Using Question-Answer Pairs in Extractive Summarization of Email Conversations

Lokesh Shrestha, Kathleen McKeown and Owen Rambow
Columbia University
New York, NY, USA
{lokesh, kathy, rambow}@cs.columbia.edu

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 appraoches to integrating question-answer pairs in the extractive summaries, and show that their use improves the quality of email summaries. We also describe the email summarization interface we have developed that allows users to summarize email conversations from email clients such as Mircosoft Outlook.

1 Introduction

In this paper, we discuss work on summarizing threads of email conversations, i.e., coherent exchanges of email messages among several participants. In (Rambow et al., 2004) we showed that sentence extraction techniques can work for summarizing email threads, but profit from email-specific features. In addition, we showed that the presentation of the summary should take into account the dialogic structure of email communication. However, the extractive summaries suffer from the possibility of incomplete summaries in cases where an extractive summary may not include the answer to a question included in the summary. In (Shrestha and McKeown, 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 paper, we present research that builds on these techniques to integrate question-answer pairs in extractive summaries of email conversations, and show that such an integrative approach improves the quality of summarization.

While summarization of email conversations seems a natural way to improve upon current methods of email management, research on email summarization is in early stages. Consider an example summary of a thread of email conversation shown in Figure 1 which was produced by the sentence extraction based email thread summarization system described in (Rambow et al., 2004). While this summary does include an answer to the first question, it does not include answers to the two questions posed subsequently even though the answers are present in the thread. This example demonstrates one of the inadequacies of sentence extraction based summarization modules: namely, the absence of sentences that would have made the summaries more readable and complete.

Further, 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-

Regarding “acm home/bjarney”, on Apr 9, 2001, Muriel Danslop wrote:
Two things: Can someone be responsible for the press releases for Stroustrup?
Responding to this on Apr 10, 2001, Theresa Feng wrote:
I think Phil, who is probably a better writer than most of us, is writing up something for dang and Dave to send out to various ACM chapters. Phil, we can just use that as our “press release”, right?
In another subthread, on Apr 12, 2001, Kevin Danquoit wrote:
Are you sending out upcoming events for this week?

Figure 1: Sample summary obtained with sentence extraction

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 exchanges, a summary that can highlight the main question(s) asked and the response(s) given would be useful.

2 Previous and Related Work

(Muresan et al., 2001) describe work on summarizing individual email messages using machine learning approaches to learn rules for salient noun phrase extraction. In contrast, our work aims at summarizing whole threads and at capturing the interactive nature of email.

(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. Our work, in contrast, aims at summarizing the whole thread, and we are precisely interested in changing the summarization algorithm itself, not in using a black box summarizer.

(Nenkova and Bagga, 2003) present work on generating extractive summaries of threads in archived discussions. Sentences from the root message and from each response to the root extracted using *ad-hoc* algorithms crafted by hand. This approach works best when the subject of the root email best describes the “issue” of the thread, and when the root email does not discuss more than one issue. In our work, we do not make any assumptions about the nature of the email, and learn sentence extraction strategies using machine learning.

(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. While the approach is quite different from ours (due to the underlying clustering algorithm and the absence of machine learning to select features), the use of email-specific features, in particular the feature related to quoted material, is similar.

(Dalli et al., 2004) describe FASIL, an email summarization system for use in a voice-based Virtual Personal Assistant developed at University of Sheffiled. The system uses a ranking function that uses the occurence 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 by selecting the sentences that are most dissimilar to those sentences already included 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. However, overall most of the issues tackled

by (Zechner, 2002) are not relevant to email summarization.

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.

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 did not provide instructions about how to choose content for the summaries, but we did instruct the annotators on the format of the summary; specifically, we requested 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.

Writing summaries is not a task that competent native speakers are necessarily good at without specific training. Furthermore, there may be many different possible summary types that address different needs, and different summaries may satisfy a particular need. Thus, when asking native speakers to write thread summaries we cannot expect to obtain summaries that are similar.

