Summarizing Emails with Conversational Cohesion and Subjectivity

Giuseppe Carenini, Raymond T. Ng and Xiaodong Zhou
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
{carenini, rng, xdzhou}@cs.ubc.ca

Abstract

In this paper, we study the problem of summarizing email conversations. We first build a sentence quotation graph that captures the conversation structure among emails. We adopt three cohesion measures: clue words, semantic similarity and cosine similarity as the weight of the edges. Second, we use two graph-based summarization approaches, Generalized ClueWordSummarizer and PageRank, to extract sentences as summaries. Third, we propose a summarization approach based on subjective opinions and integrate it with the graph-based ones. The empirical evaluation shows that the basic clue words have the highest accuracy among the three cohesion measures. Moreover, subjective words can significantly improve accuracy.

1 Introduction

With the ever increasing popularity of emails, it is very common nowadays that people discuss specific issues, events or tasks among a group of people by emails(Fisher and Moody, 2002). Those discussions can be viewed as conversations via emails and are valuable for the user as a personal information repository(Ducheneaut and Bellotti, 2001). In this paper, we study the problem of *summarizing email conversations*. Solutions to this problem can help users access the information embedded in emails more effectively. For instance, 10 minutes before a meeting, a user may want to quickly go through a previous discussion via emails that is going to be discussed soon. In that case, rather than reading each individual email one by one, it would be preferable to read a concise summary of the previous discussion with the major information summarized. Email summarization is also helpful for mobile email users on a small screen.

Summarizing email conversations is challenging due to the characteristics of emails, especially the conversational nature. Most of the existing methods dealing with email conversations use the email *thread* to represent the email conversation structure, which is not accurate in many cases (Yeh and Harnly, 2006). Meanwhile, most existing email summarization approaches use quantitative features to describe the conversation structure, e.g., number of recipients and responses, and apply some general multi-document summarization methods to extract some sentences as the summary (Rambow et al., 2004) (Wan and McKeown, 2004). Although such methods consider the conversation structure somehow, they simplify the conversation structure into several features and do not fully utilize it into the summarization process.

In contrast, in this paper, we propose new summarization approaches by sentence extraction, which rely on a fine-grain representation of the conversation structure. We first build a *sentence quotation graph* by content analysis. This graph not only captures the conversation structure more accurately, especially for selective quotations, but it also represents the conversation structure at the finer granularity of sentences. As a second contribution of this paper, we study several ways to measure the cohesion between parent and child sentences in the quotation graph: *clue words* (re-occurring words in the reply)

Proceedings of ACL-08: HLT, pages 353–361,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

(Carenini et al., 2007), *semantic similarity* and *cosine similarity*. Hence, we can directly evaluate the importance of each sentence in terms of its cohesion with related ones in the graph. The extractive summarization problem can be viewed as a node ranking problem. We apply two summarization algorithms, Generalized ClueWordSummarizer and Page-Rank to rank nodes in the sentence quotation graph and to select the corresponding most highly ranked sentences as the summary.

Subjective opinions are often critical in many conversations. As a third contribution of this paper, we study how to make use of the subjective opinions expressed in emails to support the summarization task. We integrate our best cohesion measure together with the subjective opinions. Our empirical evaluations show that subjective words and phrases can significantly improve email summarization.

To summarize, this paper is organized as follows. In Section 2, we discuss related work. After building a sentence quotation graph to represent the conversation structure in Section 3, we apply two summarization methods in Section 4. In Section 5, we study summarization approaches with subjective opinions. Section 6 presents the empirical evaluation of our methods. We conclude this paper and propose future work in Section 7.

2 Related Work

Rambow et al. proposed a sentence extraction summarization approach for email threads (Rambow et al., 2004). They described each sentence in an email conversations by a set of features and used machine learning to classify whether or not a sentence should be included into the summary. Their experiments showed that features about emails and the email thread could significantly improve the accuracy of summarization.

