Using Question-Answer Pairs in Extractive Summarization of Email Conversations

Abstract

While sentence extraction as an approach to summarization has been shown to work in documents of certain genres, because of the conversational nature of email communication, sentence extraction may not result in a coherent summary. In this paper, we present our work on augmenting extractive summaries of threads of email conversations with automatically detected question-answer pairs. We compare various approaches to integrating question-answer pairs in the extractive summaries, and show that their use improves the quality of email summaries.

1 Introduction

Email conversations are a natural means of getting answers to one's questions. And, the asynchronous nature of email conversation makes it possible for one to pursue several questions in parallel. As a consequence, question-answer exchanges figure as one of the dominant uses of email conversations. In fact, in our corpus of email exchanges, we found that about 20% of all email threads focus primarily on a question-answer exchange, while about 40% of all email threads involve question-answer exchange of some form, whether one question is posed and multiple people respond or whether multiple questions are posed and multiple responses given. For these type of email threads, a summary that can highlight the main question(s) asked and the response(s) given would be useful.

The most common technique for summarization is the use of sentence extraction using variants of lexical frequency (Nenkova et al., 2003; Blair-Goldensohn et al., 2004). In (Anonymous-A, 2004) we showed that sentence extraction can also be successfully applied to summarize email threads if augmented with email-specific features and presented using the dialogic structure of email communication. However, these kinds of approaches ignore the key characteristics of question-answer exchange threads; an extractive summary may not include the answer to a question included in the summary.

In this paper, we present a summarization system that integrates question-answer pairs in extractive summaries of email conversations, and show that such an integrative approach improves the quality of summarization for question-answer exchange threads. Our work explores three strategies for inclusion of question-answer pairs, demonstrating that an approach which is sensitive to the characteristics of the thread has the best performance. In order to experiment with different strategies, we use automatic techniques to create training data and validate that the data is accurate.

2 Previous and Related Work

While there has been no work on using automatically detected question and answer pairs in summarizing threads of email conversations, we present here previously reported work on individual email as well as archived discussion lists summarization. (Muresan et al., 2001) describe work on summarizing individual email messages using machine learning approaches to learn rules for salient noun phrase extraction. (Lam et al., 2002) present work on email summarization by exploiting the thread structure of email conversation and common features such as named entities and dates. They summarize the message only, though the content of the message to be summarized is "expanded" using the content from its ancestor messages. The expanded message is passed to a document summarizer which is used as a black

box to generate summaries.

(Newman and Blitzer, 2003) also address the problem of summarizing archived discussion lists. They cluster messages into topic groups, and then extract summaries for each cluster. The summary of a cluster is extracted using a scoring metric based on sentence position, lexical similarity of a sentence to cluster centroid, and a feature based on quotation, among others.

(Dalli et al., 2004) describe FASIL, an email summarization system for use in a voice-based Virtual Personal Assistant developed at University of Sheffield. The system uses a ranking function that uses the occurrence of named entities and other empirically determined parameters to rank original sentences in individual email messages, and selects the top required number of sentences to generate an extractive summary of the email. The system also uses anaphora resolution to improve the quality of the generated summaries.

Email is different in important respects from (multi-party) dialog. However, there has been some work on summarizing meetings that bears some relation to ours. (Zechner, 2002), for example, presents a meeting summarization system which uses the MMR algorithm to find sentences that are most salient while minimizing the redundancy in the summary. The similarity weights in the MMR algorithm are modified using three features, including whether a sentence belongs to a question-answer pair. The use of question-answer pair detection is an interesting proposal that is also applicable to our work.

3 The Data

Our corpus consists of about 300 threads of emails sent during one academic year among the members of the board of the student organization of the ACM at our institution; hence, we call it the ACM corpus. The emails deal mainly with planning events of various types, though other issues were also addressed. On average, each thread contained 3.25 email messages, with all threads containing at least two messages, and the longest thread containing 18 messages. The threads in our corpus were created using the "In-Reply-To" header information. Although this approach can lead to ill-formed threads primarily resulting from the erroneous use of the "Reply" command in email clients even when a new conversation is being started, this is not a problem in the case of our corpus. This is because the participants of email conversations in our corpus use a single thread exclusively for a single topic, although it may be the case that a thread also discusses topics related to the main topic under discussion.