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

The manually written summaries by the annotators were then used in the manner described in (Rambow et al., 2004) to obtain new sentence extraction rules based on the question-answer threads at new compression rates. Specifically, we used the handwritten summaries to identify important sentences in the threads in the following manner. 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 (Rambow et al., 2004) assumes a compression rate of 80%, 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 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 about 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 Figure 1.

Recall measures the proportion of the positive gold standard sentences that are correctly categorized using the SimFinder scores. Precision measures the proportion of the positively categorized sentences that are positive gold standard sentences. F-measure is the harmonic mean of recall and precision. 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 also. These observations suggest that the best compression to use would be 55 percent (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.

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.

[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.

4 Extractive Summarization and Question-Answer Pair Detection

4.1 Extractive Summarization

This section describes experiments with machine learning techniques for extractive summarization of threads of email conversation as described in (Rambow et al., 2004) as well as new research that validates our approach. 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%. 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 42% and that in annotator B's data set is about 39%. Table 3 gives the baseline results for each of the two data sets. The baselines are obtained through random classification, i.e., randomly classify each sentence in the data set such that the number of random positive classifications is the same as the number of positive classifications. This results

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

in the same number of false positives and false negatives and, thus, in an equal scores of recall, precision and f-measure. [2] Further, it can be seen that the baseline f-measure scores approximately the same as the summary sizes of the two data sets (502/1174 = 0.43 and 342/876 = 0.39).

	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

	Precision	**Recall**	**F-measure**
Annotator A	0.422	0.422	0.422
Annotator B	0.392	0.392	0.392

Table 3: Baseline scores for the two data sets obtained through random classification

	Precision	**Recall**	**F-measure**
Annotator A	0.550	0.516	0.532
Annotator B	0.514	0.468	0.490

Table 4: Sentence extraction results using the full feature set at 55% compression

Table 4 shows the results for extractive summarization of email threads with 5-fold cross validation using the full feature set described in (Rambow et al., 2004). This feature set includes the standard set of features such as sentence length, TF-IDF weights, relative position in thread, as well as features derived from the structure of email thread and from the email headers. 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, which is expected as we gave no clear guidelines to the annotators for writing the summaries. This difference in style and content could affect the learnability of sentence extraction rules.

4.2 Question-Answer Pair Detection

This section summarizes automatic learning of rules for question-answer pairing in threads of email conversation as described in (Shrestha and McKeown, 2004) and applies them to our new data. The rules 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. As described in (Shrestha and McKeown, 2004), we obtained one set of training data by taking the union of the annotations of the two annotators mentioned in Section 3.2. This data set was further 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. We show in (Shrestha and McKeown, 2004) that learning sep-

[2] Because the number of positive classifications in the random classification is the same as the number of positive classifications in the training data, each false positive random prediction introduces a false negative prediction. And, because recall = $\frac{\text{true positives}}{\text{true positives + false negatives}}$ and precision = $\frac{\text{true positives}}{\text{true positives + false positives}}$, recall, precision, and, therefore, f-measure scores are the same.

arate rules for the different subset of training data results in better performance. 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

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 a feature in our machine learning-based extractive summarization approach. The second approach is to add automatically detected answers to questions that appear in the extractive summaries. This approach can be further developed by adding questions whose answers appear in the extractive summaries. 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 5 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 Figure 4 for annotator A's data set primarily due an improvement in precision, an improvement is not seen for annotator B's data set.

	Precision	**Recall**	**F-measure**
Annotator A	0.591	0.506	0.545
Annotator B	0.502	0.459	0.479

Table 5: Results with adding an "answer" feature in extractive summarization

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. The results for this experiment are presented in Table 6. As the results show, when compared with the results in Table 4, we get a marked improvement for both the precision and recall scores, hence the f-measures scores too, for both the data sets.

	Precision	**Recall**	**F-measure**
Annotator A	0.564	0.562	0.563
Annotator B	0.523	0.502	0.513

Table 6: Results with sentence extraction augmented with answer sentences for extracted question sentences

This approach can be further improved. 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 7. As can be seen when these results are compared with those in Table 6, 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.