Wan et al. proposed a summarization approach for decision-making email discussions (Wan and McKeown, 2004). They extracted the issue and response sentences from an email thread as a summary. Similar to the issue-response relationship, Shrestha et al.(Shrestha and McKeown, 2004) proposed methods to identify the question-answer pairs from an email thread. Once again, their results showed that including features about the email thread could greatly improve the accuracy. Similar results were obtained by Corston-Oliver et al. They studied how to identify “action” sentences in email messages and use those sentences as a summary(Corston-Oliver et al., 2004). All these approaches used the email thread as a coarse representation of the underlying conversation structure.

In our recent study (Carenini et al., 2007), we built a fragment quotation graph to represent an email conversation and developed a ClueWordSummarizer (CWS) based on the concept of clue words. Our experiments showed that CWS had a higher accuracy than the email summarization approach in (Rambow et al., 2004) and the generic multi-document summarization approach MEAD (Radev et al., 2004). Though effective, the CWS method still suffers from the following four substantial limitations. First, we used a fragment quotation graph to represent the conversation, which has a coarser granularity than the sentence level. For email summarization by sentence extraction, the fragment granularity may be inadequate. Second, we only adopted one cohesion measure (clue words that are based on stemming), and did not consider more sophisticated ones such as semantically similar words. Third, we did not consider subjective opinions. Finally, we did not compared CWS to other possible graph-based approaches as we propose in this paper.

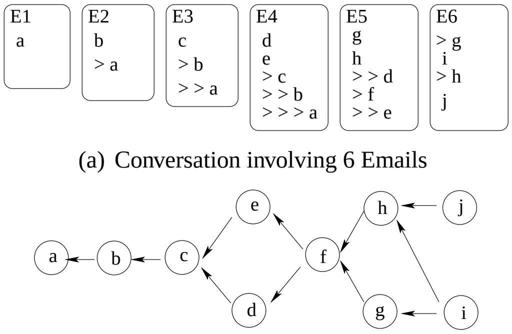
Other than for email summarization, other document summarization methods have adopted graph-ranking algorithms for summarization, e.g., (Wan et al., 2007), (Mihalcea and Tarau, 2004) and (Erkan and Radev, 2004). Those methods built a complete graph for all sentences in one or multiple documents and measure the similarity between every pair of sentences. Graph-ranking algorithms, e.g., PageRank (Brin and Page, 1998), are then applied to rank those sentences. Our method is different from them. First, instead of using the complete graph, we build the graph based on the conversation structure. Second, we try various ways to compute the similarity among sentences and the ranking of the sentences.

Several studies in the NLP literature have explored the reoccurrence of similar words within one document due to text cohesion. The idea has been formalized in the construct of *lexical chains* (Barzilay and Elhadad, 1997). While our approach and lexical chains both rely on lexical cohesion, they are quite different with respect to the kind of linkages considered. Lexical chain is only based on similarities between lexical items in contiguous sentences. In contrast, in our approach, the linkage is based on the existing conversation structure. In our approach, the "chain" is not only "lexical" but also "conversational", and typically spans over several emails.

3 Extracting Conversations from Multiple Emails

In this section, we first review how to build a fragment quotation graph through an example. Then we extend this structure into a sentence quotation graph, which can allow us to capture the conversational relationship at the level of sentences.

3.1 Building the Fragment Quotation Graph



(a) Conversation involving 6 Emails

(b) Fragment Quotation Graph

Figure 1: A Real Example

Figure 1(a) shows a real example of a conversation from a benchmark data set involving 6 emails. For the ease of representation, we do not show the original content but abbreviate them as a sequence of fragments. In the first step, all new and quoted fragments are identified. For instance, email E_3 is decomposed into 3 fragments: new fragment c and quoted fragments b, which in turn quoted a. E_4 is decomposed into de, c, b and a. Then, in the second step, to identify distinct fragments (nodes), fragments are compared with each other and overlaps are identified. Fragments are split if necessary (e.g., fragment gh in E_5 is split into g and h when matched with E_6), and duplicates are removed. At the end, 10 distinct fragments $a, \ldots, j$ give rise to 10 nodes in the graph shown in Figure 1(b).