Two annotators were asked to perform two tasks: write summaries of the email threads in the ACM corpus, and highlight and link question-answer pairs in the email threads. We instructed the annotators on the format of the summary; specifically, we asked them to use the past tense, and to use speech-act verbs and embedded clauses (for example, *Dolores reported she'd gotten 7 people to sign up* instead of *Dolores got 7 people to sign up*). We requested the length to be about 5% to 20% of the original text length, but not longer than 100 lines. As for the content for the summaries, we asked the annotators to use their judgment.

Research has shown that there are many different possible good summaries given an input document set (Nenkova and Passonneau, 2004; Teufel and van Halteren, 2004) In fact, the kappa statistic for the agreement between the two annotators in deciding whether a sentence is a summary sentence is 0.45, suggesting that this observation holds for email summarization as well. We discuss this more in Section 4.1.

As for question detection, the annotators were asked to highlight only those questions that were asked to obtain some information whether the quesion was posed in an interrogative form with a question mark ending the question or was posed in a declarative form such as "I was wondering if ...". We asked annotators to ignore rhetorical questions (questions used for purposes other than to obtain the information the question asked).

3.1 Sentence Extraction Data

Creating labeled data for sentence extraction is a tedious and time-consuming job. Our approach features the use of automatic techniques applied to human summarization to develop a training corpus for extractive summarization of email threads. In extractive summarization a certain function, heuristically determined or machine learned, maps extractions, or their representations, of the document to be summarized into a binary classification. Those positively classified are used in the summary and the rest discarded. To create the training data for our machine learning approach, we repre-

sent each sentence of the email threads in the corpus with a feature vector along with its binary classification. The binary classifications of these sentences are derived from the human written summaries. Since our annotators were not asked to categorize thread sentences according to whether the sentence should be a summary sentence or not, but rather asked to write the manual summaries, a more natural task, the task of categorizing the sentences for training data had to be done automatically. This automatic approach must select the sentences whose content is reflected in the corresponding manual summaries and also determine the compression rate used by the human summarizers.

In our earlier work (Anonymous-A, 2004), we used the sentence-similarity finder SimFinder (Hatzivassiloglou et al., 2001) in order to rate the similarity of each sentence in a thread to each sentence in the corresponding manual summary. SimFinder uses a combination of lexical and linguistic features to assign a similarity score to an input pair of texts. For each sentence in the thread, excluding sentences that are being quoted, signatures and the like, we retained the highest similarity score with the corresponding manual summary sentences. Using these highest scores, we ranked the thread sentences, and categorized a certain proportion of the top ranked thread sentences with scores greater than 0 as summary sentences. We call this proportion the summary size (for example, a summary size of 20% , which implies a compression rate of 80%, means that the sentences with their scores in the top 20% will be categorized as summary sentences). Thus, for each email thread in the corpus, we used SimFinder to determine which thread sentences contained the information in the corresponding human written summary, and these sentences were used as positive examples, while the rest of the thread sentences were used as negative examples.

While (Anonymous-A, 2004) assumes a compression rate of 80%, there is no guarantee that an 80% compression rate would yield the best results. For this paper, we investigated what summary size would best match the compression rates used by the human summarizers. Also, we investigated whether the use of SimFinder (Hatzivassiloglou et al., 2001) in identifying summary sentences was a reasonable approach. To do this, we first randomly chose about 10% of the ACM threads, which we call gold standard threads, and manually classified the sentences in these threads, which we call gold standard sentences, according to whether these sentences' content were manually verified to be reflected in one of the human written summaries. Those gold standard sentences whose content were reflected in the corresponding human summary were given a classification of "Y", implying that the sentence is a summary sentence, and the rest were given a classification of "N", implying that the sentence is not a summary sentence, giving us the gold standard classification.[1] In doing this we found out that of the 109 total gold standard sentences from the selected threads, 59 were selected as being reflected in the human written summaries while 50 were disregarded. This implies a compression rate of less than 50% (50/109) for the selected threads while we had instructed the annotators to use a compression rate of at least 80%. After obtaining the gold standard classifications, we used SimFinder to generate the automated classification. This was done by using SimFinder to score the gold standard sentences against their respective summary sentences. These scores were then used to automatically classify the gold standard sentences at different compression rates. For example, at a compression rate of 80%, the sentences with top 20% scores in a thread were classified as summary sentences. We then compared these Simfinder induced automated classification with the manual gold standard classification. The results are shown in Table 1.

While F-measure score is the highest at a compression rate of 50%, precision at this rate is lower than that at a compression rate of 45%. Further, we are interested in minimizing the summary size. These observations suggest that the best compression to use would be 55% (a summary size of 45%). Also, it is interesting to note that the precision score does not go below 75% for all the compression rates we investigated. This implies that Simfinder can be used to automate and approximate the task of selecting thread sentences whose content are reflected in the human written summaries, and thus validates our approach to the development of the training data for sentence extraction.

