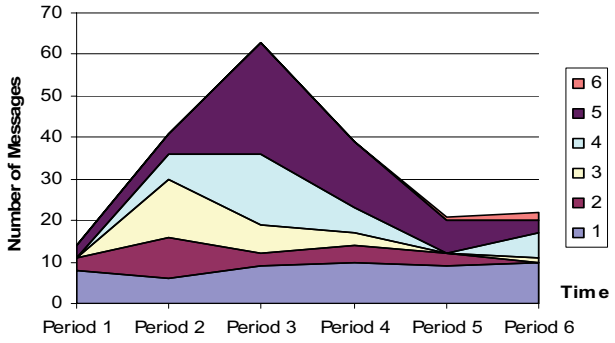

Figure 3. Temporal nature of discussion topics.

When we investigated the topic shifts within each thread, we found more variances. Some discussions are very coherent while others have varying topics. Figure 4 shows the topic shifts for three sample discussion threads. In Thread 4711, all the messages focus on the same topic, while Thread 4716 has only one message that leaves the main topic. Thread 4830 shows several topic changes.

Message 5 in Thread 4830 is classified as 'others' because it did not contain any terms defined in our ontology. These messages contain courtesy words or acknowledgements such as 'Thank you' or 'It makes sense'.

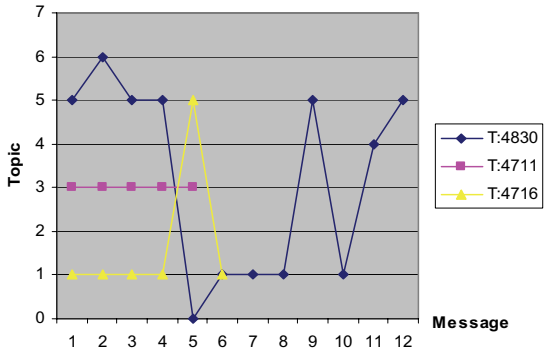

Figure 4. Topic shifting within threads.

We examined the coherence of each thread using the coherence measure

$$coherence = \frac{\max(\text{frequency of a topic})}{\#\text{ of messages}} \quad (9)$$

Using the coherence scores, we classified the threads into three categories: low, medium, and high, corresponding to the coherence intervals [0, 0.4), [0.4, 0.8), and [0.8, 1], respectively. As shown in Figure 5, most of the threads fall into the high and medium intervals. The results from these analyses can be used for information extraction or retrieval. For example, in retrieving answers to a question from a discussion corpus, we can remove irrelevant information and identify a coherent set of data to answer the question.

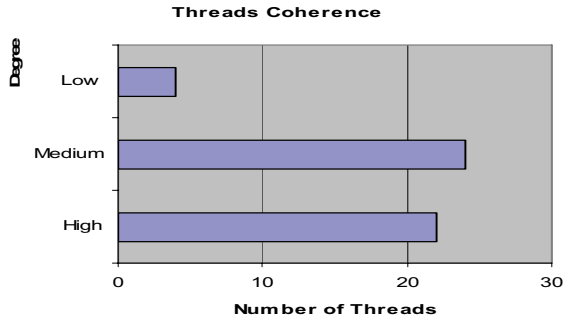

Figure 5. Thread coherence.

For classification accuracy, we used the following measure

$$accuracy = \frac{\#\text{ of correctly classified threads}}{\#\text{ of threads}} \quad (10)$$

For the vector representation, we use TF*IDF to compute vector elements. Both classify-as-a-whole and classify-by-dominance strategies use the same Rocchio-style classifier with the same induced ontology. Messages are stemmed before classification. Table 2 shows both the accuracy scores for all the threads and the scores for the threads with low or medium coherence. As shown in the table, most strategies work similarly for high coherence threads. However, D_SUM outperforms the classify-as-a-whole approach on the data set with lower coherence (low+medium) by 7.2 percent. The accuracies are 0.643 and 0.571 respectively. The error rate reduction is 16.8%. It appears that voting by classification scores can dominate the messages that are less relevant and noisier in the threads by considering individual cosine similarities together instead of the cosine similarity of a whole thread vector in the vector space.