As the third step, we create edges, which represent the replying relationship among fragments. In general, it is difficult to determine whether one fragment is actually replying to another fragment. We assume that *any new fragment is a potential reply to neighboring quotations – quoted fragments immediately preceding or following it.* Let us consider E_6 in Figure 1(a). there are two edges from node i to g and h, while there is only a single edge from j to h. For E_3, there are the edges (c, b) and (c, a). Because of the edge (b, a), the edge (c, a) is not included in Figure 1(b). Figure 1(b) shows the fragment quotation graph of the conversation shown in Figure 1(a) with all the redundant edges removed. In contrast, if threading is done at the coarse granularity of entire emails, as adopted in many studies, the threading would be a simple chain from E_6 to E_5, E_5 to E_4 and so on. Fragment f reflects a special and important phenomenon, where the original email of a quotation does not exist in the user's folder. We call this as the *hidden email* problem. This problem and its influence on email summarization were studied in (Carenini et al., 2005) and (Carenini et al., 2007).

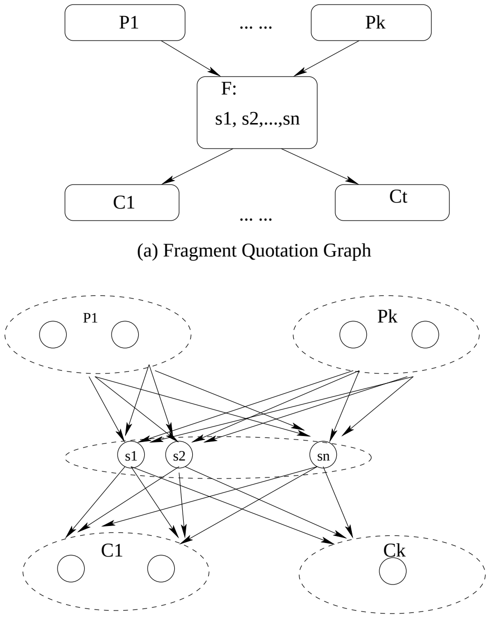
3.2 Building the Sentence Quotation Graph

A fragment quotation graph can only represent the conversation in the fragment granularity. We notice that some sentences in a fragment are more relevant to the conversation than the remaining ones. The fragment quotation graph is not capable of representing this difference. Hence, in the following, we describe how to build a sentence quotation graph from the fragment quotation graph and introduce several ways to give weight to the edges.

In a sentence quotation graph GS, each node represents a distinct sentence in the email conversation, and each edge (u, v) represents the replying relationship between node u and v. The algorithm to create the sentence quotation graph contains the following 3 steps: create nodes, create edges and assign weight to edges. In the following, we first illustrate how to create nodes and edges. In Section 3.3, we discuss different ways to assign weight to edges.

Given a fragment quotation graph GF, we first split each fragment into a set of sentences. For each sentence, we create a node in the sentence quotation graph GS. In this way, each sentence in the email conversation is represented by a distinct node in GS.

As the second step, we create the edges in GS. The edges in GS are based on the edges in GF



(a) Fragment Quotation Graph

(b) Sentence Quotation Graph

Figure 2: Create the Sentence Quotation Graph from the Fragment Quotation Graph

because the edges in GF already reflect the replying relationship among fragments. For each edge $(u, v) \in GF$, we create edges from each sentence of u to each sentence of v in the sentence quotation graph GS. This is illustrated in Figure 2.

Note that when each distinct sentence in an email conversation is represented as one node in the sentence quotation graph, the extractive email summarization problem is transformed into a standard node ranking problem within the sentence quotation graph. Hence, general node ranking algorithms, e.g., Page-Rank, can be used for email summarization as well.