[1] While this process selects those sentences in an email thread whose content are reflected in the manual summaries, our use of Simfinder attempts to automate and approximate this manual process.

Summary size	**20%**	**30%**	**40%**	**45%**	**50%**	**55%**	**60%**
Recall	0.268	0.500	0.625	0.768	0.803	0.821	0.857
Precision	0.750	0.824	0.833	0.827	0.803	0.780	0.750
F-measure	0.394	0.622	0.714	0.796	0.803	0.80	0.80

Table 1: Results for comparing Simfinder induced sentence classification using various summary sizes with that of manual sentence classification

3.2 Question-Answer Pair Detection Data

The two annotators were each asked to highlight and link question and answer pairs in the ACM corpus as mentioned earlier in this section. Our work presented here is based on the work these annotators had completed at the time of this writing. One of the annotators has completed work on 200 threads of the ACM corpus of which there are 80 QA threads (threads with question and answer pairs), 98 question segments, and 142 question and answer pairs. The other annotator has completed work on 138 threads of which there are 61 QA threads, 72 question segments, and 92 question and answer pairs. We consider a segment to be a question segment if a sentence in that segment has been highlighted as a question. Similarly, we consider a segment to be an answer segment if a sentence in that segment has been paired with a question to form a question and answer pair. The kappa statistic (Carletta, 1996) for identifying question segments is 0.68, and for linking question and answer segments given a question segment is 0.81, indicating that identification of question and answer segments is a more objective task than writing a summary.

4 Extractive Summarization and Question-Answer Pair Detection

4.1 Extractive Summarization

Our approach entails the creation of training data, as described in Section 3.1, and the use of this data in learning sentence classifiers. For the experiments we discuss here, we used Ripper (Cohen, 1996) as our machine learning tool, and the results we present are based on 5-fold cross-validation. Since we are interested in performance improvement of extractive summarization with the use of question-answer pairs in email threads, we confine our experiments to those email threads in the ACM corpus that have at least one question-answer pair as annotated by the annotators. As mentioned in Section 3.2, annotator A had identified 80 threads with question-answer pairs among the 200 threads that she had worked on. Annotator B had identified 61 threads with question-answer pairs among 138 threads he had worked on. Using these two subsets of ACM threads, we obtained two sets of training data for learning sentence extraction rules as described in Section 3.1 at a compression rate of 55%. Each set of training data contains a feature vector representation of each sentence along with their SimFinder derived classification.

As a baseline for comparison against our integration of questions and answers, we used the sentence extraction features from our earlier work (Anonymous-A, 2004), which included the standard set of features such as length, position in the document, TFIDF scores of the terms in the sentences as well as other features derived from the nature of email conversation and the structure of the email thread. Table 2 summarizes the information on the two data sets. From the table, it can be seen that the effective summary size in annotator A's data set is about 43% and that in annotator B's data set is about 39%. The summary sizes aren't 45% because some of the thread sentences that are in the top 45% of the SimFinder scored sentences have insignificant similarity scores, and, hence, not considered as summary sentence.

	Sentences	**Positives**	**Threads**
Annotator A	1174	502	80
Annotator B	876	342	61

Table 2: Summary of training data for sentence extraction showing the number of sentences, number of summary sentences (i.e., those classified as positive), and the number of threads in each of the two data sets

	P	R	F	Summary Size
Anno A	0.550	0.516	0.532	0.41
Anno B	0.514	0.468	0.490	0.36

Table 3: 5-fold cross-validation sentence extraction results using the full feature set at 55% compression showing Precision(P), Recall(R), F-measure(F), and Summary Size for the two datasets

Table 3 shows the results for extractive summarization of email threads with 5-fold cross validation using the full feature set. Anno A represents the results for annotator A's dataset, and Anno B

represents that of annotator B. Column 'P' shows the precision for the two datasets, whereas column 'R' shows the recall and column 'F' shows the f-measure scores. As can be seen from the table, results are better with the data set obtained from annotator A than with those obtained from annotator B. This could be because the data set obtained from annotator A has more data points to train from. Furthermore, the kappa statistic for sentence classification (whether a sentence is a summary sentence or not) using the human written summaries of the two annotators through Simfinder scores is 0.45. This implies a marked difference in the both the content and the style of the summaries of the two annotators, and could have affected the learnability of sentence extraction rules.

