Learning to Detect Conversation Focus of Threaded Discussions

Donghui Feng   **Erin Shaw**   **Jihie Kim**   **Eduard Hovy**

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

Abstract

In this paper we present a novel feature-enriched approach that learns to detect the conversation focus of threaded discussions by combining NLP analysis and IR techniques. Using the graph-based algorithm HITS, we integrate different features such as lexical similarity, poster trustworthiness, and speech act analysis of human conversations with feature-oriented link generation functions. It is the first quantitative study to analyze human conversation focus in the context of online discussions that takes into account heterogeneous sources of evidence. Experimental results using a threaded discussion corpus from an undergraduate class show that it achieves significant performance improvements compared with the baseline system.

1 Introduction

Threaded discussion is popular in virtual cyber communities and has applications in areas such as customer support, community development, interactive reporting (blogging) and education. Discussion threads can be considered a special case of human conversation, and since we have huge repositories of such discussion, automatic and/or semi-automatic analysis would greatly improve the navigation and processing of the information.

A discussion thread consists of a set of messages arranged in chronological order. One of the main challenges in the Question Answering domain is how to extract the most informative or important message in the sequence for the purpose of answering the initial question, which we refer to as the *conversation focus* in this paper. For example, people may repeatedly discuss similar questions in a discussion forum and so it is highly desirable to detect previous conversation focuses in order to automatically answer queries (Feng et al., 2006).

Human conversation focus is a hard NLP (Natural Language Processing) problem in general because people may frequently switch topics in a real conversation. The threaded discussions make the problem manageable because people typically focus on a limited set of issues within a thread of a discussion. Current IR (Information Retrieval) techniques are based on keyword similarity measures and do not consider some features that are important for analyzing threaded discussions. As a result, a typical IR system may return a ranked list of messages based on keyword queries even if, within the context of a discussion, this may not be useful or correct.

Threaded discussion is a special case of human conversation, where people may express their ideas, elaborate arguments, and answer others' questions; many of these aspects are unexplored by traditional IR techniques. First, messages in threaded discussions are not a flat document set, which is a common assumption for most IR systems. Due to the flexibility and special characteristics involved in human conversations, messages within a thread are not necessarily of equal importance. The real relationships may differ from the analysis based on keyword similarity measures, e.g., if a 2nd message "corrects" a 1st one, the 2nd message is probably more important than the 1st. IR systems may give different results. Second, messages posted by different users may have different degrees of correctness and trustworthiness, which we refer to as *poster trustworthiness* in this paper. For instance, a domain expert is likely to be more reliable than a layman on the domain topic.

In this paper we present a novel feature-enriched approach that learns to detect conversation focus of threaded discussions by combining NLP analysis and IR techniques. Using the graph-based algorithm HITS (Hyperlink Induced Topic Search, Kleinberg, 1999), we conduct discussion analysis taking into account different features, such as lexical similarity, poster trustworthiness, and speech act relations in human conversations. We generate a weighted threaded discussion graph by applying feature-oriented link generation functions. All the features are quantified and integrated as part of the weight of graph edges. In this way, both quantitative features and qualitative features are combined to analyze human conversations, specifically in the format of online discussions.

To date, it is the first quantitative study to analyze human conversation that focuses on threaded discussions by taking into account heterogeneous evidence from different sources. The study described here addresses the problem of conversation focus, especially for extracting the best answer to a particular question, in the context of an online discussion board used by students in an undergraduate computer science course. Different features are studied and compared when applying our approach to discussion analysis. Experimental results show that performance improvements are significant compared with the baseline system.

The remainder of this paper is organized as follows: We discuss related work in Section 2. Section 3 presents thread representation and the weighted HITS algorithm. Section 4 details feature-oriented link generation functions. Comparative experimental results and analysis are given in Section 5. We discuss future work in Section 6.