3.3 Measuring the Cohesion Between Sentences

After creating the nodes and edges in the sentence quotation graph, a key technical question is how to measure the degree that two sentences are related to each other, e.g., a sentence s_u is replying to or being replied by s_v. In this paper, we use text cohesion between two sentences s_u and s_v to make this assessment and assign this as the weight of the corresponding edge (s_u, s_v). We explore three types of cohesion measures: (1) clue words that are based on stems, (2) semantic distance based on WordNet and (3) cosine similarity that is based on the word TFIDF vector. In the following, we discuss these three methods separately in detail.

3.3.1 Clue Words

Clue words were originally defined as re-occurring words with the same stem between two adjacent fragments in the fragment quotation graph. In this section, we re-define clue words based on the sentence quotation graph as follows. *A clue word in a sentence S is a non-stop word that also appears (modulo stemming) in a parent or a child node (sentence) of S in the sentence quotation graph.*

The frequency of clue words in the two sentences measures their cohesion as described in Equation 1.

$$weight(s_u, s_v) = \sum_{w_i \in s_u} freq(w_i, s_v) \qquad (1)$$

3.3.2 Semantic Similarity Based on WordNet

Other than stems, when people reply to previous messages they may also choose some semantically related words, such as synonyms and antonyms, e.g., “talk” vs. “discuss”. Based on this observation, we propose to use semantic similarity to measure the cohesion between two sentences. We use the well-known lexical database WordNet to get the semantic similarity of two words. Specifically, we use the package by (Pedersen et al., 2004), which includes several methods to compute the semantic similarity. Among those methods, we choose “lesk” and “jcn”, which are considered two of the best methods in (Jurafsky and Martin, 2008). Similar to the clue words, we measure the semantic similarity of two sentences by the total semantic similarity of the words in both sentences. This is described in the following equation.

$$weight(s_u, s_v) = \sum_{w_i \in s_u} \sum_{w_j \in s_v} \sigma(w_i, w_j), \qquad (2)$$

3.3.3 Cosine Similarity

Cosine similarity is a popular metric to compute the similarity of two text units. To do so, each sentence is represented as a word vector of TFIDF values. Hence, the cosine similarity of two sentences s_u and s_v is then computed as $\frac{\overrightarrow{s_u} \cdot \overrightarrow{s_v}}{||\overrightarrow{s_u}|| \cdot ||\overrightarrow{s_v}||}$.

4 Summarization Based on the Sentence Quotation Graph

Having built the sentence quotation graph with different measures of cohesion, in this section, we develop two summarization approaches. One is the generalization of the CWS algorithm in (Carenini et al., 2007) and one is the well-known Page-Rank algorithm. Both algorithms compute a score, $SentScore(s)$, for each sentence (node) s, which is used to select the top-k% sentences as the summary.

4.1 Generalized ClueWordSummarizer

Given the sentence quotation graph, since the weight of an edge (s,t) represents the extent that s is related to t, a natural assumption is that the more relevant a sentence (node) s is to its parents and children, the more important s is. Based on this assumption, we compute the weight of a node s by summing up the weight of all the outgoing and incoming edges of s. This is described in the following equation.

$$SentScore(s) = \sum_{(s,t)\in GS} weight(s,t) + \sum_{(p,s)\in GS} weight(p,s) \tag{3}$$

The weight of an edge (s,t) can be any of the three metrics described in the previous section. Particularly, when the weight of the edge is based on clue words as in Equation 1, this method is equivalent to Algorithm CWS in (Carenini et al., 2007). In the rest of this paper, let CWS denote the Generalized ClueWordSummarizer when the edge weight is based on clue words, and let *CWS-Cosine* and *CWS-Semantic* denote the summarizer when the edge weight is cosine similarity and semantic similarity respectively. *Semantic* can be either “lesk” or “jcn”.