4.2 Question-Answer Pair Detection

In order to include questions and answers in email summaries, we first need to be able to detect them in the input email threads. In (Anonymous-B, 2004) we presented work on the detection of question and answer pairs in email threads, and showed that various features based on the structure of email threads can be used in conjunction with lexical similarity of discourse segments for question-answer pairing.

In this section, we describe the application of our earlier work on question and answer detection to our new data. The machine learned rules, described in (Anonymous-B, 2004), tell us whether a discourse segment following a question segment in an email conversation was offered as an answer to the question. A question segment is a paragraph of written text in an email that contains a question. We show in (Anonymous-B, 2004) that learning separate rules for the different subset of training data results in better performance for automatic question-answer pair detection. Specifically, the training data set mentioned in Section 3.2 was divided into two sets, one set containing question segments with two or fewer answer segments as annotated by either of the annotators and the other set containing question segments with three or more answer segments. Thus, separate rule sets for QA pairing were learned for these two data sets resulting in a final precision score of 0.728, recall score of 0.732 and F-Measure score of 0.730. These rules were then used to identify the question-answer segment pairs in the data sets for each of the two annotations of the two annotators mentioned in Section 3.1.

5 Integrating Question-Answer Pairs with Extractive Sentences

In our analysis of the two sets of training data derived from the annotations of two human annotators and SimFinder scores, we found that questions and answers have a better probability of being a summary sentence than non question-answer sentences. For example, for annotator A's data set, a sentence had about 43% probability of being a summary sentence, whereas, for automatically detected questions the number was about 57%, and for automatically detected answers it was 61%. Similarly, for annotator B's data set, a sentence had about 39% probability of being a summary sentence, whereas, for a question the number was 52%, and for an answer it was 52%. This tells us that detection of question and answer pairs in threads of email conversation and their subsequent use in summaries improves their quality. Further, out of the 154 QA pairs annotated by annotator A, 96 have positive SimFinder derived classification. Similarly, out of 94 QA pairs annotated by annotator B, 52 have positive classification. These observations tell us that inclusion of QA pairs in summaries enhances their quality.

We have identified three types of approaches to integrating automatically detected question-answer pairs in threads of email conversations with their extractive summaries. The first approach is to use the fact that a sentence figures as an answer to a question asked earlier in the thread as an additional feature in our machine learning-based extractive summarization approach. In other words, in this approach, sentence extraction rules are machine learned using training data which represents each thread sentence with a feature vector which includes a feature that says whether the sentence is an answer offered to a question asked earlier in the email thread. The second approach is to add automatically detected answers to questions that appear in the extractive summaries produced by the approach described in Section 4.1. This approach can be further developed by adding questions whose answers appear in the extractive summaries. Effectively, the second approach tries to improve the coherency of extractive summaries by adding questions to extracted answers and answers to extracted questions so that the summary reader has a better context for understanding the

summary. In the third approach we start with automatically detected question-answer pair sentences which are then augmented with extractive sentences that do not appear already in the question-answer pair sentences.

Table 4 shows the results of the first approach, i.e., adding an extra feature to our extractive summarization approach that says whether a sentence figures as an answer to a question asked earlier in the thread. While we get an improvement over the results shown in Table 3 for annotator A's data set primarily due to an improvement in precision, an improvement is not seen for annotator B's data set.

	P	R	F	Summary Size
Anno A	0.591	0.506	0.545	0.37
Anno B	0.502	0.459	0.479	0.36

Table 4: Results with adding an "answer" feature in extractive summarization showing Precision(P), Recall(R), F-measure(F), and Summary Size for the two datasets

Our second approach in integrating question-answer pairs with extractive sentences is to include an answer sentence for all question sentences identified as an extractive sentence if the extractive summary does not already contain the answer sentence. This attempts to mitigate the problem of summaries which do not include answers to questions appearing in the summary as described in Section 1. Also, there are cases when the extractive approach to summarization of email threads selects sentences from an answer segment but does not include the corresponding question that the answer segment attempts to answer. Results for an extractive summary augmented with both answer sentences for extracted question sentences and question sentences for extracted answer sentences are shown in Table 5. As can be seen when these results are compared with those in Table 4, we get an improvement in recall for both the data sets, while the precision suffers a little for the annotator A's data set. Overall we get an improvement of f-measure. Also, the summary size is reasonable.