2   Related Work

Human conversation refers to situations where two or more participants freely alternate in speaking (Levinson, 1983). What makes threaded discussions unique is that users participate asynchronously and in writing. We model human conversation as a set of messages in a threaded discussion using a graph-based algorithm.

Graph-based algorithms are widely applied in link analysis and for web searching in the IR community. Two of the most prominent algorithms are Page-Rank (Brin and Page, 1998) and the HITS algorithm (Kleinberg, 1999). Although they were initially proposed for analyzing web pages, they proved useful for investigating and ranking structured objects. Inspired by the idea of graph based algorithms to collectively rank and select the best candidate, research efforts in the natural language community have applied graph-based approaches on keyword selection (Mihalcea and Tarau, 2004), text summarization (Erkan and Radev, 2004; Mihalcea, 2004), word sense disambiguation (Mihalcea et al., 2004; Mihalcea, 2005), sentiment analysis (Pang and Lee, 2004), and sentence retrieval for question answering (Otterbacher et al., 2005). However, until now there has not been any published work on its application to human conversation analysis specifically in the format of threaded discussions. In this paper, we focus on using HITS to detect conversation focus of threaded discussions.

Rhetorical Structure Theory (Mann and Thomson, 1988) based discourse processing has attracted much attention with successful applications in sentence compression and summarization. Most of the current work on discourse processing focuses on sentence-level text organization (Soricut and Marcu, 2003) or the intermediate step (Sporleder and Lapata, 2005). Analyzing and utilizing discourse information at a higher level, e.g., at the paragraph level, still remains a challenge to the natural language community. In our work, we utilize the discourse information at a message level.

Zhou and Hovy (2005) proposed summarizing threaded discussions in a similar fashion to multi-document summarization; but then their work does not take into account the relative importance of different messages in a thread. Marom and Zukerman (2005) generated help-desk responses using clustering techniques, but their corpus is composed of only two-party, two-turn, conversation pairs, which precludes the need to determine relative importance as in a multi-ply conversation.

In our previous work (Feng et al., 2006), we implemented a discussion-bot to automatically answer student queries in a threaded discussion but extract potential answers (the most informative message) using a rule-based traverse algorithm that is not optimal for selecting a best answer; thus, the result may contain redundant or incorrect information. We argue that pragmatic knowledge like speech acts is important in conversation focus analysis. However, estimated speech act labeling between messages is not sufficient for detecting

human conversation focus without considering other features like author information. Carvalho and Cohen (2005) describe a dependency-network based collective classification method to classify email speech acts. Our work on conversation focus detection can be viewed as an immediate step following automatic speech act labeling on discussion threads using similar collective classification approaches.

We next discuss our approach to detect conversation focus using the graph-based algorithm HITS by taking into account heterogeneous features.

3 Conversation Focus Detection

In threaded discussions, people participate in a conversation by posting messages. Our goal is to be able to detect which message in a thread contains the most important information, i.e., the *focus* of the conversation. Unlike traditional IR systems, which return a ranked list of messages from a flat document set, our task must take into account characteristics of threaded discussions.

First, messages play certain roles and are related to each other *by a conversation context*. Second, messages written by different authors may *vary in value*. Finally, since postings occur in parallel, by various people, message threads are not necessarily coherent so the *lexical similarity* among the messages should be analyzed. To detect the focus of conversation, we integrate a pragmatics study of conversational speech acts, an analysis of message values based on poster trustworthiness and an analysis of lexical similarity. The subsystems that determine these three sources of evidence comprise the features of our feature-based system.

Because each discussion thread is naturally represented by a directed graph, where each message is represented by a node in the graph, we can apply a graph-based algorithm to integrate these sources and detect the focus of conversation.