4.2 Page-Rank-based Summarization

The Generalized ClueWordSummarizer only considers the weight of the edges without considering the importance (weight) of the nodes. This might be incorrect in some cases. For example, a sentence replied by an important sentence should get some of its importance. This intuition is similar to the one inspiring the well-known Page-Rank algorithm. The traditional Page-Rank algorithm only considers the outgoing edges. In email conversations, what we want to measure is the cohesion between sentences no matter which one is being replied to. Hence, we need to consider both incoming and outgoing edges and the corresponding sentences.

Given the sentence quotation graph G_s, the Page-Rank-based algorithm is described in Equation 4. $PR(s)$ is the Page-Rank score of a node (sentence) s. d is the dumping factor, which is initialized to 0.85 as suggested in the Page-Rank algorithm. In this way, the rank of a sentence is evaluated globally based on the graph.

5 Summarization with Subjective Opinions

Other than the conversation structure, the measures of cohesion and the graph-based summarization methods we have proposed, the importance of a sentence in emails can be captured from other aspects. In many applications, it has been shown that sentences with subjective meanings are paid more attention than factual ones(Pang and Lee, 2004)(Esuli and Sebastiani, 2006). We evaluate whether this is also the case in emails, especially when the conversation is about decision making, giving advice, providing feedbacks, etc.

A large amount of work has been done on determining the level of subjectivity of text (Shanahan et al., 2005). In this paper we follow a very simple approach that, if successful, could be extended in future work. More specifically, in order to assess the degree of subjectivity of a sentence s, we count the frequency of words and phrases in s that are likely to bear subjective opinions. The assumption is that the more subjective words s contains, the more likely that s is an important sentence for the purpose of email summarization. Let $SubjScore(s)$ denote the number of words with a subjective meaning. Equation 5 illustrates how SubjScore(s) is computed. $SubjList$ is a list of words and phrases that indicate subjective opinions.

$$SubjScore(s) = \sum_{w_i \in SubjList, w_i \in s} freq(w_i) \tag{5}$$

The SubjScore(s) alone can be used to evaluate the importance of a sentence. In addition, we can combine SubjScore with any of the sentence scores based on the sentence quotation graph. In this paper, we use a simple approach by adding them up as the final sentence score.

$$PR(s) = (1-d) + d * \frac{\sum_{s_i \in child(s)} weight(s, s_i) * PR(s_i) + \sum_{s_j \in parent(s)} weight(s_j, s) * PR(s_j)}{\sum_{s_i \in child(s)} weight(s, s_i) + \sum_{s_j \in parent(s)} weight(s_j, s)} \quad (4)$$

As to the subjective words and phrases, we consider the following two lists generated by researchers in this area.

- $OpFind$: The list of subjective words in (Wilson et al., 2005). The major source of this list is from (Riloff and Wiebe, 2003) with additional words from other sources. This list contains 8,220 words or phrases in total.

- $OpBear$: The list of opinion bearing words in (Kim and Hovy, 2005). This list contains 27,193 words or phrases in total.

6 Empirical Evaluation

6.1 Dataset Setup

There are no publicly available annotated corpora to test email summarization techniques. So, the first step in our evaluation was to develop our own corpus. We use the Enron email dataset, which is the largest public email dataset. In the 10 largest *inbox* folders in the Enron dataset, there are 296 email conversations. Since we are studying summarizing email conversations, we required that each selected conversation contained at least 4 emails. In total, 39 conversations satisfied this requirement. We use the MEAD package to segment the text into 1,394 sentences (Radev et al., 2004).

We recruited 50 human summarizers to review those 39 selected email conversations. Each email conversation was reviewed by 5 different human summarizers. For each given email conversation, human summarizers were asked to generate a summary by directly selecting important sentences from the original emails in that conversation. We asked the human summarizers to select 30% of the total sentences in their summaries.