	P	R	F	Summary Size
Anno A	0.535	0.590	0.561	0.47
Anno B	0.507	0.564	0.533	0.43

Table 5: Results with sentence extraction augmented with answer sentences for extracted question sentences and question sentences for extracted answer sentences showing Precision(P), Recall(R), F-measure(F), and Summary Size for the two datasets

The final approach we considered is to add the extractive sentences to the question and answer pair sentences if needed. We first start with the question-answer pairs detected in an email thread. The question sentence in the question segment and the sentence in the answer segment which is most similar to its question segment using cosine similarity of TF-IDF vectors are selected as summary sentences. Then extractive sentences are added if they are not in the automatically detected question or the answer segment. With this approach, we are assuming that a sentence pair from each of the question-answer pair segments must be included in the summary along with other extractive sentences that are not in any question-answer pair segments. The results with this approach is shown in Table 6. These results show that we do not necessarily get an improvement over the results shown in Table 5, however we do get smaller summaries and some improvement in precision for annotator A's dataset.

	P	R	F	Summary Size
Anno A	0.556	0.542	0.549	0.42
Anno B	0.493	0.532	0.512	0.42

Table 6: Results with question-answer pairs augmented with extractive sentences showing Precision(P), Recall(R), F-measure(F), and Summary Size for the two datasets

Because the third approach starts with the automatically detected Question-Answer pairs as the basis for the summary, this approach seems better suited for email-threads which are specifically dedicated to question and answer exchanges. Similarly, the second approach seems better suited for email threads which contain a few, if any, question and answer exchanges because the extractive sentences, which form the basis of the summaries for this approach, are meant to capture the gist of the thread conversation, and the automatically question-answer pairs added to augment the coherency of the extracted sentences. To verify this we divided annotator A's dataset into two subsets, one with email threads specifically dedicated to question-answer exchanges (QA threads) and the other which have some question-answer exchanges but are not central to the main topic under discussion (non-QA threads). We determined this thread category using the human annotations. During the annotation process, we had asked the human annotators to categorize email threads into five categories, one of which was

Question-Answer. The two sets of annotator A's dataset were then tested with 5-fold crossvalidation using the second and the third approach. Table 7 shows the results for annotator A's QA threads. The first row shows the results for question-answer pair sentences augmented with extractive sentences, the second row shows the results for extractive sentences augmented with question-answer pair sentences. These show that the results for question-answer pair sentences augmented with extractive sentences performes better in all aspects compared to the other approach.

	P	R	F	Summary Size
QA + SE	0.550	0.630	0.588	0.42
SE + QA	0.527	0.615	0.567	0.47

Table 7: Results showing Precision(P), Recall(R), F-measure(F), and Summary Size for QA only threads from Annotator A's dataset with different approaches

Similarly, Table 8 shows the results for annotator A's non-QA threads. The first row shows the results for question-answer pair sentences augmented with extractive sentences, the second row shows the results for extractive sentences augmented with question-answer pair sentences. These show that the results for question-answer pair sentences augmented with extractive sentences performs better on recall and f-measure, while the precision suffers a little.

	P	R	F	Summary Size
QA + SE	0.559	0.537	0.547	0.42
SE + QA	0.540	0.576	0.557	0.46

Table 8: Results showing Precision(P), Recall(R), F-measure(F), and Summary Size for non-QA threads from Annotator A's dataset with different approaches

Finally, when we combine the results of summarizing QA threads with question-answer pair sentences augmented with extractive sentences and the results of summarizing non-QA threads with extractive sentences augmented with question-answer pair sentences, namely the first row of Table 7 and the second row of Table 8, we get the best overall f-measure score.

	P	R	F	Summary Size
Combination	0.543	0.594	0.567	0.45

Table 9: Combined results for non-QA threads with SE+QA summarization and QA threads with QA+SE summarization from Annotator A's dataset

6 Postprocessing Extracted Sentences

Extracted sentences are sent to a module that wraps these sentences with the names of the senders, the dates at which they were sent, and a speech act verb chosen according to the type of the sentence. For example, if a sentence figures as a question, the wrapper would use the speech act verb "asked", and if a sentence is an answer to a question "answered" speech act verb would be used. Otherwise, verbs such as "wrote" and "mentioned" would be used. The wrapper also tries to show the relation between two adjacent sentences in the summary. For example, if two adjacent sentences are from the same email, the wrapper for the second sentence would say "In the same email message,". If the two adjacent sentences are from two adjacent emails in the thread, the wrapper for the second sentence would say "Responding to this,". And, if the second sentence is from an email in a different sub-thread, meaning they share the same ancestor email but the first is not the ancestor of the second, the wrapper for the second sentence would say "In another sub-thread,". And, finally, if the two adjacent sentences are from emails in the same sub-thread, the wrapper for the second sentence would say "In a subsequent message in the same thread,". Furthermore, for readability, the sentences are sorted by the order in which they appear in the email thread. The wrapper for the foremost sentence in the summary also states the subject of the thread. For example, ""Regarding "meeting on sunday", on April 4, 2005, James asked, "When is the meeting at?". Responding to this, on April 4, 2005, Rita replied, "at 12 pm."."