	Precision	**Recall**	**F-measure**
Annotator A	0.561	0.571	0.566
Annotator B	0.523	0.532	0.528

Table 7: Results with sentence extraction augmented with answer sentences for extracted question sentences and question sentences for extracted answer sentences

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 8. These results show that we get an improvement in recall over all of the approaches described above. Though precision suffers, the overall f-measure score is the best we get.

	Precision	**Recall**	**F-measure**
Annotator A	0.534	0.617	0.573
Annotator B	0.501	0.608	0.550

Table 8: Results with question-answer pairs augmented with extractive sentences

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. The speech act verb is chosen as a function of the structure of the email thread in order to make this structure more apparent to the reader. Furthermore, for readability, the sentences are sorted by the order in which they appear in the email thread.

7 Email Summarization Interface

We have developed a system for on-the-fly email categorization and 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. The server accepts connections from the email client of any user, and upon authentication starts a session of client-server communication. During the duration of the session, the client and server each communicate with the other through XML formatted text messages. These messages tell the server which commands to invoke, and the client what the outcome of its requests are.

The client can make various requests such as categorization of individual email, categorization of an email thread, summarization of individual email, summarization of an email thread, and submission of an email for preprocessing. When new email arrives in a person's mailbox, these emails will be sent to the server for preprocessing. Preprocessing involves processing of the content and the headers of the email for future use in on-the-fly summarization and categorization. Currently, preprocessing involves extraction of email headers and content, removal of signatures, quoted material and greetings from the content of the email body, sentence boundary detection of the email body, part of speech tagging of the content of the email body, lemmatization of the content of the email body, creation of the email thread using the references to previous emails, and the categorization of individual emails. Preprocessing is especially a necessity for the summarization and categorization of email threads. Because an email thread might increase in size in time as new email messages arrive, with preprocessed data readily available, on-the-fly summarization of threads requires far less time than otherwise. A sample session is shown in Figure 2.

8 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 start with all question-answer pairs as the basis for the summary, and then add additional sentences as identified by the extraction module. We get better precision and shorter summaries using the model in which we start with all the extractive sentences and add sentences from the question-answer pairs. Thus, for our email client we used this approach for email threads containing question-answer pairs, and extractive summarization for other threads. 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

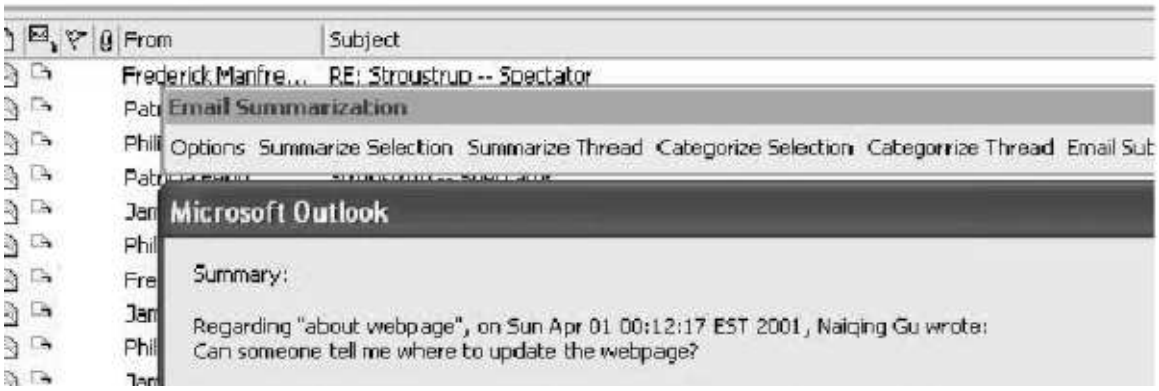

Figure 2: The interface to our email summarizer in Microsoft Outlook, showing the new taskbar and a summarization window

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 to obtain better results. 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

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

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 Amit Bagga. 2003. Facilitating email thread access by extractive summary generation. In *Proceedings of RANLP*, Bulgaria.

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

Owen Rambow, Lokesh Shrestha, John Chen, and Christy Lauridsen. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL 2004 Short*, Boston, USA.

Lokesh Shrestha and Kathleen McKeown. 2004. Detection of question-answer pairs in email conversations. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

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