Moreover, human summarizers were asked to classify each selected sentence as either *essential* or *optional*. The essential sentences are crucial to the email conversation and have to be extracted in any case. The optional sentences are not critical but are useful to help readers understand the email conversation if the given summary length permits. By classifying essential and optional sentences, we can distinguish the core information from the supporting ones and find the most convincing sentences that most human summarizers agree on.

As essential sentences are more important than the optional ones, we give more weight to the essential selections. We compute a $GSValue$ for each sentence to evaluate its importance according to the human summarizers' selections. The score is designed as follows: for each sentence s, one essential selection has a score of 3, one optional selection has a score of 1. Thus, the GSValue of a sentence ranges from 0 to 15 (5 human summarizers x 3). The GSValue of 8 corresponds to 2 essential and 2 optional selections. If a sentence has a GSValue no less than 8, we take it as an *overall essential* sentence. In the 39 conversations, we have about 12% overall essential sentences.

6.2 Evaluation Metrics

Evaluation of summarization is believed to be a difficult problem in general. In this paper, we use two metrics to measure the accuracy of a system generated summary. One is *sentence pyramid precision*, and the other is *ROUGE recall*. As to the statistical significance, we use the 2-tail pairwise student t-test in all the experiments to compare two specific methods. We also use ANOVA to compare three or more approaches together.

The sentence pyramid precision is a relative precision based on the GSValue. Since this idea is borrowed from the pyramid metric by Nenkova et al.(Nenkova et al., 2007), we call it the *sentence pyramid precision*. In this paper, we simplify it as the *pyramid precision*. As we have discussed above, with the reviewers' selections, we get a GSValue for each sentence, which ranges from 0 to 15. With this GSValue, we rank all sentences in a descendant order. We also group all sentences with the same GSValue together as one tier T_i, where i is the corresponding GSValue; i is called the *level* of the tier T_i. In this way, we organize all sentences into a pyramid: a sequence of tiers with a descendant order of levels. With the pyramid of sentences, the accuracy of a summary is evaluated over the best summary we can achieve under the same summary length. The best summary of k sentences are the top k sentences in terms of GSValue.

Other than the sentence pyramid precision, we also adopt the ROUGE recall to evaluate the generated summary with a finer granularity than sentences, e.g., n-gram and longest common subsequence. Unlike the pyramid method which gives more weight to sentences with a higher GSValue, ROUGE is not sensitive to the difference between essential and optional selections (it considers all sentences in one summary equally). Directly applying ROUGE may not be accurate in our experiments. Hence, we use the overall essential sentences as the gold standard summary for each conversation, i.e., sentences in tiers no lower than T_8. In this way, the ROUGE metric measures the similarity of a system generated summary to a gold standard summary that is considered important by most human summarizers. Specifically, we choose ROUGE-2 and ROUGE-L as the evaluation metric.

6.3 Evaluating the Weight of Edges

In Section 3.3, we developed three ways to compute the weight of an edge in the sentence quotation graph, i.e., clue words, semantic similarity based on WordNet and cosine similarity. In this section, we compare them together to see which one is the best. It is well-known that the accuracy of the summarization method is affected by the length of the summary. In the following experiments, we choose the summary length as 10%, 12%, 15%, 20% and 30% of the total sentences and use the aggregated average accuracy to evaluate different algorithms.

Table 1 shows the aggregated pyramid precision over all five summary lengths of CWS, CWS-Cosine, two semantic similarities, i.e., CWS-lesk and CWS-jcn. We first use ANOVA to compare the four methods. For the pyramid precision, the F ratio is 50, and the p-value is 2.1E-29. This shows that the four methods are significantly different in the average accuracy. In Table 1, by comparing CWS with the other methods, we can see that CWS obtains the highest precision (0.60). The widely used cosine similarity does not perform well. Its precision (0.39) is about half of the precision of CWS with a p-value less than 0.0001. This clearly shows that CWS is significantly better than CWS-Cosine. Meanwhile, both semantic similarities have lower accuracy than CWS, and the differences are also statistically significant even with the conservative Bonferroni adjustment (i.e., the p-values in Table 1 are multiplied by three).

