Automatic Summarization of Open-Domain Multiparty Dialogues in Diverse Genres

Klaus Zechner*
Educational Testing Service

Automatic summarization of open-domain spoken dialogues is a relatively new research area. This article introduces the task and the challenges involved and motivates and presents an approach for obtaining automatic-extract summaries for human transcripts of multiparty dialogues of four different genres, without any restriction on domain.

We address the following issues, which are intrinsic to spoken-dialogue summarization and typically can be ignored when summarizing written text such as news wire data: (1) detection and removal of speech disfluencies; (2) detection and insertion of sentence boundaries; and (3) detection and linking of cross-speaker information units (question-answer pairs).

*A system evaluation is performed using a corpus of 23 dialogue excerpts with an average duration of about 10 minutes, comprising 80 topical segments and about 47,000 words total. The corpus was manually annotated for relevant text spans by six human annotators. The global evaluation shows that for the two more informal genres, our summarization system using dialogue-specific components significantly outperforms two baselines: (1) a maximum-marginal-relevance ranking algorithm using TF*IDF term weighting, and (2) a* LEAD *baseline that extracts the first* n *words from a text.*

1. Introduction

Although the field of summarizing written texts has been explored for many decades, gaining significantly increased attention in the last five to ten years, summarization of spoken language is a comparatively recent research area. As the number of spoken audio databases is growing rapidly, however, we predict that the need for high-quality summarization of information contained in this medium will increase substantially. Summarization of spoken dialogues, in particular, may aid in the archiving, indexing, and retrieval of various records of oral communication, such as corporate meetings, sales interactions, or customer support.

The purpose of this article is to explore the issues of spoken-dialogue summarization and to describe and evaluate an implementation addressing some of the core challenges intrinsic to the task. We will use an implementation of a state-of-the-art text summarization method (maximum marginal relevance, or MMR) as the main baseline for comparative evaluations, and then add a set of components addressing issues specific to spoken dialogues to this MMR module to create our spoken dialogue summarization system, which we call DIASUMM.

We consider the following dimensions to be relevant for our research; the combination of these dimensions distinguishes our work from most other work in the field

* Educational Testing Service, Rosedale Road MS 11-R, Princeton, NJ 08541. E-mail: kzechner@ets.org

© 2002 Association for Computational Linguistics

of summarization:

- spoken versus written language
- multiparty dialogues versus texts written by one author
- unrestricted versus restricted domains
- diverse genres versus a single genre

The main challenges this work has to address, in addition to the challenges of written-text summarization, are as follows:

- coping with speech disfluencies
- identifying the units for extraction
- maintaining cross-speaker coherence
- coping with speech recognition errors

We will discuss these challenges in more detail in the following section. Although we have addressed the issue of speech recognition errors in previous related work (Zechner and Waibel 2000b), for the purpose of this article, we exclusively use human transcripts of spoken dialogues.

Intrinsic evaluations of text summaries usually use sentences as their basic units. For our data, however, sentence boundaries are typically not available in the first place. Thus we devise a word-based evaluation metric derived from an average relevance score from human relevance annotations (section 6.2).

The organization of this article is as follows: Section 2 provides the motivation for our research, introducing and discussing the main challenges of spoken-dialogue summarization, followed by a section on related work (section 3). Section 4 describes the corpus we use to develop and evaluate our system, along with the procedures employed for corpus annotation. The system architecture and its components are described in detail in section 5, along with evaluations thereof. Section 6 presents the global evaluation of our approach, before we conclude the article with a discussion of our results, contributions, and directions for future research in this field (sections 7 and 8).

2. Motivation

Consider the following example from a phone conversation drawn from the English CALLHOME database (LDC 1996). It is a transcript of a conversation between two native speakers of American English; one person is in the New York area (speaker *a*), the other one (speaker *b*) in Israel. It was recorded about a month after Yitzhak Rabin's assassination (1995). This dialogue segment is about one minute of real time. The audio is segmented into speaker turns using silence heuristics,[1] and each turn is marked with a turn number and with the speaker label. Noises are removed to increase readability.[2]

1 Therefore, in some cases, we can find several turns of one speaker following each other.
2 Hence there can be "missing" turns (e.g., turn 37), in case they contain only noises and no actual words.

```
28 a: oh
29 b: they didn't know he was going to get shot but it
      was at a peace rally so i mean it just worked out
30 b: i mean it was a good place for the poor guy to die
      i mean because it was you know right after the rally
      and everything was on film and everything
31 a: yeah
32 b: oh the whole country we just finished the thirty days
      mourning for him now you know it's uh oh everybody's
      still in shock it's
33 a: oh
34 a: i know
35 b: terrible what's going on over here
36 b: and this guy that killed him they show him on t v
      smiling he's all happy he did it and everything he
      isn't even sorry or anything
38 a: there are i
39 b: him him he and his brother you know the two of
      them were in it together and there's a whole group
      now it's like a a conspiracy oh it's eh
40 a: mm
41 a: with the kahane chai
42 b: unbelievable
43 b: yeah yeah it's all those people yeah you probably see
      them running around new york don't you they're all
44 a: yeah
45 a: oh yeah they're here
46 b: new york based yeah
47 a: oh there's
48 a: all those fanatics
49 a: like the extreme
50 b: oh but
51 b: but wh- what's the reaction in america really i mean
      i mean do people care you know i mean you know do they
52 a: yeah mo- most pe- i mean uh
53 a: i don't know what commu- i mean like the jewish community
54 a: a lot e- all of us were
55 a: very upset and there were lots all the
56 b: yeah
57 a: like two days after did it happen like on a sunday
58 b: yeah it hap- it happened on it happened on a saturday night
```

By looking at this transcript we can readily identify some of the phenomena that would cause difficulties for conventional summarizers of written texts:

- Some turns (e.g., turn 51) contain many disfluencies that (1) make them hard to read and (2) reduce the relevance of the information contained therein.
- Some (important) pieces of information are distributed over a sequence of turns (e.g., turns 53–54–55, 45–47–48–49); this is due to a silence-based

segmentation algorithm that causes breaks in logically connected clauses. A traditional summarizer might render these sequences incompletely.

- Some turns are quite long (e.g., 36, 39) and contain several sentences; a within-turn segmentation seems necessary to avoid the extraction of too much extraneous information when only parts of a turn contain relevant information.
- Some of the information is constructed interactively by both speakers; the prototypical cases are question-answer pairs (e.g., turns 51–52ff., turns 57–58). A traditional text summarizer might miss either question or answer and hence produce a less meaningful summary.

We shall discuss these arising issues along with an indication of our computational remedies in the following subsections. We want to stress beforehand, though, that the originality of our system should not be seen in the particular implementation of its individual components, but rather in their selection and specific composition to address the issues at hand in an effective and also efficient way.

2.1 Disfluency Detection

The two main negative effects speech disfluencies have on summarization are that they (1) decrease the readability of the summary and (2) increase its noncontent noise. In particular for informal conversations, the percentage of disfluent words is quite high, typically around 20% of the total words spoken.[3] This means that this issue should, in our opinion, be addressed to improve the quality (readability and conciseness) of the generated summaries.

In section 5.3 we shall present three components for identifying most of the major classes of speech disfluencies in the input of the summarization system, such as filled pauses, repetitions, and false starts. All detected disfluencies are marked in this process and can be selectively excluded during summary generation.

2.2 Sentence Boundary Detection

Unlike written texts, in which punctuation markers clearly indicate clause and sentence boundaries, spoken language is generated as a sequence of streams of words, in which pauses (silences between words) do not always match linguistically meaningful segments: A speaker can pause in the middle of a sentence or even a phrase, or, on the other hand, might not pause at all after the end of a sentence or clause.

This mismatch between acoustic and linguistic segmentation is reflected in the output of a speech recognizer, which typically generates a sequence of speaker turns whose boundaries are marked by periods of silence (or nonspeech). As a result, one speaker's turn may contain multiple sentences, or, on the other hand, a speaker's sentence might span more than one turn. In a test corpus of five English CALLHOME dialogues with an average length of 320 turns, we found on average of about 30 such continuations of logical clauses over automatically determined acoustic segments per dialogue.

The main problems for a summarizer would thus be (1) the lack of coherence and readability of the output because of incomplete sentences and (2) extraneous information due to extracted units consisting of more than one sentence. In section 5.4 we

3 Although other studies have found percentages lower than this figure, we included content-less categories such as discourse markers or rhetorical connectives, which are often not regarded as disfluencies per se.

describe a component for sentence segmentation that addresses both of these problems.

2.3 Distributed Information

Since we have multiparty conversations as opposed to monologues, sometimes the crucial information is found in a sequence of turns from several speakers, the prototypical case of this being a question-answer pair. If the summarizer were to extract only the question or only the answer, the lack of the corresponding answer or question would often cause a severe reduction of coherence in the summary.

In some cases, either the question or the answer is very short and does not contain any words with high relevance that would yield a substantial weight in the summarizer. In order not to lose these short sentences at a later stage, when only the most relevant sentences are extracted, we need to identify matching question-answer pairs ahead of time, so that the summarizer can output the matching sentences during summary generation as one unit. We describe our approach to cross-speaker information linking in section 5.5.

2.4 Other Issues

We see the work reported in this article as the first in-depth analysis and evaluation in the area of open-domain spoken-dialogue summarization. Given the large scope of this undertaking, we had to restrict ourselves to those issues that are, in our opinion, the most salient for the task at hand.

A number of other important issues for summarization in general and for speech summarization in particular are either simplified or not addressed in this article and left for future work in this field. In the following, we briefly mention some of these issues, indicating their potential relevance and promise.

2.4.1 Topic Segmentation. In many cases, spoken dialogues are multitopical. For the English CALLHOME corpus, we determined an average topic length of about one to two minutes' speaking time (or about 200–400 words). Summarization can be accomplished faster and more concisely if it operates on smaller topical segments rather than on long pieces of input consisting of diverse topics.

Although we have implemented a topic segmentation component as part of our system for these reasons, all of the evaluations are based on the topical segments determined by human annotators. Therefore, this component will not be discussed in this article. Furthermore, topical segmentation is not an issue intrinsic to spoken dialogues, which in our opinion justifies this simplification.

2.4.2 Anaphora Resolution. An analogous reasoning holds for the issue of anaphora resolution: Although it would certainly be desirable, for the sake of increased coherence and readability, to employ a well-working anaphora resolution component, this issue is not specific to the task at hand, either. One could argue that particularly for summarization of more informal conversations, in which personal pronouns are rather frequent, anaphora resolution might be more helpful than for, say, summarization of written texts. But we conjecture that this task might also prove more challenging than written-text anaphora resolution. In our system, we did not implement a module for anaphora resolution.

2.4.3 Discourse Structure. Previous work indicates that information about discourse structure from written texts can help in identifying the more salient and relevant sentences or clauses for summary generation (Marcu 1999; Miike et al. 1994). Much

less exploration has been done, however, in the area of automatic analysis of discourse structure for non-task-oriented spoken dialogues in unrestricted domains, such as CALLHOME (LDC 1996). Research for those kinds of corpora reported in Jurafsky et al. (1998), Stolcke et al. (2000), Levin et al. (1999), and Ries et al. (2000) focuses more on detecting localized phenomena such as speech acts, dialogue games, or functional activities. We conjecture that there are two reasons for this: (1) free-flowing spontaneous conversations have much less structure than task-oriented dialogues, and (2) the automatic detection of hierarchical structure would be much harder than it is for written texts or dialogues based on a premeditated plan.

Although we believe that in the long run attempts to automatically identify the discourse structure of spoken dialogues may benefit summarization, in this article, we greatly simplify this matter and exclusively look at local contexts in which speakers interactively construct shared information (the question-answer pairs).

2.4.4 Speech Recognition Errors. Throughout this article, our simplifying assumption is that our input comes from a perfect speech recognizer; that is, we use human textual transcripts of the dialogues in our corpus. Although there are cases in which this assumption is justifiable, such as transcripts provided by news services in parallel to the recorded audio data, we believe that in general a spoken dialogue summarizer has to be able to accept corrupted input from an automatic speech recognizer (ASR), as well. Our system is indeed able to work with ASR output; it is integrated in a larger system (Meeting Browser) that creates, summarizes, and archives meeting records and is connected to a speech recognition engine (Bett et al. 2000). Further, we have shown in previous work how we can use ASR confidence scores (1) to reduce the word error rate within the summary and (2) to increase the summary accuracy (Zechner and Waibel 2000b).