3.1 Thread Representation

A discussion thread consists of a set of messages posted in chronological order. Suppose that each message is represented by m_i, $i =1,2,..., n$. Then the entire thread is a directed graph that can be represented by $G= (V, E)$, where V is the set of nodes (messages), $V= \{m_i, i=1,...,n\}$, and E is the set of directed edges. In our approach, the set V is automatically constructed as each message joins in the discussion. E is a subset of VxV. We will discuss the feature-oriented link generation functions that construct the set E in Section 4.

We make use of speech act relations in generating the links. Once a speech act relation is identified between two messages, links will be generated using generation functions described in next section. When m_i is a message node in the thread graph, $F(m_i) \subset V$ represents the set of nodes that node m_i points to (i.e., children of m_i), and $B(m_i) \subset V$ represents the set of nodes that point to m_i (i.e., parents of m_i).

3.2 Graph-Based Ranking Algorithm: HITS

Graph-based algorithms can rank a set of objects in a collective way and the affect between each pair can be propagated into the whole graph iteratively. Here, we use a weighted HITS (Kleinberg, 1999) algorithm to conduct message ranking.

Kleinberg (1999) initially proposed the graph-based algorithm HITS for ranking a set of web pages. Here, we adjust the algorithm for the task of ranking a set of messages in a threaded discussion. In this algorithm, each message in the graph can be represented by two identity scores, *hub score* and *authority score*. The hub score represents the quality of the message as a pointer to valuable or useful messages (or resources, in general). The authority score measures the quality of the message as a resource itself. The weighted iterative updating computations are shown in Equations 1 and 2.

$$hub^{r+1}(m_i) = \sum_{m_j \in F(m_i)} w_{ij} * authority^r(m_j) \qquad (1)$$

$$authority^{r+1}(m_i) = \sum_{m_j \in B(m_i)} w_{ji} * hub^r(m_j) \qquad (2)$$

where r and $r+1$ are the numbers of iterations.

The number of iterations required for HITS to converge depends on the initialization value for each message node and the complexity of the graph. Graph links can be induced with extra knowledge (e.g. Kurland and Lee, 2005). To help integrate our heterogeneous sources of evidence with our graph-based HITS algorithm, we introduce link generation functions for each of the three features, (g_i, $i=1, 2, 3$), to add links between messages.

4 Feature-Oriented Link Generation

Conversation structures have received a lot of attention in the linguistic research community (Levinson, 1983). In order to integrate conversational features into our computational model, we must convert a qualitative analysis into quantitative scores. For conversation analysis, we adopted the theory of Speech Acts proposed by (Austin, 1962; Searle, 1969) and defined a set of speech acts (SAs) that relate every pair of messages in the corpus. Though a pair of messages may only be labeled with one speech act, a message can have multiple SAs with other messages.

We group speech acts by function into three categories, as shown in Figure 1. Messages may involve a request (REQ), provide information (INF), or fall into the category of interpersonal (INTP) relationship. Categories can be further divided into several single speech acts.
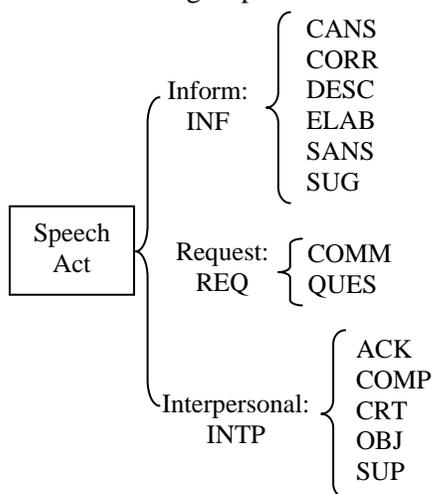

Figure 1. Categories of Message Speech Act.

The SA set for our corpus is given in Table 1. A speech act may a represent a positive, negative or neutral response to a previous message depending on its attitude and recommendation. We classify each speech act as a direction as POSITIVE (+), NEGATIVE (−) or NEUTRAL, referred to as *SA Direction*, as shown in the right column of Table 1.