	CWS	CWS-Cosine	CWS-lesk	CWS-jcn
Pyramid	0.60	0.39	0.57	0.57
p-value		<0.0001	0.02	0.005
ROUGE-2	0.46	0.31	0.39	0.35
p-value		<0.0001	<0.001	<0.001
ROUGE-L	0.54	0.43	0.49	0.45
p-value		<0.0001	<0.001	<0.001

Table 1: Generalized CWS with Different Edge Weights

The above experiments show that the widely used cosine similarity and the more sophisticated semantic similarity in WordNet are less accurate than the basic CWS in the summarization framework. This is an interesting result and can be viewed at least from the following two aspects. First, clue words, though straight forward, are good at capturing the important sentences within an email conversation. The higher accuracy of CWS may suggest that people tend to use the same words to communicate in email conversations. Some related words in the previous emails are adopted exactly or in another similar format (modulo stemming). This is different from other documents such as newspaper articles and formal reports. In those cases, the authors are usually professional in writing and choose their words carefully, even intentionally avoid repeating the same words to gain some diversity. However, for email conversation summarization, this does not appear to be the case.

Moreover, in the previous discussion we only considered the accuracy in precision without considering the runtime issue. In order to have an idea of the runtime of the two methods, we did the following comparison. We randomly picked 1000 pairs of words from the 20 conversations and compute their semantic distance in “jcn”. It takes about 0.056 seconds to get the semantic similarity for one pair on the average. In contrast, when the weight of edges are computed based on clue words, the average runtime to compute the SentScore for all sentences in a conversation is only 0.05 seconds, which is even a little less than the time to compute the semantic similarity of one pair of words. In other words, when CWS has generated the summary of one conversation, we can only get the semantic distance between one pair of words. Note that for each edge in the sentence quotation graph, we need to compute the distance for every pair of words in each sentence. Hence, the empirical results do not support the use of semantic similarity. In addition, we do not discuss the runtime performance of CWS-cosine here because of its extremely low accuracy.

6.4 Comparing Page-Rank and CWS

Table 2 compares Page-Rank and CWS under different edge weights. We compare Page-Rank only with CWS because CWS is better than the other Generalized CWS methods as shown in the previous section. This table shows that Page-Rank has a lower accuracy than that of CWS and the difference is significant in all four cases. Moreover, when we compare Table 1 and 2 together, we can find that, for each kind of edge weight, Page-Rank has a lower accuracy than the corresponding Generalized CWS. Note that Page-Rank computes a node's rank based on all the nodes and edges in the graph. In contrast, CWS only considers the similarity between neighboring nodes. The experimental result indicates that for email conversation, the local similarity based on clue words is more consistent with the human summarizers' selections.

6.5 Evaluating Subjective Opinions

Table 3 shows the result of using subjective opinions described in Section 5. The first 3 columns in this table are pyramid precision of CWS and using 2 lists of subjective words and phrases alone. We can see that by using subjective words alone, the precision of each subjective list is lower than that of CWS. However, when we integrate CWS and subjective words together, as shown in the remaining 2 columns, the precisions get improved consistently for both lists. The increase in precision is at least 0.04 with statistical significance. A natural question to ask is whether clue words and subjective words overlap much. Our analysis shows that the overlap is minimal. For the list of OpFind, the overlapped words are about 8% of clue words and 4% of OpFind that appear in the conversations. This result clearly shows that clue words and subjective words capture the importance of sentences from different angles and can be used together to gain a better accuracy.