2.4.5 Prosodic Information. A further simplifying assumption of this work is that prosodic information is not available, with the exception of start and end times of speaker turns. Considering the results reported by Shriberg et al. (1998) and Shriberg et al. (2000), we conjecture that future work in this field will demonstrate the additional benefit of incorporating prosodic information, such as stress, pitch, and intra-turn pauses, into the summarization system. In particular, we would expect improved system performance when speech recognition hypotheses are used as input: In that case, the prosodic information could compensate to some extent for incorrect word information.

3. Related Work

The vast majority of summarization research in the past clearly has focused exclusively on written text. A good selection of both early seminal papers and more recent work can be found in Mani and Maybury (1999). In general, most summarization approaches can be classified as either corpus-based, statistical summarization (such as Kupiec, Pedersen, and Chen [1995]), or knowledge-based summarization (such as Reimer and Hahn [1988]) in which the text domain is restricted. (The MMR method [Carbonell, Geng, and Goldstein 1997], which we are using as the summarization engine for our DIASUMM system, belongs to the first category.) More recently, Marcu (1999) presented work on using automatically detected discourse structure for summarization. Knight and Marcu (2000) and Berger and Mittal (2000) presented approaches in which summarization can be reformulated as a problem of machine translation:

translating a long sentence into a shorter sentence, or translating a Web page into a brief gist, respectively.

Two main areas are exceptions to the focus on text summarization in past work: (1) summarization of task-oriented dialogues in restricted domains and (2) summarization of spoken news in unrestricted domains. We shall discuss both of these areas in the following subsections, followed by a discussion of prosody-based emphasis detection in spoken language, and finally by a summary of research most closely related to the topic of this work.

3.1 Summarization of Dialogues in Restricted Domains

During the past decade, there has been significant progress in the area of closed-domain spoken-dialogue translation and understanding, even with automatic speech recognition input. Two examples of systems developed in that time frame are JANUS (Lavie et al. 1997) and VERBMOBIL (Wahlster 1993).

In that context, several spoken-dialogue summarization systems have been developed whose goal it is to capture the essence of the task-based dialogues at hand. The MIMI system (Kameyama and Arima 1994; Kameyama, Kawai, and Arima 1996) deals with the travel reservation domain and uses a cascade of finite-state pattern recognizers to find the desired information. Within VERBMOBIL, a more knowledge-rich approach is used (Alexandersson and Poller 1998; Reithinger et al. 2000). The domain here is travel planning and negotiation of a trip. In addition to finite-state transducers for content extraction and statistical dialogue act recognition, VERBMOBIL also uses a dialogue processor and a summary generator that have access to a world knowledge database, a domain model, and a semantic database. The abstract representations built by this summarizer allow for summary generation in multiple languages.

3.2 Summarization of Spoken News

Within the context of the Text Retrieval Conference (TREC) spoken document retrieval (SDR) conferences (Garofolo et al. 1997; Garofolo et al. 1999) as well as the recent Defense Advanced Research Project Agency (DARPA) broadcast news workshops, a number of research groups have been developing multimedia browsing tools for text, audio, and video data, which should facilitate the access to news data, combining different modalities.

Hirschberg et al. (1999) and Whittaker et al. (1999) present a system that supports local navigation for browsing and information extraction from acoustic databases, using speech recognizer transcripts in tandem with the original audio recording. Although their interface helps users in the tasks of relevance ranking and fact finding, it is less helpful in the creating of summaries, partly because of imperfect speech recognition.

Valenza et al. (1999) present an audio summarization system that combines acoustic confidence scores with relevance scores to obtain more accurate and reliable summaries. An evaluation showed that human judges preferred summaries with a compression rate of about 15% (30 words per minute at a speaking rate of about 200 words per minute) and that the summary word error rate was significantly smaller than the word error rate for the full transcript.

Hori and Furui (2000) use salience features in combination with a language model to reduce Japanese broadcast news captions by about 30–40% while keeping the meaning of about 72% of all sentences in the test set. Another speech-related reduction approach was presented recently by Koumpis and Renals (2000), who summarize voice mail in the Small Message format.

3.3 Prosody-Based Emphasis Detection in Spoken Audio

Whereas most approaches to summarizing acoustic data rely on the word information (provided by a human or ASR transcript), there have been attempts to generate summaries based on emphasized regions in a discourse, using only prosodic features. Chen and Withgott (1992) train a hidden Markov model on transcripts of spontaneous speech, labeled for different degrees of emphasis by a panel of listeners. Their "audio summaries" on an unseen (but rather small) test set achieve a remarkably good agreement with human annotators ($\kappa > 0.5$). Stifelman (1995) uses a pitch-based emphasis detection algorithm developed by Arons (1994) to find emphasized passages in a 13-minute discourse. In her analysis, she finds good agreement between these emphasized regions and the beginnings of manually marked discourse segments (in the framework of Grosz and Sidner [1986]). Although these are promising results, being suggestive of the role of prosody for determining emphasis, relevance, or salience in spoken discourse, in this work we restrict the use of prosody to the turn length and interturn pause features. We conjecture, however, that the integration of prosodic and word level information would be a fruitful research area that would have to be explored in future work.

3.4 Spoken Dialogue Summarization in Unrestricted Domains

Waibel, Bett, and Finke (1998) report results of their summarizer on automatically transcribed SWITCHBOARD (SWBD) data (Godfrey, Holliman, and McDaniel 1992), the word error rate being about 30%. Their implementation used an algorithm inspired by MMR, but they did not address any dialogue- or speech-related issues in their summarizer. In a question-answer test with summaries of five dialogues, participants could identify most of the key concepts using a summary size of only five turns. These results varied widely (between 20% and 90% accuracy) across the five different dialogues tested in this experiment.

Our own previous work (Zechner and Waibel 2000a) addressed for the first time the combination of challenges of dialogue summarization with summarization of spoken language in unrestricted domains. We presented a first prototype of DIASUMM that addressed the issues of disfluency detection and removal and sentence boundary detection, as well as cross-speaker information linking.

This work extends and expands these initial attempts substantially, in that we are now focusing on (1) a systematic training of the major components of the DIASUMM system, enabled by the recent availability of a large corpus of disfluency-annotated conversations (LDC 1999b), and (2) the exploration of three more genres of spoken dialogues in addition to the English CALLHOME corpus (NEWSHOUR, CROSSFIRE, GROUP MEETINGS). Further, the relevance annotations are now performed by a set of six human annotators, which makes the global system evaluation more meaningful, considering the typical divergence among different annotators' relevance judgments.

4. Data Annotation

4.1 Corpus Characteristics

Table 1 provides the statistics on the corpus used for the development and evaluation of our system. We use data from four different genres, two being more informal, two more formal:

- English CALLHOME and CALLFRIEND: from the Linguistic Data Consortium (LDC) collections, eight dialogues for the `devtest` set

Table 1
Data characteristics for the corpus (average over dialogues). 8E-CH, 4E-CH: English CallHome; NHOUR: NewsHour; XFIRE: CrossFire; G-MTG: Group Meetings.

Data Set	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
Formal/informal	informal	informal	formal	formal	informal
Topics predetermined	no	no	yes	yes	yes
Dialogue excerpts (total)	8	4	3	4	4
Topical segments (total)	28	23	8	14	7
Different speakers	2.1	2	2	6	7.5
Turns	242	276	25	96	140
Sentences	280	366	101	281	304
Sentences per turn	1.2	1.3	4.1	2.9	2.2
Questions (in %)	3.7	6.4	6.3	9.8	4.0
False starts (in %)	12.1	11.0	2.0	7.2	13.9
Words	1685	1905	1224	3165	2355
Words per sentence	6.0	5.2	12.1	11.3	7.7
Disfluent (in %)	16.0	16.3	5.1	4.2	13.2
Disfluencies	222	259	48	95	266
Disfluencies per sentence	0.79	0.71	0.48	0.34	0.87
Empty coordinating conjunctions (in %)	30.3	30.4	64.8	50.7	24.3
Lexicalized filled pauses (in %)	18.8	21.0	17.2	23.5	13.9
Editing terms (in %)	3.6	1.6	3.4	5.7	3.3
Nonlexicalized filled pauses (in %)	20.8	29.9	0.7	2.3	29.5
Repairs (in %)	26.6	17.1	13.8	17.8	29.0

(8E-CH) and four dialogues for the `eval` set (4E-CH).[4] These are recordings of phone conversations between two family members or friends, typically about 30 minutes in length; the excerpts we used were matched with the transcripts, which typically represent 5–10 minutes of speaking time.

- NEWSHOUR (NHOUR): Excerpts from PBS's *NewsHour* television show with Jim Lehrer (recorded in 1998).
- CROSSFIRE (XFIRE): Excerpts from CNN's *CrossFire* television show with Bill Press and Robert Novak (recorded in 1998).
- GROUP MEETINGS (G-MTG): Excerpts from recordings of project group meetings in the Interactive Systems Labs at Carnegie Mellon University.

Furthermore, we used the Penn Treebank distribution of the SWITCHBOARD corpus, annotated with disfluencies, to train the major components of the system (LDC 1999b).

From Table 1 we can see that the two more formal corpora, NEWSHOUR and CROSSFIRE, have longer sentences, more sentences per turn, and fewer disfluencies (particularly nonlexicalized filled pauses and false starts) than English CALLHOME and the GROUP MEETINGS. This means that their flavor is more like that of written text and not so close to the conversational speech typically found in the SWITCHBOARD or CALLHOME corpora.

4 We used the `devtest` set corpus for system development and tuning and set aside the `eval` set for the final global system evaluation. For the other three genres, two dialogue excerpts each were used for the `devtest` set, the remainder for the `eval` set.

4.2 Corpus Annotation

4.2.1 First Annotation Phase. All the annotations were performed on human-generated transcripts of the dialogues. The CALLHOME and GROUP MEETINGS dialogues were automatically partitioned into speaker turns (by means of a silence heuristic); the other corpora were segmented manually (based on the contents and flow of the conversation).[5]

There were six naive human annotators performing the task;[6] only four, however, completed the entire set of dialogues. Thus, the number of annotations available for each dialogue varies from four to six. Prior to the relevance annotations, the annotators had to mark topical boundaries, because we want to be able to define and then create summaries for each topical segment separately (as opposed to a whole conversation consisting of multiple topics). The notion of a *topic* was informally defined as a region in the text that ends, according to the annotation manual, "when the speakers shift their topic of discussion."

Once the topical segments were marked, for each such segment, each annotator had to identify the most relevant information units (IUs), called *nucleus IUs*, and somewhat relevant IUs, called *satellite IUs*. IUs are often equivalent to sentences but can span longer or shorter contiguous segments of text, dependent on the annotator's choice. The overall goal of this relevance markup was to create a concise and readable summary containing the main information present in the topical segment. Annotators were also asked to mark the most salient words within their annotated IUs with a +, which would render a summary with a somewhat more telegraphic style (+-marked words).

We also asked that the human annotators stay within a preset target length for their summaries: The +-marked words in all IUs within a topical segment should be 10–20% of all the words in the segment. The guideline was enforced by a checker program that was run during and after annotation of a transcript and that also ensured that no markup errors and no accidental word deletions occurred. We provide a brief example here (`n[`, `n]` mark the beginning and end of a nucleus IU, the phrase *they fly to Boston* was +-marked as the core content within this IU):

```
B: heck it might turn out that you know n[ if
 +they +fly in +to +boston i can n]
```

4.2.2 Creation of Gold-Standard Summaries. After the first annotation phase, in which each coder worked independently according to the guidelines described above, we devised a second phase, in which two coders from the initial group were asked to create a common-ground annotation, based on the majority opinion of the whole group. To construct such a majority opinion guideline automatically, we assigned weights to all words in nucleus IUs and satellite IUs and added all weights for all marked words of all coders for every turn.[7] The total turn weights were then sorted by decreasing value to provide a guide for the two coders in the second phase as to which turns they should focus their annotations on for the common-ground or gold-standard summaries.

5 This fact may partially account for NEWSHOUR and CROSSFIRE turns being longer than CALLHOME and GROUP MEETING turns.

6 *Naive* in this context means that they were nonexperts in linguistics or discourse analysis.

7 The weights were set as follows: nucleus IUs: 3.0 if +-marked, 2.0 otherwise; satellite IUs: 1.0 if +-marked, 0.5 otherwise.

Table 2
Nuclei and satellites: Length in tokens and relative frequency (in % of all tokens).