The features we wish to include in our approach are lexical similarity between messages, poster trustworthiness, and speech act labels between message pairs in our discussion corpus.

The feature-oriented link generation is conducted in two steps. First, our approach examines in turn all the speech act relations in each thread and generates two types of links based on lexical similarity and SA strength scores. Second, the sys-

tem iterates over all the message nodes and assigns each node a self-pointing link associated with its poster trustworthiness score. The three features are integrated into the thread graph accordingly by the feature-oriented link generation functions. Multiple links with the same start and end points are combined into one.

Speech Act	Name	Description	Dir.
ACK	Acknowl-edge	Confirm or acknowledge	+
CANS	Complex Answer	Give answer requiring a full description of procedures, reasons, etc.	
COMM	Command	Command or announce	
COMP	Compli-ment	Praise an argument or suggestion	+
CORR	Correct	Correct a wrong answer or solution	−
CRT	Criticize	Criticize an argument	−
DESC	Describe	Describe a fact or situation	
ELAB	Elaborate	Elaborate on a previous argument or question	
OBJ	Object	Object to an argument or suggestion	−
QUES	Question	Ask question about a specific problem	
SANS	Simple Answer	Answer with a short phrase or few words (e.g. factoid, yes/no)	
SUG	Suggest	Give advice or suggest a solution	
SUP	Support	Support an argument or suggestion	+

Table 1. Types of message speech acts in corpus.

4.1 Lexical Similarity

Discussions are constructed as people express ideas, opinions, and thoughts, so that the text itself contains information about what is being discussed. Lexical similarity is an important measure for distinguishing relationships between message pairs. In our approach, we do not compute the lexical similarity of any arbitrary pair of messages, instead, we consider only message pairs that are present in the speech act set. The cosine similarity between each message pair is computed using the TF*IDF technique (Salton, 1989).

Messages with similar words are more likely to be semantically-related. This information is represented by term frequency (TF). However, those

211

with more general terms may be unintentionally biased when only TF is considered so Inverse Document Frequency (IDF) is introduced to mitigate the bias. The lexical similarity score can be calculated using their cosine similarity.

$$W^l = \cos\_sim(m_i, m_j) \qquad (3)$$

For a given a speech act, $SA_{ij}(m_i \rightarrow m_j)$, connecting message m_i and m_j, the link generation function g_1 is defined as follows:

$$g_1(SA_{ij}) = arc_{ij}(W^l) \qquad (4)$$

The new generated link is added to the thread graph connecting message node m_i and m_j with a weight of W^l.

4.2 Poster Trustworthiness

Messages posted by different people may have different degrees of trustworthiness. For example, students who contributed to our corpus did not seem to provide messages of equal value. To determine the trustworthiness of a person, we studied the responses to their messages throughout the entire corpus. We used the percentage of POSITIVE responses to a person's messages to measure that person's trustworthiness. In our case, POSITIVE responses, which are defined above, included SUP, COMP, and ACK. In addition, if a person's message closed a discussion, we rated it POSITIVE.

Suppose the poster is represented by $person_k$, the poster score, W^p, is a weight calculated by

$$W^p(person_k) = \frac{count(positive\_feedback(person_k))}{count(feedback(person_k))}$$

$$\qquad (5)$$

For a given single speech act, $SA_{ij}(m_i \rightarrow m_j)$, the poster score indicates the importance of message m_i by itself and the generation function is given by

$$g_2(SA_{ij}) = arc_{ii}(W^p) \qquad (6)$$

The generated link is self-pointing, and contains the strength of the poster information.