We have also developed a system for on-the-fly email summarization of email conversations that can be seamlessly integrated into a user's existing email client such as Microsoft Outlook. Our implementation of the email summarization interface employs a client-server architecture; the client portion of the model resides in a user's email client while the multi-user capable server can be run anywhere, and most possibly in a dedicated host in the network.

7 Conclusion and Future Work

We presented various approaches to integrating automatically detected question-answer pairs of threads of email conversations with their extractive summaries all of which outperform machine learning based extractive summarization without

consideration of question-answer pair data. We saw the best f-measure performance using the model in which we categorize an email conversation thread with question-answer exchanges according to whether the question-answer exchanges are central to the conversation and applying separate approach to their summarization according to their category. Thus, for our email client we categorize an email thread into whether it is a QA thread or not. If it is a QA thread, we augment the question-answer pair sentences with extractive sentences. Otherwise, we augment the extractive sentences with question-answer pair sentences. We also presented our approach to wrapping these extractive sentences to generate summaries for email conversations that are devoted to question-answer exchanges along with a description of a system to summarize and categorize email threads.

In future work, we intend to perform an evaluation of the approaches we have identified here based on human feedback. While the approaches we have identified attempt to learn the process by which our annotators wrote their summaries, a difficult task as evident from our performance scores, we think that our use of extractive sentences for summarization can be further refined by learning extractive approaches that identify sub-sentence level content for summarization, like removing redundant or unwanted clauses from a sentence. Furthermore, use of abstraction in summarization is also an interesting area of research to us. In cases where multiple answers were offered to an opinion question, for example, the detection of agreement and disagreement in these answers can be used to generate an abstract summary of such question-answer exchanges.

References

Anonymous-A. 2004. Citation obscured to conceal identity of authors.

Anonymous-B. 2004. Citation obscured to conceal identity of authors.

Sasha Blair-Goldensohn, David Evans, Vasileios Hatzivassiloglou, Kathleen McKeown, Ani Nenkova, Rebecca Passonneau, Barry Schiffman, Andrew Schlaikjer, Advaith Siddharthan, and Sergei Siegelman. 2004. Columbia university at duc 2004. In *4th Document Understanding Conference 2004 (DUC 2004)*.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

William Cohen. 1996. Learning trees and rules with set-valued features. In *Fourteenth Conference of the American Association of Artificial Intelligence*. AAAI.

Angelo Dalli, Yunqing Xia, and Yorick Wilks. 2004. Fasil email summarisation system. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Vasileios Hatzivassiloglou, Judith Klavans, Melissa Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. 2001. SimFinder: A flexible clustering tool for summarization. In *Proceedings of the NAACL Workshop on Automatic Summarization*, Pittsburgh, PA.

Derek Lam, Steven L. Rohall, Chris Schmandt, and Mia K. Stern. 2002. Exploiting e-mail structure to improve summarization. In *ACM 2002 Conference on Computer Supported Cooperative Work (CSCW2002), Interactive Posters*, New Orleans, LA.

Smaranda Muresan, Evelyne Tzoukermann, and Judith Klavans. 2001. Combining Linguistic and Machine Learning Techniques for Email Summarization. In *Proceedings of the CoNLL 2001 Workshop at the ACL/EACL 2001 Conference*.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. In *NAACL-HLT 2004*.

Ani Nenkova, Barry Schiffman, Andrew Schlaiker, Sasha Blair-Goldensohn, Regina Barzilay, Sergey Sigelman, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2003. Columbia university at duc 2003. In *3rd Document Understanding Conference 2003 (DUC 2003)*.

Paula Newman and John Blitzer. 2003. Summarizing archived discussions: a beginning. In *Proceedings of Intelligent User Interfaces*.

Simone Teufel and Hans van Halteren. 2004. Evaluating information content by factoid analysis: human annotation and stability. In *EMNLP-04*.

Klaus Zechner. 2002. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*, 28(4):447–485.