Annotator/ Data Set	Avg. Nuc. Length	Avg. Sat. Length	Nuc-Tokens (in %)	Nuc-+-Marked (in %)	Sat-Tokens (in %)	Sat-+-Marked (in %)
LB	12.993	13.732	11.646	8.558	5.363	3.818
BR	16.507	14.551	11.978	8.339	10.558	7.645
SC	20.720	14.093	29.412	18.045	6.517	4.796
RW	22.899	19.576	19.352	11.332	2.757	1.718
RC	23.741	18.553	43.573	15.434	12.749	0.333
JK	39.203	9.794	26.355	11.204	0.711	0.465
Gold	21.763	6.462	13.934	6.573	0.179	0.000
CALLHOME	17.108	13.099	21.962	11.003	5.126	1.932
NEWSHOUR	25.828	16.733	29.536	13.530	4.300	2.947
CROSSFIRE	33.923	22.132	21.705	10.615	1.853	0.976
MEETINGS	37.674	23.413	23.034	9.222	7.456	1.123
All Dialogues	23.152	16.173	22.796	10.807	4.665	1.636

Other than this guideline, the requirements were almost exactly identical to those in phase 1, except that (1) the pair of annotators was required to work together on this task to be able to reach a consensus opinion, and (2) the preset relative word length of the gold summary (10–20%) applied only to the nucleus IUs.

As for the topical boundaries, which obviously vary among coders, a list of boundary positions chosen by the majority (at least half) of the coders in the first phase was provided. In this gold-standard phase, the two coders mostly stayed with these suggestions and changed less than 15% of the suggested topic boundaries, the majority of which were minor (less than two turns' difference in boundary position).

4.2.3 General Annotation Analysis. Table 2 provides the statistics on the frequencies of the annotated nucleus and satellite IUs. We make the following observations:

- On average, about 23% of all tokens were assigned to a nucleus IU and 5% to a satellite IU; counting only the +-marked tokens, this reduces to about 11% and 2% of all tokens, respectively.
- The average total lengths of nuclei and satellites vary widely across corpora: between 17.1 (13.1) tokens for CALLHOME and 37.7 (23.4) tokens for GROUP MEETINGS data. This is most likely a reflection on the typical length of turns in the different subcorpora.
- A similar variation is also observed across annotators: between 12 and 40 tokens for nucleus-IUs and between 9 and 20 tokens for satellites. The granularity of IUs is quite different across annotators.
- Since some annotators mark a larger number of IUs than others, there is an even larger discrepancy in the relative number of words assigned to nucleus IUs and satellite IUs among the different annotators: 11–44% (nucleus IUs) and 0–13% (satellite IUs).
- The ratio of nucleus versus satellite tokens also varies greatly among the annotators: from about 1:1 to 40:1.

- The ratio of nucleus and satellite tokens that are +-marked varies greatly: between 36 and 77% for nucleus IUs and between 2 and 80% for satellite IUs.

From these observations, we conclude that merging the nucleus and satellite IUs into one class would yield a more consistent picture than keeping them separate. A similar argument can be made for the +-marked passages, in which we also find a quite high intercoder variation in relative +-marking. This led us to the decision of giving equal weight to any word in an IU, irrespective of IU type or marking, for the purpose of global system evaluation.

Finally, we conjecture that the average length of our extraction units should be in the 10–40 words range, which roughly corresponds to about 3–12 seconds of real time, assuming an average word length of 300 milliseconds. As a comparison, we note that Valenza et al. (1999) found summaries with 30-grams[8] working well in their experiments, a finding that is in line with our observations here on typical human IU lengths.

4.2.4 Intercoder Agreement. Agreement between coders (and between automatic methods and coders) has been measured in the summarization literature with quite a wide range of methods: Rath, Resnick, and Savage (1961) use Kendall's τ; Kupiec, Pedersen, and Chen (1995) (among many others) use percentage agreement; and Aone, Okurowski, and Gorlinsky (1997) (among others) use the notions of precision, recall, and F_1-score, which are commonly employed in the information retrieval community. Similarly, in the literature on discourse segmentation and labeling, a variety of different agreement measures have been used, including precision and recall (Hearst 1997; Passonneau and Litman 1997), Krippendorff's (1980) α (Passonneau and Litman 1997) and Cohen's (1960) κ (Carletta et al. 1997).

In this work, we use the two following metrics: (1) the κ-statistic in its extension for more than two coders (Davies and Fleiss 1982); and (2) precision, recall, and F_1-score.[9] We will discuss the κ-statistic first.

For intercoder agreement with respect to topical boundaries, agreement is found if boundaries fall within the same 50-word bin of a dialogue. Relevance agreements are computed at the word level. For relevance markings, we compute κ both for the three-way case (nucleus IUs, satellite IUs, unmarked) and the two-way case (any IUs, unmarked).[10] Topical-boundary agreement was not evaluated for two of the GROUP MEETINGS dialogues, in which only one of four annotators marked any text-internal topic boundary. We compute agreements for each dialogue separately and report the arithmetic means for the five subcorpora in Table 3. We observe that agreement for topical boundaries is much higher than for relevance markings. Furthermore, agreement is generally higher for CALLHOME and comparatively low for the GROUP MEETINGS corpus.

As a second evaluation metric, we compute precision, recall, and F_1-scores for the same four annotators and the same sets of subcorpora as before. For topical boundaries, a match means that the boundaries fall within ± 3 turns of each other, and for relevant

8 A 30-gram is a passage of text containing 30 adjacent words.

9 Precision is the ratio of correctly matched items over all items (boundaries, marked words); recall is the ratio of correctly matched items over all items that need to be matched; and the F_1-score combines precision (P) and recall (R) in the following way: $F_1 = \frac{2PR}{P+R}$.

10 These computations were performed for those four (out of six) annotators who completed the entire corpus markup.

Table 3
Intercoder annotation κ agreement for topical boundaries and relevance markings.

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG	Overall
Topical boundaries	0.503	0.402	0.256	0.331	0.174	0.384
Relevance markings (3 way)	0.147	0.161	0.123	0.089	0.040	0.117
Relevance markings (2 way)	0.157	0.169	0.124	0.100	0.046	0.126

Table 4
Intercoder annotation F_1-agreement for topical boundaries and relevance markings.

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG	Overall
Topical boundaries	.54	.44	.53	.38	.18	.45
Relevance markings (2 way)	.38	.39	.38	.32	.32	.36

words a match means that the two words to be compared are both in a nucleus or satellite IU. The results can be seen in Table 4.

4.2.5 Disfluency and Sentence Boundary Annotation. In addition to the annotation for topic boundaries and relevant text spans, the corpus was also annotated for speech disfluencies in the same style as the Penn Treebank SWITCHBOARD corpus (LDC 1999b). One coder (different from the six annotators mentioned before) manually tagged the corpus for disfluencies and sentence boundaries following the SWITCHBOARD disfluency annotation style book (Meteer et al. 1995).

4.2.6 Question-Answer Annotation. A final type of annotation was performed on the entire corpus to mark all questions and their answers, for the purpose of training and evaluation of the question-answer linking system component. Questions and answers were annotated in the following way: Every sentence that is a question was marked as either a Yes-No-question or a Wh-question. Exceptions were back-channel questions, such as "Is that right?"; rhetorical questions, such as "Who would lie in public?"; and other questions that do not refer to a propositional content. These were not supposed to be marked (even if they have an apparent answer), since we see the latter class of questions as irrelevant for the purpose of increasing the local coherence within summaries. For each Yes-No-question and Wh-question that has an answer, the answer was marked with its relative offset to the question to which it belongs. Some answers are continued over several sentences, but only the core answer (which usually consists of a single sentence) was marked. This decision was made to bias the answer detection module toward brief answers and to avoid the question-answer regions' getting too lengthy, at the expense of summary conciseness.

5. Dialogue Summarization System

5.1 System Architecture

The global system architecture of the spoken-dialogue summarization system presented in this article (DIASUMM) is depicted in Figure 1. The input data are a time-ordered sequence of speaker turns with the following quadruple of information: start time, end time, speaker label, and word sequence. The seven major components are executed sequentially, yielding a pipeline architecture.

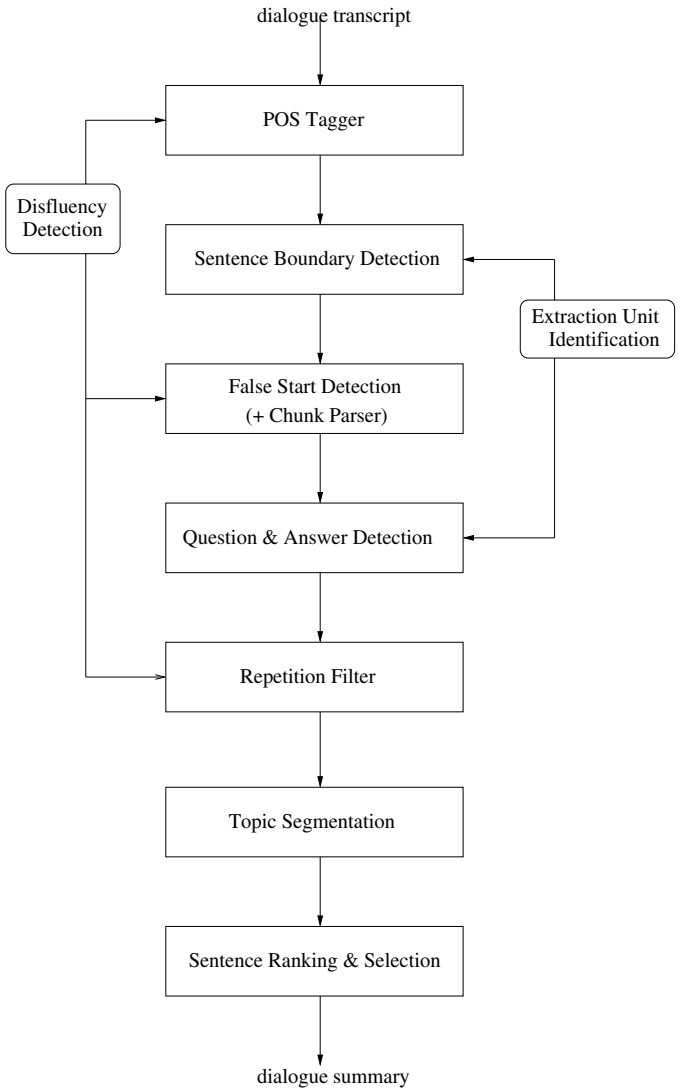

Figure 1
Global system architecture.

The following subsections describe the components of the system in more detail. As argued earlier, the topic detection component is not relevant for the way we conduct the global system evaluation and hence is not discussed here. (We implemented a variant of Hearst's [1997] TextTiling algorithm.) The three components involved in disfluency detection are the part-of-speech (POS) tagger, the false-start detection module, and the repetition filter. They are discussed in subsection 5.3, followed by a subsection on sentence boundary detection (5.4). The question-answer pair detection is described in subsection 5.5, and the sentence selection module, performing relevance ranking, in subsection 5.6.

5.2 Input Tokenization

We eliminate all human and nonhuman noises and incomplete words from the input transcript. Further, we eliminate all information on case and punctuation, since

we emulate the ASR output in that regard, which does not provide this information.

Contractions such as `don't` or `I'll` are divided and treated as separate words—in these examples we would obtain `do n't` and `I 'll`.

5.3 Disfluency Detection

5.3.1 Motivation. Conversational, informal spoken language is quite different from written language in that a speaker's utterances are typically much less well-formed than a writer's sentences. We can observe a set of disfluencies such as false starts, hesitations, repetitions, filled pauses, and interruptions. Additionally, in speech there is no good match between linguistically motivated sentence boundaries and turn boundaries or recognition hypotheses from automatic speech recognition.

5.3.2 Types of Disfluencies. The classification of disfluencies in this work follows Shriberg (1994), Meteer et al. (1995), and Rose (1998). It is worth noting, however, that any disfluency classification will be only an approximation of the assumed real phenomena and that often boundaries between different classes are fuzzy and hard to decide for human annotators (cf. Meteer et al. [1995] on annotators' problems with the classification of the word *so*).

- **Filled pauses**: We follow Rose's (1998) classification of nonlexicalized filled pauses (typically *uh*, *um*) and lexicalized filled pauses (e.g., *like, you know*). Whereas the former are usually nonambiguous and hence easy to detect, the latter are ambiguous and much harder to detect accurately.
- **Restarts or repairs**: These are fragments that are resumed, but without completely abandoning the first attempt. We follow the notation in Meteer et al. (1995) and Shriberg (1994), which has these parts: (1) reparandum, (2) interruption point (+), (3) interregnum (editing phase, {...}), and (4) repair.
  - **Repetition:** A restart with a verbatim repetition of a word or a sequence of words: *[she is + she is] happy*.
  - **Insertion:** A repetition of the reparandum, with some word(s) inserted: *[she liked + {um} she really liked] it*.
  - **Substitution**: The reparandum is not repeated: *[she + {uh} my wife] liked it*.
- **False starts:** These are abandoned, incomplete clauses. In some cases, they may occur at the end of an utterance, and they can be due to interruption by another speaker. Example: *so we didn't—they have not accepted our proposal.*

5.3.3 Related Work. The past decade has produced a substantial amount of research in the area of detecting intonational and linguistic boundaries in conversational speech, as well as in the area of detecting and correcting speech disfluencies. Whereas earlier work tended to look at these phenomena in isolation (Nakatani and Hirschberg 1994; Stolcke and Shriberg 1996), more recent work has attempted to solve several tasks within one framework (Heeman and Allen 1999; Stolcke et al. 1998).