The D_INIT and D_CLOSING results are relatively worse as they only consider one message in the full thread, and the coherence of the whole thread is not always guaranteed. However, they might provide a good indication of the real topic for the whole thread, and are potentially useful for quickly identifying the topic by processing only one message.

Approach		Accuracy	
		All threads	Low and medium coherence threads
Classify-as-a-whole		0.68	0.571
Classify-by-dominance	D_COUNT	0.66	0.536
	D_SUM	**0.72**	**0.643**
	D_NEAREST	0.68	0.607
	D_INIT	0.62	0.464
	D_CLOSING	0.64	0.571
	D_LCS	0.64	0.5

Table 2. Classification accuracy of threads.

Related Work

Online discussions have attracted attention in several research communities. Zhou and Hovy (2005) proposed summarizing discussion threads using multiple document summarization techniques but did not consider the relationship between messages and threads. In our previous work (Feng et al., 2006a), we implemented a discussion-bot for answering student queries in threaded discussions. To extract the most useful/correct information, we applied a rule-based extraction algorithm using boundary nodes that are defined with respect to speech act annotations and assumed the boundaries could produce coherent answers. Our thread analysis approach can help determine topic changes within each thread to find better extraction boundaries.

Improved topic classification and text categorization models are the focus of much research (e.g. Joachims, 1997; Liu et al., 2004; Yang et al., 2005). Knowledge-based topic classification approaches use individual documents from a news or story stream as a classification unit (Lin, 1997; D'Alessio et al., 1998). Studies of the relations between individual messages topics and thread topics are limited. Studies that focus on collaborative

interactions in online environments include managing continuous dialogue for personal assistants (Nguyen & Wobcke 2005), scaffolding conversations in a virtual meeting space (Isbister et al., 2000), and collaborative learning (Soller & Lesgold 2003). Our work provides a complementary capability by providing details about the topics used when interacting and collaborating.

Rocchio-style classification is used widely in IR and information filtering, and much of the research focuses on parameter learning for Rocchio (Allan, 1996; Callan, 1998; Zhang, 2004; Yang et al., 2005). These learning mechanisms may improve our system performance. We are also interested in studying the balance between human labeling costs and system performance.

Conclusions

This study focuses on identifying topics of threaded discussions in online discussion boards. To overcome the absence of labeled data, we used a Rocchio-style classifier, which is a high-bias but low-variance classifier. Rocchio plays an important role in bridging the analysis of message and thread topics based on the analysis of their text terms. To build a topic classifier, we automatically induced a coarse domain ontology from a canonical text. We explored several classification strategies that make use of relations between individual messages and threads. The results show that the new classify-by-dominance strategy can work better when individual messages contain noise. We believe a better construction and refinement of both the ontology and the topic-subtree mapping could improve the system performance and the assessment of discussion threads. We are investigating an approach that makes use of dependencies among the messages by combining speech acts analysis and topic classification.

Acknowledgements

This work was supported in part by DARPA grant DOI-NBC Contract No. NBCHC050051, *Learning by Reading*, and in part by a grant from the Lord Corporation Foundation to the USC Distance Education Network. The authors want to thank Chin-Yew Lin, Feng Pan, and Rahul Bhagat for their helpful suggestions on this work.

References

Allan, J. 1996. Incremental relevance feedback for information filtering. In *Proceedings of SIGIR-96*.

Cakir, M., Xhafa, F., Zhou, N., and Stahl, G. 2005. Thread-based analysis of patterns of collaborative interaction in chat. In *Proceedings of AIED-2005*.

Callan, J. 1998. Learning while filtering documents. In *Proceedings of SIGIR-98*.

D'Alessio, S., Murray, K., Kershenbaum, A., and Schiaffino, R. 1998. The Effect of Topological Structure on Hierarchical Text Categorization. In *Very Large Corpora Workshop at COLING-ACL '98*.