4.3 Speech Act Analysis

We compute the strength of each speech act in a generative way, based on the author and trustworthiness of the author. The strength of a speech act is a weighted average over all authors.

$$W^s(SA) = sign(dir) \sum_{person_k} \frac{count(SA_{person_k})}{count(SA)} W^P(person_k) \, (7)$$

where the sign function of *direction* is defined with Equation 8.

$$sign(dir) = \begin{cases} -1 & \text{if } dir \text{ is NEGATIVE} \\ 1 & \text{Otherwise} \end{cases} \qquad (8)$$

All SA scores are computed using Equation 7 and projected to [0, 1]. For a given speech act, $SA_{ij}(m_i \rightarrow m_j)$, the generation function will generate a weighted link in the thread graph as expressed in Equation 9.

$$g_3(SA_{ij}) = \begin{cases} arc_{ii}(W^s) & \text{if } SA_{ij} \text{ is NEUTRAL} \\ arc_{ij}(W^s) & \text{Otherwise} \end{cases} \qquad (9)$$

The SA scores represent the strength of the relationship between the messages. Depending on the direction of the SA, the generated link will either go from message m_i to m_j or from message m_i to m_i (i.e., to itself). If the *SA* is NEUTRAL, the link will point to itself and the score is a recommendation to itself. Otherwise, the link connects two different messages and represents the recommendation degree of the parent to the child message.

5 Experiments

5.1 Experimental Setup

We tested our conversation-focus detection approach using a corpus of threaded discussions from three semesters of a USC undergraduate course in computer science. The corpus includes a total of 640 threads consisting of 2214 messages, where a thread is defined as an exchange containing at least two messages.

Length of thread	Number of threads
3	139
4	74
5	47
6	30
7	13
8	11

Table 2. Thread length distribution.

From the complete corpus, we selected only threads with lengths of greater than two and less than nine (messages). Discussion threads with lengths of only two would bias the random guess of our baseline system, while discussion threads with lengths greater than eight make up only 3.7% of the total number of threads (640), and are the least coherent of the threads due to topic-switching and off-topic remarks. Thus, our evaluation corpus included 314 threads, consisting of 1307 messages, with an average thread length of 4.16 messages per

212

thread. Table 2 gives the distribution of the lengths of the threads.

The input of our system requires the identification of speech act relations between messages. Collective classification approaches, similar to the dependency-network based approach that Carvalho and Cohen (2005) used to classify email speech acts, might also be applied to discussion threads. However, as the paper is about investigating how an SA analysis, along with other features, can benefit conversation focus detection, so as to avoid error propagation from speech act labeling to subsequent processing, we used manually-annotated SA relationships for our analysis.

Code	Frequency	Percentage (%)
ACK	53	3.96
CANS	224	16.73
COMM	8	0.6
COMP	7	0.52
CORR	20	1.49
CRT	23	1.72
DESC	71	5.3
ELAB	105	7.84
OBJ	21	1.57
QUES	450	33.61
SANS	23	1.72
SUG	264	19.72
SUP	70	5.23

Table 3. Frequency of speech acts.

The corpus contains 1339 speech acts. Table 3 gives the frequencies and percentages of speech acts found in the data set. Each SA generates feature-oriented weighted links in the threaded graph accordingly as discussed previously.

Number of best answers	Number of threads
1	250
2	56
3	5
4	3

Table 4. Gold standard length distribution.

We then read each thread and choose the message that contained the best answer to the initial query as the gold standard. If there are multiple best-answer messages, all of them will be ranked as best, i.e., chosen for the top position. For example, different authors may have provided sugges-

tions that were each correct for a specified situation. Table 4 gives the statistics of the numbers of correct messages of our gold standard.

We experimented with further segmenting the messages so as to narrow down the best-answer text, under the assumption that long messages probably include some less-than-useful information. We applied TextTiling (Hearst, 1994) to segment the messages, which is the technique used by Zhou and Hovy (2005) to summarize discussions. For our corpus, though, the ratio of segments to messages was only 1.03, which indicates that our messages are relatively short and coherent, and that segmenting them would not provide additional benefits.