Most approaches use some kind of prosodic information, such as duration of pauses, stress, and pitch contours, and most of them combine this prosodic information with information about word identity and sequence (*n*-grams, hidden Markov

models). In the study of Stolcke et al. (1998), the goal was to detect sentence boundaries and a variety of speech disfluencies on a large portion of the SWITCHBOARD corpus. An explicit comparison was made between prosodic and word-based models, and the results showed that an *n*-gram model, enhanced with segmental information about turn boundaries, significantly outperformed the prosodic model. Model combination improved the overall results, but only to a small extent. In more recent research, Shriberg et al. (2000) reported that for sentence boundary detection in two different corpora (BROADCAST NEWS and SWITCHBOARD), prosodic models outperform word-based language models and a model combination yields additional performance gains.

5.3.4 Overview. In the following, we will discuss the three components of the DIASUMM system that perform disfluency detection:

- a POS tagger that tags, in addition to the standard SWITCHBOARD Treebank-3 tag set (LDC 1999b), the following disfluent regions or words:

  1. coordinating conjunctions that don't serve their usual connective role, but act more as links between subsequent speech acts of a speaker (e.g., *and then*; we call these *empty coordinating conjunctions* in this work)
  2. lexicalized filled pauses (labeled as *discourse markers* in the Treebank-3 corpus; e.g., *you know, like*)
  3. editing terms within speech repairs (e.g., *I mean*)
  4. nonlexicalized filled pauses (e.g., *um, uh*)

- a decision tree (supported by a shallow chunk parser) that decides whether to label a particular sentence as a false start
- a repetition detection script (for repeated sequences of up to four words)

5.3.5 Training Corpus. For training, we used a part of the SWITCHBOARD transcripts that was manually annotated for sentence boundaries, POS, and the following types of disfluent regions (LDC 1999b):

- {A...}: asides (very rare; we ignore them in our experiments)
- {C...}: empty coordinating conjunctions (e.g., *and then*)
- {D...}: discourse markers (i.e., *lexicalized filled pauses* in our terminology, e.g., *you know*)
- {E...}: editing terms (within repairs; e.g., *I mean*)
- {F...}: filled pauses (nonlexicalized; e.g., *uh*)
- [... + ...]: repairs: the part before the + is called reparandum (to be removed), the part after the + repair (proper)

Sentence boundaries can be at the end of completed sentences (E_S) or of noncompleted sentences, such as false starts or abandoned clauses (N_S).

Table 5
Precision, recall and F_1-scores of the four disfluency tag categories for the SWITCHBOARD test set.

Description	Count	Tag	Precision	Recall	F_1
Empty coordinating conjunctions	5,990	CO	0.84	0.93	0.88
Lexicalized filled pauses	5,787	DM	0.95	0.90	0.93
Editing terms	1,004	ET	0.98	0.94	0.96
Nonlexicalized filled pauses	12,926	UH	0.98	0.98	0.98

Table 6
POS tagging accuracy on five subcorpora (evaluated on 500-word samples).

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
Known words	92.8	90.6	92.7	90.6	93.2
Unknown words (total)	48.0 (25)	44.4 (9)	69.6 (23)	86.4 (22)	92.6 (27)
Overall	90.6	89.8	91.6	90.4	93.2

5.3.6 POS Tagger. We are using Brill's rule-based POS tagger (Brill 1994). Its basic algorithm at run time (after training) can be described as follows:

1. Tag every word with its most likely tag, predicting tags of unknown words based on rules.
2. Change every tag according to its right and left context (both words and tags are considered), following a list of rules.

For preprocessing, we replaced the tags in the regions of {C...}, {D...}, and {E...} with the tags CO (coordinating), DM (discourse marker), and ET (editing term), respectively. (The filler regions {F...} are already tagged with UH in the corpus.) Lines that contain typographical errors were excluded from the training corpus. We further eliminated all incomplete words (XX tag) and combined multiwords, marked by a GW tag, into a single word (hence eliminating the GW tag).[11] The entire resulting new tag set had 42 tags.[12]

Training of the POS tagger proceeded in three stages, using about 250,000 tagged words for each stage. The trained POS tagger's performance on an unseen test set of about 185,000 words is 94.1% tag accuracy (untrained baseline: 84.8% accuracy).

Table 5 shows precision, recall, and F_1-scores for the four categories of disfluency tags, measured on the test set after the last training phase. We see that the nonlexicalized filler words are almost perfectly tagged ($F_1 = 0.98$), whereas the hardest task for the tagger is the empty coordinating conjunctions ($F_1 = 0.88$): There are a few highly ambiguous words in that set, such as *and*, *so*, and *or*.

Table 6 shows the POS tagging accuracy on the five subcorpora of our dialogue corpus, evaluated on a sample of 500 words per subcorpus. We see that the POS-tagging accuracy is slightly lower than for the SWITCHBOARD set that was used for

11 The sole function of the GW tag is to label words that are considered to be parts of other words but were transcribed separately, such as: drug/GW testing/NN.
12 For a description of the POS tags used in that database see Santorini (1990) and LDC (1999a).

Table 7
Disfluency tag detection (F_1) for five subcorpora (results in parentheses: less than 10 tags to be detected).

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
CO	.89	.89	.38	.77	.54
DM	.93	.73	.90	.82	.30
ET	.95	.95	(.94)	.85	.88
UH	.56	.62	(.14)	(.28)	.45

training (approximately 90–93%; global average: 91.1%). Further we observe that with the exception of the CALLHOME corpora, the majority of unknown words were actually tagged correctly. The most frequent errors were (1) conjunctions tagged as empty coordinated conjunctions, (2) proper names tagged as regular nouns, and (3) adverbs tagged as adjectives.

Finally, we look at the POS tagger's performance for the four disfluency tags CO, DM, ET, and UH in our five subcorpora; the results of this evaluation are presented in Table 7. We can see that the detection accuracy is generally lower than for the corpus on which we trained the tagger (SWITCHBOARD), but still quite good in general. The major exceptions are the UH tags, on which the F_1-scores are comparatively low for all subcorpora. The reason for this can be found mostly in words like *yes*, *no*, *uh-huh*, *right*, *okay*, and *yeah*, which are often tagged with UH in SWITCHBOARD but frequently are not considered to be irrelevant words in our corpus and hence not marked as disfluent (e.g., if they are considered to be the answer to a question or a summary-relevant acknowledgment). We circumvent potential exclusion from the summary output of these and other words that might be erroneously tagged as nonlexicalized filled pauses (UH) by marking a small set of words as exempt from removal (see section 5.5.6).

5.3.7 False Start Detection. False starts are quite frequent in spontaneous speech, occurring at a rate of about 10–15% of all sentences (SWITCHBOARD, CALLHOME). They involve less than 10% of the total words of a dialogue; about 34% of the words in these incomplete sentences are part of some other disfluencies, such as filled pauses or repairs. (In complete sentences, only about 15% of the words are part of these disfluencies.) For CALLHOME, the average length of complete sentences is about 6 words, of incomplete sentences about 4.1 words (including disfluencies).

We trained a C4.5 decision tree (Quinlan 1992) on 8,000 sentences of SWITCHBOARD. As features we use the first and last four trigger words (words that have a high incidence around sentence boundaries) and POS of every sentence, as well as the first and last four chunks from a POS-based chunk parser. This chunk parser is based on a simple context-free POS grammar for English. It outputs a phrasal bracketing of the input string (e.g., noun phrases or prepositional phrases). Further, we encode the length of the sentence in words and the number of the words not parsed by the chunk parser. We observed that whereas the chunk information itself does not improve performance over the baseline of using trigger words and POS information only, the derived feature of "number of not parsed words" actually does improve the results.

We ran the decision tree on data with perfect POS tags (for SWITCHBOARD only), disfluency tags (except for repairs), and sentence boundaries. The evaluations were performed on independent test sets of about 3,000 sentences for SWITCHBOARD and of our complete dialogue corpus. Table 8 shows the results of these experiments. Typical errors, where complete sentences were classified as incomplete, are inverted forms or

Table 8
False start classification results for different corpora (F_1).

	SWBD	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
False start frequency (in %)	12.3	12.1	11.0	2.0	7.2	13.9
False start detection (F_1)	.611	.545	.640	.286	.352	.557

Table 9
Detection accuracy for repairs on the basis of individual word tokens using the repetition filter.

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
Repair tokens (%)	4.7	3.8	2.2	1.3	7.9
Precision	.88	.78	.25	.35	.91
Recall	.41	.32	.01	.04	.27
F_1-score	.56	.45	.02	.08	.41

ellipsis at the end of a sentence (e.g., *neither do I*, *it seems to*). The performance for the informal corpora (CALLHOME, GROUP MEETINGS) is better than that for the formal corpora (NEWSHOUR, CROSSFIRE); this is related to the fact that the relative frequency of false starts is markedly lower in these latter data sets and that these corpora are more dissimilar to the training corpus (SWITCHBOARD).

5.3.8 Repetition Detection. The repetition detection component is concerned with (verbatim) repetitions within a speaker's turn, the most frequently occurring case of all speech repairs for informal dialogues (insertions and substitutions are comparatively less frequent). Repeated phrases can potentially be interrupted by other disfluencies, such as filled pauses or editing terms. Repetition detection is performed with a script that can identify repetitions of word/POS sequences of length one to four (longer repetitions are extremely rare: on average, less than 1% of all repetitions). Words that have been marked as disfluent by the POS tagger are ignored when the repeated sequences are considered, so we can correctly detect repetitions such as *[he said uh to + he said to] him*. . . .

We are evaluating the precision, recall, and F_1-scores for this component at the level of individual words when the POS tagger and the sentence boundary detection component are used. Table 9 shows the results. We see that for the informal subcorpora, we get very good precision (only a few repetitions detected are incorrect), and recall is in the 25–45% range (since we cannot detect substitution or insertion type of repairs). The results for the formal subcorpora are considerably worse, so this filter should probably not be used for corpora with as few repetitions as NEWSHOUR or CROSSFIRE. We checked all of the 95 *false positives* of this evaluation and observed that in the majority of cases (41%), the repetition was correctly detected but was not marked by the human annotator, since it might be considered a case of emphasis. We believe that although some nuances of the sentence(s) might be lost, for the purpose of summarization it makes perfect sense to reduce this information. Sometimes, individual words are repeated for emphasis, sometimes whole sentences (e.g., "Good./ Good./"). In the following example from English CALLHOME, the emphasis is rather extreme:

```
203 B: [...] How is the new person doing? q/
204 A: Very very very very very well. / [...]
```

Further, about 19% of false positives were correct but not annotated because they span multiple turns, and about 14% were erroneously missed by the human annotator. Only the remaining cases (26%) were actual false positives, caused by incorrect POS tags (5%, typically an incorrectly tagged "that/WDT that/DT" sequence at the beginning of a relative clause) or incorrect sentence boundaries (21%).

There have been attempts to get a more complete coverage of detection and correction of all types of speech repairs (Heeman and Allen 1999). We decided, however, to use a simple method here that works well for a large subset of cases and is very efficient at the same time.

5.3.9 Disfluency Correction in DIASUMM. After detection, the correction of disfluencies is straightforward. When DIASUMM generates its output from the ranked list of sentences, it skips the false starts, the repetitions, and the words that were tagged with CO, DM, ET, or UH by the POS tagger.

5.4 Sentence Boundary Detection

5.4.1 Introduction. The purpose of the sentence boundary detection component is to insert linguistically meaningful sentence boundaries in the text, given a POS-tagged input. We consider all intraturn and interturn boundary positions for every speaker in a conversation. We use the abbreviations EOS for *end of complete sentence* (E_S in the SWITCHBOARD corpus) and NEOS for *end of noncomplete sentence* (N_S in the SWITCHBOARD corpus). The frequency of sentence boundaries (with respect to the total number of words) is about 13.3%, most of the boundaries (almost 90%) being end markers of completed sentences (SWITCHBOARD).

5.4.2 Training and Testing. We trained a C4.5 decision tree and computed its input features from a context of four words before and after a potential sentence boundary, motivated by the results of Gavaldà, Zechner, and Aist (1997). Also following Gavaldà, Zechner, and Aist (1997), we used 60 trigger words with high predictive potential, employing the score computation method described in this article.