	CWS	PR-Clue	PR-Cosine	PR-lesk	PR-jcn
Pyramid	0.60	0.51	0.37	0.54	0.50
p-value		< 0.0001	< 0.0001	< 0.0001	< 0.0001
ROUGE-2	0.46	0.4	0.26	0.36	0.39
p-value		0.05	< 0.0001	0.001	0.02
ROUGE-L	0.54	0.49	0.36	0.44	0.48
p-value		0.06	< 0.0001	0.0005	0.02

Table 2: Compare Page-Rank with CWS

	CWS	OpFind	OpBear	CWS+OpFind	CWS+OpBear
Pyramid	0.60	0.52	0.59	**0.65**	**0.64**
p-value		0.0003	0.8	<0.0001	0.0007
ROUGE-2	0.46	0.37	0.44	**0.50**	**0.49**
p-value		0.0004	0.5	0.004	0.06
ROUGE-L	0.54	0.48	0.56	**0.60**	**0.59**
p-value		0.01	0.6	0.0002	0.002

Table 3: Accuracy of Using Subjective Opinions

7 Conclusions

We study how to summarize email conversations based on the conversational cohesion and the subjective opinions. We first create a sentence quotation graph to represent the conversation structure on the sentence level. We adopt three cohesion metrics, clue words, semantic similarity and cosine similarity, to measure the weight of the edges. The Generalized ClueWordSummarizer and Page-Rank are applied to this graph to produce summaries. Moreover, we study how to include subjective opinions to help identify important sentences for summarization.

The empirical evaluation shows the following two discoveries: (1) The basic CWS (based on clue words) obtains a higher accuracy and a better runtime performance than the other cohesion measures. It also has a significant higher accuracy than the Page-Rank algorithm. (2) By integrating clue words and subjective words (phrases), the accuracy of CWS is improved significantly. This reveals an interesting phenomenon and will be further studied.

References

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of* *the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, Spain*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web*, pages 107–117.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2005. Scalable discovery of hidden emails from large folders. In *ACM SIGKDD'05*, pages 544–549.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 91–100.

Simon Corston-Oliver, Eric K. Ringger, Michael Gamon, and Richard Campbell. 2004. Integration of email and task lists. In *First conference on email and anti-Spam(CEAS)*, Mountain View, California, USA, July 30-31.

Nicolas Ducheneaut and Victoria Bellotti. 2001. E-mail as habitat: an exploration of embedded personal information management. *Interactions*, 8(5):30–38.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research(JAIR)*, 22:457–479.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation*, May 24-26.

Danyel Fisher and Paul Moody. 2002. Studies of automated collection of email records. In *University of Irvine ISR Technical Report UCI-ISR-02-4*.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Second Edition)*. Prentice-Hall.

Soo-Min Kim and Eduard Hovy. 2005. Automatic detection of opinion bearing words and sentences. In *Proceedings of the Second International Joint Conference on Natural Language Processing: Companion Volume*, Jeju Island, Republic of Korea, October 11-13.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, July.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: incorporating human content selection variation in summarization evaluation. *ACM Transaction on Speech and Language Processing*, 4(2):4.

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 271–278.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, pages 38–41, May 3-5.

Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938, November.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In *HLT/NAACL*, May 2–7.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 105–112.

James G. Shanahan, Yan Qu, and Janyce Wiebe. 2005. *Computing Attitude and Affect in Text: Theory and Applications (The Information Retrieval Series)*. Springer-Verlag New York, Inc.

Lokesh Shrestha and Kathleen McKeown. 2004. Detection of question-answer pairs in email conversations. In *Proceedings of COLING'04*, pages 889–895, August 23–27.

Stephen Wan and Kathleen McKeown. 2004. Generating overview summaries of ongoing email thread discussions. In *Proceedings of COLING'04, the 20th International Conference on Computational Linguistics*, August 23–27.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559, Prague, Czech Republic, June.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35.

Jen-Yuan Yeh and Aaron Harnly. 2006. Email thread reassembly using similarity matching. In *Third Conference on Email and Anti-Spam (CEAS)*, July 27 - 28.