Denny, M. 2002. *Ontology Building: A Survey of Editing Tools*.

Feng, D., Shaw, E., Kim, J., and Hovy, E.H. 2006a. An Intelligent Discussion-Bot for Answering Student Queries in Threaded Discussions. In *Proceedings of IUI 2006*.

Feng, D., Shaw, E., Kim, J., and Hovy, E.H. 2006b. Learning to Detect Conversation Focus of Threaded Discussions. In *Proceedings of HLT-NAACL 2006*.

Fiscus, J. and Wheatley, B. 2004. Overview of the TDT 2004 Evaluation and Results. In *TDT-2004*.

Gabrilovich, E., Dumais, S., and Horvitz, E. 2004. NewsJunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty. In *WWW-2004*.

Gruber, T. R. 1993. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220.

Hermjakob, U., Hovy, E.H., and Lin, C-Y. 2002. Automated Question Answering in Webclopedia - A Demonstration. In *Proceedings of ACL-02*.

Hovy, E.H. 2005. Methodologies for the Reliable Construction of Ontological Knowledge. In *Proceedings of ICCS 2005*.

Isbister, K. Nakanishi, H., Ishida, T., Nass, C. 2000. Helper agent: Designing an assistant for human-human interaction in a virtual meeting space. In *Proceeding of CHI'2000*.

Joachims, T. 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In *Proceedings of ICML-1997*.

Levinson, S. 1983. *Pragmatics*. Cambridge University Press.

Lin, Chin-Yew. 1997. *Robust Automated Topic Identification*. Ph.D. Thesis. University of Southern California, August 1997.

Liu, B. Li, X., Lee, W-S, and Yu, P.S. 2004. Text Classification by Labeling Words. In *AAAI-2004*.

Marom, Y. and Zukerman, I. 2005. Corpus-based Generation of Easy Help-desk Responses. *Technical Report*, School of Computer Science and Software Engineering, Monash University.

Murray, K., Lowrance, J., Appelt, D., and Rodriguez, A., 2005, Estimating similarity among collaboration contributions. In *Proceedings of K-CAP 2005*.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. 2000. Text Classification from Labeled and Unlabelled Document using EM. *Machine Learning*, Vol. 39, No. 2/3, pp. 103-134.

Nguyen, A. and Wobcke, W. 2005. An Agent-Based Approach to Dialogue Management in Personal Assistants. In *Proceedings of IUI-2005*.

Painter, C., Coffin, C., and Hewings, A. 2003. Impacts of Directed Tutorial Activities in Computer Conferencing: A Case Study. *Distance Education*, Vol. 24, No. 2.

Raskutti, B., Kowalczyk, A., and Ferra, H. 2002. Combining Clustering and Co-training to Enhance Text Classification Using Unlabelled Data. In *Proceedings of SIGKDD*.

Rocchio, J.J. 1971. Relevance Feedback in Information Retrieval. In G. Salton, *The SMART Retrieval System: Experiments in Automatic Doc. Processing*, pp. 313-323.

Shen, X., Dumais, S., and Horvitz, E. 2005. Analysis of topic dynamics in web search. In *Proceedings of WWW-2005*.

Soller, A., & Lesgold, A. 2003. A Computational Approach to Analyzing Online Knowledge Sharing Interaction. In *Proceedings of AIED-2003*.

Yang, Y.,Yoo, S., Zhang, J., and Kisiel, B. 2005. Robustness of adaptive filtering methods in a cross-benchmark evaluation. In *Proceedings of SIGIR-2003*.

Zhang, Y. 2004. Using Bayesian priors to combine classifiers for adaptive filtering. In *Proceedings of ACM SIGIR-2004*.

Zhou, L. and Hovy, E.H. 2005. Digesting virtual "geek" culture: the summarization of technical internet relay chats. In *Proceedings of ACL 2005*.