The decision tree input features for every word position are as follows:

- POS tag (42 different tags)
- trigger word (60 different trigger words)
- turn boundary before this word?
- if turn boundary: length of pause after last turn of same speaker

Since NEOS boundaries occur very infrequently (only about 10% of all boundaries, which is only about 1% of all potential boundaries), we decided to merge this class with the EOS class and report results for this combined class only (CEOS). (We relied on the false-start detection module described above to identify the NEOS sentences within this merged class of sentences *after* the sentence boundary classification.)

For training, we used 25,000 words from the Treebank-3 corpus; the test set size was 1,000 words. Table 10 shows the results in detail for the various parameter combinations. We see that for good performance we need to know about one of these two features: "is there a turn boundary before this word?" or "pause duration after last turn from same speaker."

Table 10
Sentence boundary detection accuracy (F_1-score).

With Interturn Pause Duration?	Yes		No	
With Turn Boundary Info?	Yes	No	Yes	No
Training set	.904	.903	.900	.884
Test set	.887	.884	.884	.825

Table 11
Inter- and intraturn boundary detection (BD) results on 1,000-word test set.

	Occurrence (%)	Detection Accuracy (F_1)
Interturn non-BD	12 (1.2)	.56
Interturn BD	112 (11.3)	.95
Intraturn non-BD	809 (81.4)	.99
Intraturn BD	61 (6.1)	.77

5.4.3 Effect of Imperfect POS Tagging. To see how much influence an imperfect POS tagging might have on these results, we POS-tagged the test set data using the POS tagger described above. For this and the following experiments, we increased the training corpus for the decision tree to 40,000 words. The POS tagger accuracy for this test set was about 95.3%, and the F_1-score for CEOS was .882, which is 98.9% of .892 on perfect POS-tagged input. This is encouraging, since it shows that the decision tree is not very sensitive to the majority of POS errors.

5.4.4 Interturn and Intraturn Boundaries. In this analysis, we are interested in comparing the detection of sentence boundaries *between turns* (interturn) to the detection of boundaries *within a turn* (intraturn). Table 11 shows the results of this analysis (same test set as above). As might be expected, the performance is very good for the two frequent classes: sentence boundaries at the end of turns and nonboundaries within turns ($F_1 > .95$), but considerably worse for the two more infrequent cases. The very rare cases (around 1% only) of non–sentence boundaries at the end of turns (i.e. turn–continuations) show the lowest performance ($F_1 = .56$).

5.4.5 Sentence Boundary Detection on Dialogue Corpus. To get a picture of the realistic performance of the sentence boundary detection component, using the (imperfect) POS tagger and a faster, but slightly less accurate, decision tree,[13] we evaluate the sentence boundary detection accuracy for all five subcorpora of our dialogue corpus. Table 12 provides the results of these experiments. The results reflect a trend very similar to that for the SWITCHBOARD corpus, in that the two more frequent classes (interturn boundaries and intraturn nonboundaries) have high detection scores, whereas the two more infrequent classes are less well detected. Furthermore, we observe that in cases in which the relative frequency of rare classes is further reduced, the classification accuracy declines overproportionally (particularly for the rarest class of the interturn nonboundaries). Also, overall boundary detection is better for the two more informal corpora, CALLHOME and GROUP MEETINGS ($F_1 > .72$).

13 This decision tree uses a different type of encoding, but the same input features.

Table 12
Boundary detection (BD) accuracy (F_1) for five subcorpora (in parentheses: relative frequency of class in percent).

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
Interturn non-BD	.51 (2.9)	.31 (1.4)	[0] (0.0)	.10 (0.1)	.06 (0.1)
Interturn BD	.84 (9.9)	.89 (12.3)	.93 (2.0)	.89 (2.9)	.93 (5.4)
Intraturn non-BD	.97 (80.7)	.97 (79.5)	.97 (91.8)	.97 (91.2)	.97 (87.6)
Intraturn BD	.60 (6.5)	.65 (6.8)	.56 (6.2)	.42 (5.8)	.56 (6.9)
Overall BD	.75 (16.4)	.80 (19.1)	.66 (8.2)	.58 (8.7)	.72 (12.4)
Overall non-BD	.95 (83.6)	.96 (80.9)	.97 (91.8)	.97 (91.3)	.96 (87.6)

5.5 Cross-Speaker Information Linking

5.5.1 Introduction. One of the properties of multiparty dialogues is that shared information is created between dialogue participants. The most obvious interactions of this kind are question-answer (Q-A) pairs. The purpose of this component is to create automatically such coherent pieces of relevant information, which can then be extracted together while generating the summary. The effects of such linkings on actual summaries can be seen in two dimensions: (1) increased local coherence in the summary and (2) a potentially higher informativeness of the summary. Since Q-A linking has a side effect in that *other* information will be lost with respect to a summary of the same length without Q-A linking, the second claim is much less certain to hold than the first. We investigated these questions in related work (Zechner and Lavie 2001) and found that although Q-A linking does not significantly change the informativeness of summaries on average, it does increase summary coherence (fluency) significantly. In this section, we will be concerned with the following two intuitive subtasks of Q-A linking: (1) identifying questions (Qs) and (2) finding their corresponding answers.

5.5.2 Related Work. Detecting a question and its corresponding answer can be seen as a subtask of the speech act detection and classification task. Recently, Stolcke et al. (2000) presented a comprehensive approach to dialogue act modeling with statistical techniques. A good overview and comparison of recent related work can also be found in Stolcke et al.'s article. Results from their evaluations on SWITCHBOARD data show that word-based speech act classifiers usually perform better than prosody-based classifiers, but that a model combination of the two approaches can yield an improvement in classification accuracy.

5.5.3 Corpus Statistics. For training of the question detection module, we used the manually annotated set of about 200,000 SWITCHBOARD speech acts[14] (SAs);[15] for training of the answer detection component, we used the eight English CALLHOME dialogues (8E-CH), which were manually annotated for Q-A pairs. Although we were aiming to detect all questions in the question detection module, the answer detection module focuses on Q-A pairs only: We exclude all questions from consideration that are not Yes-No- (YN) or Wh-questions (such as rhetorical or back-channel questions),

14 In this work, the notions of *speech acts* and *sentences* can be considered equivalent.
15 From the Johns Hopkins University Large Vocabulary Continuous Speech Recognition (LVCSR) Summer Workshop 1997. Thanks to Klaus Ries for providing the data, which are also available from http://www.colorado.edu/ling/jurafsky/ws97/.

Table 13
Frequency of different types of questions in the 8E-CH data set.

Sentences	2,211
Wh-questions total	20
. . . With immediate answers	15 (75%)
YN-questions total	48
. . . With immediate answers	38 (79%)
Qs excluded for Q-A detection	15
Questions total	83 (3.75%)

as well as those that do not have an answer in the dialogue. Thus we employ only 68 pf the 83 questions marked in the 8E-CH data set for these evaluations. Table 13 provides the statistics concerning questions and answers for the 8E-CH subcorpus and shows that for a small but significant number of questions, the answer does not immediately follow the question speech act (*delayed answers*).

5.5.4 Automatic Question Detection. We used two different methods, both trained on SWITCHBOARD data: (1) a speech act tagger[16] and (2) a decision tree based on trigger word and part-of-speech information.

Speech act tagger. The speech act tagger tags one speech act at a time and hence can make use only of speech act unigram information. Within a speech act, it uses a language model based on POS and the 500 most frequent word/POS pairs. It was trained on the aforementioned SWITCHBOARD speech act training set. It was not optimized for the task of question detection. Its typical run time for speech act classification is about 10 speech acts per second.

Decision tree question classifier. The decision tree classifier (C4.5) uses the following set of features: (1) POS and trigger word information for the first and last five tokens of each speech act;[17] (2) speech act length, and (3) occurrence of POS bigrams. The set of trigger words is the same as for the sentence boundary detection module. The POS bigrams were designed to be most discriminative between question speech acts (q-SAs) and non–question speech acts (non-q-SAs). The bigrams were obtained as follows:

1. For a balanced set of q-SAs and non-q-SAs (about 9,000 SAs each): Count all the POS bigrams in positions $1 \ldots 5$ and $(n-4) \ldots n$ (using START and END for the first and last bigrams, respectively) and memorize position (beginning or end of SA) and type (q-SA vs. non-q-SA).
2. For all bigrams:
   (a) Add one to the count (to prevent division by zero).
   (b) Divide the q-SA count by the non-q-SA count.
   (c) If the ratio is smaller than one, invert it (ratio := 1/ratio).
   (d) Multiply the result of (c) by the sum of q-SA count and non-q-SA count.[18]
3. Extract the 100 bigrams with the highest value.

16 Thanks to Klaus Ries for providing us with the software.
17 Shorter speech acts are padded with dummies.
18 Leaving out this step favors low-frequency, high-discriminative bigrams too much and causes a slight reduction in overall Q-detection performance.

Table 14
Question detection on the 8E-CH corpus using two different classifiers.

	SA Tagger	Decision Tree
Overall error	3.2%	4.7%
Precision	.57	.63
Recall	.61	.51
F_1	.59	.56
Typical classification time (SAs/sec)	10	1,000

Experiments and results. The question detection decision tree was trained on a set of about 20,000 speech acts from the SWITCHBOARD corpus. We first evaluated the speech act tagger and the decision tree classifier on the 8E-CH data set. Whereas in the later stage of answer detection, questions without answers and nonpropositional questions are ignored, at this point we are interested in the detection of all annotated questions in the corpus. This also reflects the fact that the training set contains all possible types of questions.

Table 14 reports the results of the question detection experiments with the two classifiers used on the 8E-CH subcorpus. We note that whereas the decision tree is performing only slightly worse than the speech act tagger, its typical classification time is two orders of magnitude faster. Based on these observations, we decided to use the question detection decision tree in the Q-A linking component of the DIASUMM system.

5.5.5 Detecting the Answers. After identifying which sentences are questions, the next step is to identify the answers to them. From the 8E-CH statistics of Table 13 we observe that for more than 75% of the YN- and Wh-questions, the answer is to be found in the first sentence of the speaker talking after the speaker uttering the question. In the remainder of cases, the majority of answers are in the second (instead of the first) sentence of the responding speaker. Further, the speaker who has posed a question usually utters no (or only very few) sentences *after* the question is asked and before the next speaker starts talking.

In addition to detecting sequential Q-A pairs, we also want to be able to detect simple embedded questions, as shown in this example of a brief clarification dialogue:

```
Q  1  A: When are we meeting then?
Q  2  B: You mean tomorrow?
   3  A: Yes.
   4  B: At 4pm.
```

We devise the following heuristics to detect answers to question speech acts which have been previously identified:

- If the first speaker change after the question occurs more than *maxChg* sentences after the question, the search is stopped and no Q-A pair is returned.
- Answer hypotheses are sought for maximally *maxSeek* sentences after the first speaker change following the question, but not over interruptions by any other speaker; that is, we check within a single speaker region

(this is the stopping criterion for the following two heuristics). An exception occurs if there is an embedded question in the first single speaker region: In that case, we look at the next region where a speaker different from the initial Q-speaker is active.[19]

- Answers have to be minimally *minAns* words long; if they are shorter, we add the next sentence to the current answer hypothesis.
- Even if the minimum answer length is reached, the answer can be optionally *extended* if at least one word in the answer matches a word from the question (one of two different stop lists (*StopShort*, *StopLong*) or no stop list is used to remove function words from consideration).[20]

We have these further restrictions for the case of embedded questions:

1. If we detect a potential embedded Q-A pair, the answer to the surrounding question must immediately follow the answer to the embedded question (i.e., the region following the potential answer region of the embedded question—sentence 4 in our above example—must (1) not contain a question itself *and* (2) be from a different speaker than the surrounding question).
2. A *crossover* is prohibited; that is, we eliminate all pairs $\langle Q_j, A_l \rangle$ when a pair $\langle Q_i, A_k \rangle$ was already detected, with $i < j < k < l$ (k, l being start indices of answer spans).

The output of the algorithm is a list of triples $\langle Q, A_{start}, A_{end} \rangle$, where Q is the sentence ID of the question and A_{start} the first and A_{end} the last sentence of the answer. As mentioned above, we use only 68 of the 83 questions marked in the 8E-CH data set for these evaluations, since only these are YN- or Wh-questions that actually *have* answers in the dialogue. There are four possible outcomes for each triple: (1) irrelevant: a Q-A pair with an incorrectly hypothesized question (this is the fault of the question detection module, not of this heuristic); (2) missed: the answer was missed entirely; (3) completely correct: A_{end} coincides with the correct answer sentence ID; and (4) correct range: the answer is contained in the interval $[A_{start}, A_{end}]$ but does not coincide with A_{end}. For the calculation of precision, recall, and F_1-score, we count classes (3) and (4) as correct and use the sum of all classes for the denominator of precision and the total number of Q-A pairs (68 in this development set) as the denominator of recall.