5.2 Baseline System

To compare the effectiveness of our approach with different features, we designed a baseline system that uses a random guess approach. Given a discussion thread, the baseline system randomly selects the most important message. The result was evaluated against the gold standard. The performance comparisons of the baseline system and other feature-induced approaches are presented next.

5.3 Result Analysis and Discussion

We conducted extensive experiments to investigate the performance of our approach with different combinations of features. As we discussed in Section 4.2, each poster acquires a trustworthiness score based on their behavior via an analysis of the whole corpus. Table 5 is a sample list of some posters with their poster id, the total number of responses (to their messages), the total number of positive responses, and their poster scores W^p.

Poster ID	Total Response	Positive Response	W^p
193	1	1	1
93	20	18	0.9
38	15	12	0.8
80	8	6	0.75
47	253	182	0.719
22	3	2	0.667
44	9	6	0.667
91	6	4	0.667
147	12	8	0.667
32	10	6	0.6
190	9	5	0.556
97	20	11	0.55
12	2	1	0.5

Table 5. Sample poster scores.

Based on the poster scores, we computed the strength score of each SA with Equation 7 and projected them to [0, 1]. Table 6 shows the strength scores for all of the SAs. Each SA has a different strength score and those in the NEGATIVE category have smaller ones (weaker recommendation).

SA	$W^s(SA)$	SA	$W^s(SA)$
CANS	0.8134	COMM	0.6534
DESC	0.7166	ELAB	0.7202
SANS	0.8281	SUG	0.8032
QUES	0.6230		
ACK	0.6844	COMP	0.8081
SUP	0.8057		
CORR	0.2543	CRT	0.1339
OBJ	0.2405		

Table 6. SA strength scores.

We tested the graph-based HITS algorithm with different feature combinations and set the error rate to be 0.0001 to get the algorithm to converge. In our experiments, we computed the precision score and the MRR (Mean Reciprocal Rank) score (Voorhees, 2001) of the most informative message chosen (the first, if there was more than one). Table 7 shows the performance scores for the system with different feature combinations. The performance of the baseline system is shown at the top.

The HITS algorithm assigns both a hub score and an authority score to each message node, resulting in two sets of results. Scores in the HITS_ AUTHORITY rows of Table 7 represent the results using authority scores, while HITS_HUB rows represent the results using hub scores.

Due to the limitation of thread length, the lower bound of the MRR score is 0.263. As shown in the table, a random guess baseline system can get a precision of 27.71% and a MRR score of 0.539.

When we consider only lexical similarity, the result is not so good, which supports the notion that in human conversation context is often more important than text at a surface level. When we consider poster and lexical score together, the performance improves. As expected, the best performances use speech act analysis. More features do not always improve the performance, for example, the lexical feature will sometimes decrease performance. Our best performance produced a precision score of 70.38% and an MRR score of 0.825, which is a significant improvement over the baseline's precision score of 27.71% and its MRR score of 0.539.

Algorithm & Features		Correct (out of 314)	Precision (%)	MRR
Baseline		87	27.71	0.539
HITS_AUTHORITY	Lexical	65	20.70	0.524
	Poster	90	28.66	0.569
	SA	215	68.47	0.819
	Lexical + Poster	91	28.98	0.565
	Lexical + SA	194	61.78	0.765
	Poster + SA	**221**	**70.38**	**0.825**
	Lexical + Poster + SA	212	67.52	0.793
HITS_HUB	Lexical	153	48.73	0.682
	Poster	79	25.16	0.527
	SA	195	62.10	0.771
	Lexical + Poster	158	50.32	0.693
	Lexical + SA	177	56.37	0.724
	Poster + SA	**207**	**65.92**	**0.793**
	Lexical + Poster + SA	196	62.42	0.762

Table 7. System Performance Comparison.