To determine the best parameters, we varied them across a reasonable set of values and ran the answer detection script for all combinations of parameters. The best results (with respect to F_1-score) using questions detected by the speech act tagger and the decision tree are reported in Table 15. In the DIASUMM system, we use the following optimal parameter settings for the answer detection heuristics: $maxChg = 2$, $maxSeek = 4$, $minAns = 10$, $sim = on$, $stop = no$.

Finally, we evaluated the performance of both the Q-detection module and the combined Q-A detection on all five subcorpora, using the decision tree for question detection; the results are reported in Table 16. Except for the rather small NEWSHOUR

19 This would be sentence 4 in the example above.
20 *StopLong* contains 571 words, *StopShort* only 89 words, most of which are auxiliary verbs and filler words.

Table 15
Q-A detection results using two different classifiers for question detection (68 Q-A pairs to be detected).

	SA Tagger	Decision Tree
All hypothesized Q-A pairs	80	54
Correct [(3) and (4)]	42	31
maxChg (1–5)	4	2
maxSeek (2–4)	3–4	2–4
minAns (1–10)	5–10	2–10
Similarity extension (on/off)	on	on
Stop list (no/short/long)	no/short	no/short
Precision	.53	.57
Recall	.62	.46
F_1-score	.57	.51

Table 16
Performance comparison for Q- and Q-A detection (Q-detection with the decision tree question classifier).

	8E-CH	4E-CH	NHOUR	XFIRE	G-MTG
Q to detect	83	94	19	110	49
Q-hypotheses	67	60	16	71	52
Q-detection (F_1)	.56	.58	.80	.60	.59
Q-A pairs to detect	68	69	18	79	32
Q-A pair hypotheses	54	54	14	54	33
Q-A detection (F_1)	.51	.60	.81	.51	.51

corpus (with fewer than 20 questions or Q-A pairs to identify), the typical Q-detection F_1-score is around .6 and the Q-A detection F_1-score around .5. In two cases, the Q-A detection performance is slightly better than the Q-detection performance. This can be explained by the fact that the answer detection algorithm prunes away a number of Q-hypotheses, reducing the space for potential Q-A hypotheses.

5.5.6 Q-A Detection within DIASUMM. When we use the Q-A detection module as part of the DIASUMM system, we want to ensure that (1) there are no Q-A pairs containing Q-sentences that are false starts and that (2) the initial part of an answer is not lost in case the disfluency detection component marks some indicative words as disfluencies. To satisfy the first constraint, we block Q-detection of sentences that have been previously classified as false starts; as for the second constraint, we create a list of indicative words (relevant for YN-questions) that are not to be removed by the summary generator if they appear in the beginning (leading five words) of answers.[21]

5.6 Sentence Ranking and Selection

5.6.1 Introduction. The sentence ranking and selection component is an implementation of the MMR algorithm (Carbonell, Geng, and Goldstein 1997), applied to extracting the most relevant sentences from a topical segment of a dialogue. The component's output in isolation serves as the MMR baseline for the global system evaluation. Its

21 The current list comprises the following words: *no*, *yes*, *yeah*, *yep*, *sure*, *uh-huh*, *mhm*, *nope*.

purpose is to determine weights for terms and sentences, to rank the sentences according to their relevance within each topical segment of the dialogue, and finally to select the sentences for the summary output according to their rank, as well as to other criteria, such as question-answer linkages, established by previous components. The selected sentences are presented to the user in text order.

5.6.2 Tokenization. In addition to the tokenization rules for the global system (section 5.2), we apply a simple six-character truncation for stemming and use a stop word list to eliminate frequent noncontent words. In the experiments, we used the following five different stop word lists:

- the original SMART list (Salton 1971) (SMART-O)
- a manually edited stop list based on SMART (SMART-M)
- a stop list with all closed-class words from the POS tagger's lexicon (POS-O)
- a manually edited stop list based on the POS tagger's lexicon and frequent closed-class words in the CALLHOME training corpus (POS-M)
- an empty stop list (EMPTY)

5.6.3 Term and Sentence Weighting. The basic idea for determining the most relevant sentences within a topical segment is as follows: First, we compute a vector of word weights for the segment $\vec{tf}_q$ (including all stemmed non–stop words) and do the same for each sentence ($\vec{tf}_t$), then we compute the similarity between sentence and segment vectors for each sentence. That way, sentences that have many words in common with the segment vector are rewarded and receive a higher relevance weight.

Whereas we compose the sentence vectors $\vec{tf}_t$ using direct term frequency counts, the weights for segment terms are determined according to one of the three formulae in equation (1) (*freq*, *smax*, and *log*), inspired by Cornell University's SMART system (Salton 1971):

$$tf_{i,s} = f_{i,s} \quad \text{or} \quad 0.5 + 0.5\frac{f_{i,s}}{f_{smax}} \quad \text{or} \quad 1 + \log f_{i,s}, \tag{1}$$

where $f_{i,s}$ are the in-segment frequencies of a stem and f_{smax} are maximal segment frequencies of any stem in the segment. Finally, we multiply an inverse document frequency (IDF) weight to $\vec{tf}_s$ to obtain the segment vectors $\vec{tf}_q$, as shown in equations (2) and (3):

$$tf_{i,q} = tf_{i,s}IDF_{i,s} \tag{2}$$

$$IDF_{i,s} = 1 + \log\frac{N_{seg}}{i_{seg}} \quad \text{or} \quad \frac{N_{seg}}{i_{seg}}. \tag{3}$$

IDF values are computed with respect to a collection of topical segments, either the current dialogue (DIALOGUE) or a set of dialogues (CORPUS). N_{seg} is the total number of topical segments in the IDF corpus, and i_{seg} is the number of segments in which the token *i* appears at least once. The effect of using IDF values is to boost those words that are (relatively) unique to any given segment over those that are more evenly distributed across the corpus.

As stated above, the main algorithm is a version of the MMR algorithm (Carbonell, Geng, and Goldstein 1997; Carbonell and Goldstein 1998), which emphasizes sentences

that contain many highly weighted terms for the current segment (salience) and are sufficiently dissimilar to previously ranked sentences (diversity or antiredundancy). The MMR formula is given in equation (4):

$$nextsentence = \arg\max_{t_{nr,j}}(\lambda sim_1(query, t_{nr,j}) - (1-\lambda)\max_{t_{r,k}} sim_2(t_{nr,j}, t_{r,k})). \quad (4)$$

The MMR formula describes an iterative algorithm and states that the next sentence to be put in the ranked list will be taken from the sentences that have not yet been ranked (t_{nr}). This sentence is (1) maximally similar to a query and (2) maximally dissimilar to the sentences that have already been ranked (t_r). We use the topical segment word vector $\vec{tf}_q$ as query vector. The λ parameter ($0.0 \leq \lambda \leq 1.0$) is used to trade off the influence of salience against that of redundancy.

Both similarity metrics (sim_1, sim_2) are inner vector products of stemmed-term frequencies (equations (5) and (6)). sim_1 can be normalized in different ways: (1) to yield a cosine vector product (division by product of vector lengths), (2) division by number of content words,[22] and (3) no normalization:

$$sim_1 = \frac{\vec{tf}_q \vec{tf}_t}{|\vec{tf}_q||\vec{tf}_t|} \quad \text{or} \quad \frac{\vec{tf}_q \vec{tf}_t}{1 + \sum_i tf_{i,t}} \quad \text{or} \quad \vec{tf}_q \vec{tf}_t \quad (5)$$

$$sim_2 = \frac{\vec{tf}_{t1} \vec{tf}_{t2}}{|\vec{tf}_{t1}||\vec{tf}_{t2}|} \quad (6)$$

Emphasis factors. Every sentence's similarity weight (sim_1) can be (de)emphasized, based on a number of its properties. We implemented optional emphasis factors for:

- *Lead emphasis*: for the leading n% of a segment's sentences: $sim'_1 = sim_1 l$, with l being the lead factor.
- *Q-A emphasis*: for all sentences that belong to a question-answer pair: $sim'_1 = sim_1 q$, with q being the Q-A emphasis factor.
- *False-start deemphasis*: for all sentences that are false starts: $sim'_1 = sim_1 f$, with f being the false-start factor.
- *Speaker emphasis*: for each individual speaker s, an emphasis factor s_e can be defined: $sim'_1 = sim_1 s_e$ for all sentences of speaker s.[23]

These parameters can serve to fine-tune the system for particular applications or user preferences. For example, if the false starts are deemphasized, they are less likely to trigger a question's being linked to them in the linking process. If questions and answers are emphasized, more of them will show up in the summary, increasing its coherence and readability. In a situation in which a particular speaker's statements are of higher interest than those of other speakers, his sentences can be emphasized, as well.

Since sim_2 is a cosine vector product and hence in [0,1], we have to normalize sim_1 to [0,1] as well to enable a proper application of the MMR formula. For this normalization of sim_1, we divide each sim_1 score by the maximum of all sim_1 scores in a segment after initial computation and application of the various emphasis factors described here.

22 To avoid division by zero, we add one to every sentence length.
23 Speaker emphasis is not used in our evaluations.

5.6.4 Q-A Linking. While generating the output summary from the MMR-ranked list of sentences, whenever a question or an answer is encountered (detected previously by the Q-A detection module), the corresponding answer/question is linked to it and moved up the relevance ranking list to immediately follow the current question/answer. If the question-answer pair consists of more than two sentences, the linkages are repeated until no further questions or answers can be added to the current linkage cluster.

5.6.5 Summary Types. DIASUMM can generate several different types of summaries, the two main versions being (1) the CLEAN summary, which is based on the output of all DIASUMM components (disfluency detection, sentence boundary detection, Q-A linking), and (2) the TRANS summary, in which all dialogue specific components are ignored (essentially, this is an MMR summary of the original dialogue transcript). For the purpose of the global system evaluation, we use only these two versions of summaries, as well as LEAD baseline summaries, where the summary is formed by extracting the first n words from a topical segment.[24]

Furthermore, the system can generate phrasal summaries, which render the sentences in the same ranking order as the CLEAN summary but reduce the output to noun phrases and potentially other phrases, depending on the setting of parameters.[25]

In Figure 2 we show an example of a set of LEAD, TRANS, CLEAN, and PHRASAL summaries. The set was generated from the CALLHOME transcript presented in section 2.

5.6.6 System Tuning. This section describes how we arrive at an optimal parameter setting for each subcorpus (CALLHOME, NEWSHOUR, CROSSFIRE, GROUP MEETINGS). We want to establish an MMR baseline for the global system evaluations with which we can then compare the results of the entire DIASUMM system. Note that for all the tuning experiments described in this subsection, we did not make use of any other DIASUMM components, namely, disfluency detection, sentence boundary detection, and question-answer linking. All experiments were based on the human gold standard with respect to topical segments. We used only the `devtest` set for the four subcorpora here (8E-CH = CALLHOME, DT-NH = NEWSHOUR, DT-XF = CROSSFIRE, and DT-MTG = GROUP MEETINGS).

Since the length of turns varies widely, one could argue that an easy way to increase performance for the MMR baseline (which does *not* use automatic sentence boundary detection) might be to split overly long turns evenly into shorter chunks. This was done by Valenza et al. (1999), who experimented with lengths of 10–30 words per extract fragment. We add this option as an additional parameter to the MMR baseline. If the parameter is set to n words, turns with a length $l \geq 1.5n$ get cut into pieces of lengths n iteratively until the last remaining piece is $l < 1.5n$.

Evaluation metric. To evaluate the performance of this component, we use the word-based evaluation metric described in section 6.2, which gives the highest scores to summaries containing words with the highest average relevance scores, as marked by human annotators. We then average these scores over all topical segment summaries of a particular subcorpus.

24 Note that LEAD summaries are to be distinguished from summaries in which *lead emphasis* is used, as described above. In the latter case, the segment-initial sentence weights are increased, whereas in the former case, we strictly extract the leading n words from a given segment.

25 To determine these constituents, we use the output of the chunk parser employed by the false start detection component.

```
LEAD:
  1 a: Oh
  2 b: They didn't know he was going to get shot but
       it was at a peace rally so I mean it just worked out
  3 b: I mean it was a good place for the poor guy to die I mean
       because it was you know right after the rally and
       everything was on film and everything [...]

TRANS:
  2 b: They didn't know he was going to get shot but
       it was at a peace rally so I mean it just worked out
  3 b: I mean it was a good place for the poor guy to die
       I mean because it was you know right after the
       rally and everything was on film and everything
 11 b: Him [...]

CLEAN:
  7 b: We just finished the thirty days mourning for him now
       it's everybody's still in shock it's terrible what's
       going on over here
 31 b: What's the reaction in america really do people care [...]
 34 a: Most I don't know what I mean like the jewish community
       a lot all of us were very upset

PHRASAL:
  4 b: it just worked ... it was a good
       place for the poor guy to die ... it was [...]
  7 b: we just finished the thirty days mourning
       for him ... it's ... everybody's ...
       in shock it's ... going ...
 31 b: 's the reaction in america ... do people care ...
 34 a: i don't know ... mean like the jewish
       community a lot ... of us were
```

Note: The turn IDs are just indicating the relative position of the turns within the original text and do not always correspond to the turn numbers of the original or to the turn numbers of the other summaries. The . . . marks the position in those sentences where the length threshold for a summary was reached.

Figure 2
Example summaries of 20% length: LEAD, TRANS, CLEAN and PHRASAL.

Parameter tuning. The system tuning proceeded in three phases, in which we held the summary size constant to 15% and optimized the following set of parameters:

1. Term weight type: freq, smax, log
2. Normalization: cos, length, none
3. IDF type: corpus, dialogue, none
4. IDF method: log, mult
5. Extract span: 10–30 or original turn (orig)

Table 17
Optimally tuned parameters for MMR baseline system (tuning on `devtest` set subcorpora).

	8E-CH	**DT-NH**	**DT-XF**	**DT-Mtg**
Term weight type	smax	smax	smax	smax
Normalization	cos	none	cos	none
IDF type	corpus	corpus	corpus	corpus
IDF method	log	log	mult	log
Extract span	20	orig	25	orig
MMR-λ	0.85	0.8	1.0	0.8
Stop list	SMART-M	POS-M	POS-M	POS-M
Lead factor	1.0	1.0	1.0	2.0

6. MMR-λ: 0.8–1.0
7. Stop lists: SMART-O, SMART-M, POS-O, POS-M, EMPTY
8. Lead factor: 1.0–5.0 (applied to first 20% of sentences)

Table 17 shows the parameter settings that were determined to be optimal for the MMR baseline system (TRANS summaries).

5.7 System Performance

The majority of the system components are implemented in `Perl5`, except for the C4.5 decision tree (Quinlan 1992), the chunk parser (Ward 1991), and the POS tagger (Brill 1994), which were implemented in `C` by the respective authors. We measured the system runtime on a 300 MHz Sun Ultra60 dual-processor workstation with 1 GB main memory, summarizing all 23 dialogue excerpts from our corpus. The average runtime for the whole system, including all of its components except for the topic segmentation module, was 17.8 seconds, and for the sentence selection component alone 7.0 seconds (per-dialogue average). The average ratio of system runtime to dialogue duration was 0.029 (2.9% of real speaking time).

6. Evaluation

6.1 Introduction

Traditionally, summarization systems have been evaluated in two major ways: (1) intrinsically, measuring the amount of the core information preserved from the original text (Kupiec, Pedersen, and Chen 1995; Teufel and Moens 1997), and (2) extrinsically, measuring how much the summary can benefit in accomplishing another task (e.g., finding a document relevant to a query or classifying a document into a topical category) (Mani et al. 1998).

In this work, we focus on intrinsic evaluation exclusively. That is, we want to assess how well the summaries preserve the essential information contained in the original texts. As other studies have shown (Rath, Resnick, and Savage 1961; Marcu 1999), the level of agreement between human annotators about which passages to choose to form a good summary is usually quite low. Our own findings, reported in section 4.2.4, support this in that the intercoder agreement, here measured on a word level, is rather low.

We decided to minimize the bias that would result from selecting either a particular human annotator, or even the manually created gold standard, as a reference

for automatic evaluation; instead, we weigh all annotations from all human coders equally. Intuitively, we want to reward summaries that contain a high number of words considered to be relevant by most annotators. We formalize this notion in the following subsection.

6.2 Evaluation Metric

All evaluations are based on topically coherent segments from the dialogue excerpts of our corpus. As mentioned before, the segment boundaries were chosen from the human gold standard for the purpose of the global system evaluation.

For each segment s, for each annotator a, and for each word position w_i, we define a boolean word vector of annotations $\vec{w_{s,a}}$, each component $w_{s,a,i}$ being 1 if the word w_i is part of a nucleus IU or a satellite IU for that annotator and segment, and 0 otherwise. We then sum over all annotators' annotation vectors and normalize them by the number of annotators per segment (A) to obtain the average relevance vector for segment s, $\vec{r_s}$:

$$r_{s,i} = \frac{\sum_{a=1}^{A} w_{s,a,i}}{A}. \tag{7}$$

To obtain the summary accuracy score $sa_{s,n}$ for any segment summary with length n, we multiply the boolean summary vector $\vec{\text{summ}}_s$[26] by the average relevance vector $\vec{r_s}$, and then divide this product by the sum of the n highest scores within $\vec{r_s}$ (maximum achievable score), $\vec{\text{rsort}}_s$ being the vector $\vec{r_s}$ sorted by relevance weight in descending order:

$$sa_{s,n} = \frac{\vec{\text{summ}}_s \vec{r_s}}{\sum_{i=1}^{n} \text{rsort}_{s,i}} \tag{8}$$

It is easy to see that the summary accuracy score always is in the interval $[0.0, 1.0]$.

6.3 Global System Evaluation

Whereas section 5 was concerned with the design and evaluation of the individual system components, the goal here is to describe and analyze the quality of the global system, with all its components combined. In this section, we compare our DIASUMM system with the MMR baseline system, which operates without any dialogue-specific components, and with the LEAD baseline. We described the optimization and fine-tuning of the MMR system in subsection 5.6.6. The second column of Table 18 presents the average relevance scores for this MMR baseline, averaged over the five summary sizes of 5%, 10%, 15%, 20%, and 25% length, for the four `devtest` set and the four `eval` set subcorpora; the first column of this table shows the results for the LEAD baseline.

We used the optimized baseline MMR parameters and varied only the emphasis parameters for (1) false starts, (2) lead factor, and (3) Q-A sentences, to optimize the CLEAN summaries further. (Again, for this step, we used only the `devtest` subcorpora.) For each corpus in the `devtest` set, we determined the optimal parameter settings and report the corresponding results also for the `eval` set subcorpora. Column 3 in Table 18 provides the results for this optimized DIASUMM system. Further, in column 4, we provide the summary accuracy averages for the human gold standard (nucleus IUs only, fixed-length summaries). Table 19 shows the best emphasis parameter combinations for the DIASUMM summaries used in these evaluations.

We determined the statistical differences between the DIASUMM system and the two baselines for the `eval` set, using the Wilcoxon rank sum test for each of the four

26 Definition: 1 if $\text{summ}_{s,i}$ is contained in the summary, 0 otherwise.

Table 18
Average summary accuracy scores: `devtest` set and `eval` set subcorpora on optimized parameters, comparing LEAD, MMR baseline, DIASUMM, and the human gold standard.

Subcorpus	LEAD	MMR	DIASUMM	Gold [Nucleus IUs] (Size in %)
8E-CH	0.463	0.545	0.597	0.709 (13.1)
DT-NH	0.386	0.637	0.554	0.791 (20.9)
DT-XF	0.516	0.595	0.541	0.764 (11.4)
DT-MTG	0.488	0.594	0.606	0.705 (14.9)
4E-CH	0.438	0.526	0.614	0.793 (12.9)
EVAL-NH	0.692	0.526	0.506	0.850 (14.4)
EVAL-XF	0.378	0.564	0.566	0.790 (13.9)
EVAL-MTG	0.324	0.449	0.583	0.704 (16.0)

Table 19
Best emphasis parameters for the DIASUMM system, trained on the `devtest` set.

Corpus	False Start	Q-A	Lead Factor
CALLHOME	0.5	1.0	2.0
NEWSHOUR	0.5	2.0	1.0
CROSSFIRE	0.5	1.0	1.0
GROUP MEETINGS	0.5	1.0	3.0

Table 20
Average summary accuracy scores for different system configurations for the four different subcorpora.

Corpus	LEAD	MMR	DFF-ONLY	SB-ONLY	NO-QA	DIASUMM
4E-CH	.438	.526	.599	.547	.603	.614
EVAL-NH	.692	.526	.551	.608	.619	.506
EVAL-XF	.378	.564	.528	.525	.537	.566
EVAL-GMTG	.324	.449	.488	.513	.584	.583

subcorpora. Comparisons were made for each of the five summary sizes within each topical segment. For the CALLHOME and GROUP MEETINGS subcorpora, our DIASUMM system is significantly better than the MMR baseline ($p < 0.01$); for the two more formal subcorpora, NEWSHOUR and CROSSFIRE, the differences between the performance of the two systems are not significant. Except for on the NEWSHOUR subcorpus, both the MMR baseline and the DIASUMM system perform significantly better than the LEAD baseline.

6.4 Discussion

Table 20 shows the average performance of the following six system configurations, averaged over all topical segments and all summary sizes (5–25% length summaries; in configurations 3–5 below, components used are *in addition to* the core MMR summarizer):

1. LEAD: using the first n% of the words in a segment
2. MMR: the MMR baseline (tuned; see above)

3. DFF-ONLY: using the disfluency detection components (POS tagger, false-start detection, repetition detection), but no sentence boundary detection or question-answer linking
4. SB-ONLY: using the sentence boundary detection module, but no other dialogue-specific modules
5. NO-QA: a combination of DFF-ONLY and SB-ONLY (all preprocessing components used except for question-answer linking)
6. DIASUMM: complete system with all components (all disfluency detection components, sentence boundary detection, and Q-A linking)

We observe that in all subcorpora, except for CROSSFIRE, the addition of either the disfluency components or the sentence boundary component improves the summary accuracy over that of the MMR baseline. As we would expect, given the much higher frequency of disfluencies in the two informal subcorpora (CALLHOME, GROUP MEETINGS), the relative performance increase of DFF-ONLY over the MMR baseline is much higher here (about 10–15%) than for the two more formal subcorpora (5% and below). Looking at the performance increase of SB-ONLY, we find marked improvements over the MMR baseline for those two subcorpora that use the true original turn boundaries in the MMR baseline: GROUP MEETINGS and NEWSHOUR (>10%); for the two other subcorpora, the improvement is below 5%. Furthermore, the combination of the disfluency detection and sentence boundary detection components (NO-QA) improves the results over the configurations DFF-ONLY and SB-ONLY.

The situation is much less uniform when we add the question-answer detection component (this then corresponds to the full DIASUMM system): In the CROSSFIRE corpus, we have the largest performance increase (we also have the highest relative frequency of question speech acts here). For the two informal corpora, the change is only minor; for NEWSHOUR, the performance decreases substantially. We showed in Zechner and Lavie (2001), however, that in general, for dialogues with relatively frequent Q-A exchanges, the accuracy of a summary (informativeness) does not change significantly when the Q-A detection component is applied. On the other hand, the (local) coherence of the summary does increase significantly, but we cannot measure this increase with the evaluation criterion of summary accuracy used here.

To conclude, we have shown that using dialogue-specific components, with the possible exception of the Q-A detection module, can help in creating more accurate summaries for more informal, casual, spontaneous dialogues. When more formal conversations (which may even be partially scripted), containing relatively few disfluencies, are involved, either a simple LEAD method or a standard MMR summarizer will be much harder to improve upon.

7. Discussion and Directions for Future Work

The problem of how to generate readable and concise summaries automatically for spoken dialogues of unrestricted domains involves many challenges that need to be addressed. Some of the research issues are similar or identical to those faced in summarizing written texts (such as topic segmentation, determining the most salient/relevant information, anaphora resolution, summary evaluation), but other additional dimensions are added on top of this list, including speech disfluency detection, sentence boundary detection, cross-speaker information linking, and coping with imperfect speech recognition. The line of argument of this article has been that whereas using a

traditional approach for written text summarization (such as the MMR-based sentence selection component within DIASUMM) may be a good starting point, addressing the dialogue-specific issues is key for obtaining better summaries for informal genres.

We decided to focus on the three problems of (1) speech disfluency detection, (2) sentence boundary detection, and (3) cross-speaker information linking and implemented trainable system components to address each of these issues. Both the evaluations of the individual components of our spoken-dialogue summarization system and the global evaluations as well have shown that we can successfully make use of the SWITCHBOARD corpus (LDC 1999b) to train a system that works well on two other genres of informal dialogues, CALLHOME and GROUP MEETINGS. We conjecture that the reasons why the DIASUMM system was not able to improve over the MMR baseline for the two other corpora, which are more formal, lies in their very nature of being of a quite different genre: the NEWSHOUR and CROSSFIRE corpora have longer turns and sentences, as well as fewer disfluencies. We would also conjecture that their sentence structures are more complex than what we typically find in the other corpora of more colloquial, spontaneous conversations.