Another widely-used graph algorithm in IR is PageRank (Brin and Page, 1998). It is used to investigate the connections between hyperlinks in web page retrieval. PageRank uses a "random walk" model of a web surfer's behavior. The surfer begins from a random node m_i and at each step either follows a hyperlink with the probability of d, or jumps to a random node with the probability of ($1-d$). A weighted PageRank algorithm is used to model weighted relationships of a set of objects. The iterative updating expression is

$$PR^{r+1}(m_i) = (1-d) + d * \sum_{m_j \in B(m_i)} \frac{w_{ji}}{\sum_{m_k \in F(m_j)} w_{jk}} PR^r(m_j) \quad (10)$$

where r and $r+1$ are the numbers of iterations.

We also tested this algorithm in our situation, but the best performance had a precision score of only 47.45% and an MRR score of 0.669. It may be that PageRank's definition and modeling approach does not fit our situation as well as the HITS approach. In HITS, the authority and hub-

based approach is better suited to human conversation analysis than PageRank, which only considers the contributions from backward links of each node in the graph.

6 Conclusions and Future Work

We have presented a novel feature-enriched approach for detecting conversation focus of threaded discussions for the purpose of answering student queries. Using feature-oriented link generation and a graph-based algorithm, we derived a unified framework that integrates heterogeneous sources of evidence. We explored the use of speech act analysis, lexical similarity and poster trustworthiness to analyze discussions.

From the perspective of question answering, this is the first attempt to automatically answer complex and contextual discussion queries beyond factoid or definition questions. To fully automate discussion analysis, we must integrate automatic SA labeling together with our conversation focus detection approach. An automatic system will help users navigate threaded archives and researchers analyze human discussion.

Supervised learning is another approach to detecting conversation focus that might be explored. The tradeoff and balance between system performance and human cost for different learning algorithms is of great interest. We are also exploring the application of graph-based algorithms to other structured-objects ranking problems in NLP so as to improve system performance while relieving human costs.

Acknowledgements

References

Austin, J. 1962. *How to do things with words*. Cambridge, Massachusetts: Harvard Univ. Press.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107--117.

Carvalho, V.R. and Cohen, W.W. 2005. On the collective classification of email speech acts. *In Proceedings of SIGIR-2005*, pp. 345-352.

Erkan, G. and Radev, D. 2004. Lexrank: graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

Feng, D., Shaw, E., Kim, J., and Hovy, E.H. 2006. An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of Intelligent User Interface (IUI-2006)*, pp. 171-177.

Hearst, M.A. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of ACL-1994*.

Kleinberg, J. 1999. Authoritative sources in a hyper-linked environment. *Journal of the ACM*, 46(5).

Kurland, O. and Lee L. 2005. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR-2005*.

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

Mann, W.C. and Thompson, S.A. 1988. Rhetorical structure theory: towards a functional theory of text organization. *Text*, 8 (3), pp. 243-281.

Marom, Y. and Zukerman, I. 2005. Corpus-based generation of easy help-desk responses. *Technical Report, Monash University*. Available at: http://www.csse.monash.edu.au/publications/2005/tr-2005-166-full.pdf.

Mihalcea, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Companion Volume to ACL-2004*.

Mihalcea, R. 2005. unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT/EMNLP 2005*.

Mihalcea, R. and Tarau, P. 2004. TextRank: bringing order into texts. In *Proceedings of EMNLP 2004*.

Mihalcea, R., Tarau, P. and Figa, E. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of COLING 2004*.

Otterbacher, J., Erkan, G., and Radev, D. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of HLT/EMNLP 2005*.

Pang, B. and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL-2004*.

Salton, G. 1989. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA, 1989.

Searle, J. 1969. *Speech Acts*. Cambridge: Cambridge Univ. Press.

Soricut, R. and Marcu, D. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of HLT/NAACL-2003*.

Sporleder, C. and Lapata, M. 2005. Discourse chunking and its application to sentence compression. In *Proceedings of HLT/EMNLP 2005*.

Voorhees, E.M. 2001. Overview of the TREC 2001 question answering track. In *TREC 2001*.

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