Future work will have to address the issue of whether the availability of training data for more formal dialogues (in size and annotation style comparable to the SWITCHBOARD corpus, though) could lead to an improvement in performance on those data sets, as well, or if even then a standard written-text-based summarizer would be hard to improve upon.

Given the complexity of the task, we had to make a number of simplifying assumptions, most notably about the input data for our system: We use perfect transcripts by humans instead of ASR transcripts, which, for these genres, typically show word error rates (WERs) ranging from 15% to 35%. Previous related work (Valenza et al. 1999; Zechner and Waibel 2000b) demonstrated that the actual WERs in summaries generated from ASR output are usually substantially lower than the full-ASR-transcript WER and can further be reduced by taking acoustically derived confidence scores into account.

We further did not explore the potential improvements of components as well as of the system overall when prosodic information such as stress and pitch is added as an input feature. Past work in related fields (Shriberg et al. 1998; Shriberg et al. 2000) suggests that particularly for ASR input, noticeable improvements might be achievable when such input is provided.

Although presegmentation of the input into topically coherent segments certainly is a useful step in summarization for any kind of texts (written or spoken), we have not addressed and discussed this issue in this article.

Finally, we think that there is more work needed in the area of automatically deriving discourse structures for spoken dialogues in unrestricted domains, even if the text spans covered might be only local (because of a lack of global discourse plans). We believe that a summarizer, in addition to knowing about the interactively constructed and coherent pieces of information (such as in question-answer pairs), could make good use of such structured information and be better guided in making its selections for summary generation. In addition, this discourse structure might aid modules that perform automatic anaphora detection and resolution.

8. Conclusions

We have motivated, implemented, and evaluated an approach for automatically creating extract summaries for open-domain spoken dialogues in informal and formal genres of multiparty conversations. Our dialogue summarization system DIASUMM

uses trainable components to detect and remove speech disfluencies (making the output more readable and less noisy), to determine sentence boundaries (creating suitable text spans for summary generation), and to link cross-speaker information units (allowing for increased summary coherence).

We used a corpus of 23 dialogue excerpts from four different genres (80 topical segments, about 47,000 words) for system development and evaluation and the disfluency-annotated SWITCHBOARD corpus (LDC 1999b) for training of the dialogue-specific components. Our corpus was annotated by six human coders for topical boundaries and relevant text spans for summaries. Additionally, we had annotations made for disfluencies, sentence boundaries, question speech acts, and the corresponding answers to those question speech acts.

In a global system evaluation we compared the MMR-based sentence selection component with the DIASUMM system using all of its components discussed in this article. The results showed that (1) both a baseline MMR system as well as DIASUMM create better summaries than a LEAD baseline (except for NEWSHOUR) and that (2) DIASUMM performs significantly better than the baseline MMR system for the informal dialogue corpora (CALLHOME and GROUP MEETINGS).

Acknowledgments

We are grateful to Alex Waibel, Alon Lavie, Jaime Carbonell, Vibhu Mittal, Jade Goldstein, Klaus Ries, Lori Levin, and Marsal Gavaldà for many discussions, suggestions, and comments regarding this work. We also want to commend the corpus annotators for their efforts. Finally, we want to thank the four anonymous reviewers for their detailed feedback on a preliminary draft, which greatly helped improve this article. This work was performed while the author was affiliated with the Language Technologies Institute at Carnegie Mellon University and was supported in part by grants from the U.S. Department of Defense.

References

Alexandersson, Jan and Peter Poller. 1998. Towards multilingual protocol generation for spontaneous speech dialogues. In *Proceedings of INLG-98*, Niagara-on-the-Lake, Canada, August.

Aone, Chinatsu, Mary Ellen Okurowski, and James Gorlinsky. 1997. Trainable, scalable summarization using robust NLP and machine learning. In *ACL/EACL-97 Workshop on Intelligent and Scalable Text Summarization*, Madrid.

Arons, Barry. 1994. Pitch-based emphasis detection for segmenting speech. In *Proceedings of ICSLP-94*, pages 1931–1934.

Berger, Adam L. and Vibhu O. Mittal. 2000. OCELOT: A system for summarizing Web pages. In *Proceedings of the 23rd ACM-SIGIR Conference*.

Bett, Michael, Ralph Gross, Hua Yu, Xiaojin Zhu, Yue Pan, Jie Yang, and Alex Waibel. 2000. Multimodal meeting tracker. In *Proceedings of the Conference on Content-Based Multimedia Information Access (RIAO-2000)*, Paris, April.

Brill, Eric. 1994. Some advances in transformation-based part of speech tagging. In *Proceedings of AAAI-94*.

Carbonell, Jaime, Yibing Geng, and Jade Goldstein. 1997. Automated query-relevant summarization and diversity-based reranking. In *Proceedings of the IJCAI-97 Workshop on AI and Digital Libraries*, Nagoya, Japan.

Carbonell, Jaime and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval*, Melbourne, Australia.

Carletta, Jean, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31.

Chen, Francine R. and Margaret Withgott. 1992. The use of emphasis to automatically summarize a spoken discourse. In *Proceedings of ICASSP-92*, pages 229–332.

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Davies, Mark and Joseph L. Fleiss. 1982. Measuring agreement for multinomial data. *Biometrics*, 38:1047–1051, December.

Garofolo, John S., Ellen M. Voorhees, Cedric G. P. Auzanne, and Vincent M. Stanford.

1999. Spoken document retrieval: 1998 evaluation and investigation of new metrics. In *Proceedings of the ESCA Workshop: Accessing Information in Spoken Audio*, pages 1–7, Cambridge, UK, April.

Garofolo, John S., Ellen M. Voorhees, Vincent M. Stanford, and Karen Sparck Jones. 1997. TREC-6 1997 spoken document retrieval track overview and results. In *Proceedings of the 1997 TREC-6 Conference*, pages 83–91, Gaithersburg, MD, November.

Gavaldà, Marsal, Klaus Zechner, and Gregory Aist. 1997. High performance segmentation of spontaneous speech using part of speech and trigger word information. In *Proceedings of the fifth ANLP Conference*, Washington, DC, pages 12–15.

Godfrey, J. J., E. C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of ICASSP-92*, volume 1, pages 517–520.

Grosz, Barbara J. and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Hearst, Marti A. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Heeman, Peter A. and James F. Allen. 1999. Speech repairs, intonational phrases, and discourse markers: Modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, 25(4):527–571.

Hirschberg, Julia, Steve Whittaker, Don Hindle, Fernando Pereira, and Amit Singhal. 1999. Finding information in audio: A new paradigm for audio browsing/retrieval. In *Proceedings of the ESCA Workshop: Accessing Information in Spoken Audio*, pages 117–122, Cambridge, UK, April.

Hori, Chiori and Sadaoki Furui. 2000. Automatic speech summarization based on word significance and linguistic likelihood. In *Proceedings of ICASSP-00*, pages 1579–1582, Istanbul, Turkey, June.

Jurafsky, Daniel, Rebecca Bates, Noah Coccaro, Rachel Martin, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Andreas Stolcke, Paul Taylor, and Carol Van Ess-Dykema. 1998. SwitchBoard discourse language modeling project: Final report. Research Note 30, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD.

Kameyama, Megumi, and I. Arima. 1994. Coping with aboutness complexity in information extraction from spoken dialogues. In *Proceedings of ICSLP 94*, pages 87–90, Yokohama, Japan.

Kameyama, Megumi, Goh Kawai, and Isao Arima. 1996. A real-time system for summarizing human-human spontaneous spoken dialogues. In *Proceedings of ICSLP-96*, pages 681–684.

Knight, Kevin and Daniel Marcu. 2000. Statistics-based summarization—Step one: Sentence compression. In *Proceedings of the 17th National Conference of the AAAI*.

Koumpis, Konstantinos and Steve Renals. 2000. Transcription and summarization of voicemail speech. In *Proceedings of ICSLP-00*, pages 688–691, Beijing, China, October.

Krippendorff, Klaus. 1980. *Content Analysis*. Sage, Beverly Hills, CA.

Kupiec, J., J. Pedersen, and F. Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th ACM-SIGIR Conference*, pages 68–73.

Lavie, Alon, Alex Waibel, Lori Levin, Michael Finke, Donna Gates, Marsal Gavaldà, Torsten Zeppenfeld, and Puming Zhan. 1997. Janus III: Speech-to-speech translation in multiple languages. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich.

Levin, Lori, Klaus Ries, Ann Thymé-Gobbel, and Alon Lavie. 1999. Tagging of speech acts and dialogue games in Spanish call home. In *Proceedings of the ACL-99 Workshop on Discourse Tagging*, College Park, MD.

Linguistic Data Consortium (LDC). 1996. CallHome and CallFriend LVCSR databases.

Linguistic Data Consortium (LDC). 1999a. Addendum to the part-of-speech tagging guidelines for the Penn Treebank project (Modifications for the SwitchBoard corpus). LDC CD-ROM LDC99T42.

Linguistic Data Consortium (LDC). 1999b. Treebank-3: Databases of disfluency annotated Switchboard transcripts. LDC CD-ROM LDC99T42.

Mani, Inderjeet, David House, Gary Klein, Lynette Hirschman, Leo Obrst, Therese Firmin, Michael Chrzanowski, and Beth Sundheim. 1998. The TIPSTER SUMMAC text summarization evaluation. Technical Report MTR 98W0000138, Mitre Corporation, October 1998.

Mani, Inderjeet and Mark T. Maybury, editors. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge.

Marcu, Daniel. 1999. Discourse trees are good indicators of importance in text. In I. Mani and M. T. Maybury, editors,

Advances in Automatic Text Summarization. MIT Press, Cambridge, pages 123–136.
Meteer, Marie, Ann Taylor, Robert MacIntyre, and Rukmini Iyer. 1995. Dysfluency annotation stylebook for the Switchboard corpus. Linguistic Data Consortium (LDC) CD-ROM LDC99T42.
Miike, Seiji, Etuso Itoh, Kenji Onon, and Kazuo Sumita. 1994. A full-text retrieval system with a dynamic abstract generation function. In *Proceedings of the 17th ACM-SIGIR Conference*, pages 318–327.
Nakatani, Christine H. and Julia Hirschberg. 1994. A corpus-based study of repair cues in spontaneous speech. *Journal of the Acoustic Society of America*, 95(3):1603–1616.
Passonneau, Rebecca J. and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139.
Quinlan, J. Ross. 1992. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
Rath, G. J., A. Resnick, and T. R. Savage. 1961. The formation of abstracts by the selection of sentences. *American Documentation*, 12(2):139–143.
Reimer, U. and U. Hahn. 1988. Text condensation as knowledge base abstraction. In *Proceedings of the fourth Conference on Artificial Intelligence Applications*, pages 338–344, San Diego.
Reithinger, Norbert, Michael Kipp, Ralf Engel, and Jan Alexandersson. 2000. Summarizing multilingual spoken negotiation dialogues. In *Proceedings of the 38th Conference of the Association for Computational Linguistics*, pages 310–317, Hong Kong, China, October.
Ries, Klaus, Lori Levin, Liza Valle, Alon Lavie, and Alex Waibel. 2000. Shallow discourse genre annotation in CALLHOME Spanish. In *Proceedings of the Second Conference on Language Resources and Evaluation (LREC-2000)*, Athens, May/June.
Rose, Ralph Leon. 1998. *The Communicative Value of Filled Pauses in Spontaneous Speech*. Ph.D. thesis, University of Birmingham, Birmingham, UK.
Salton, Gerard, editor. 1971. *The SMART Retrieval System—Experiments in Automatic Text Processing*. Prentice Hall, Englewood Cliffs, NJ.
Santorini, Beatrice. 1990. Part-of-Speech Tagging guidelines for the Penn Treebank project. Linguistic Data Consortium (LDC) CD-ROM LDC99T42.
Shriberg, Elizabeth E. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, University of Berkeley, Berkeley.
Shriberg, Elizabeth, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3–4):439–487.
Shriberg, Elizabeth, Andreas Stolcke, Dilek Hakkani-Tür, and Gökhan Tür. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1–2):127–154.
Stifelman, Lisa J. 1995. A discourse analysis approach to structured speech. In *AAAI-95 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford, March.
Stolcke, Andreas, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.
Stolcke, Andreas and Elizabeth Shriberg. 1996. Automatic linguistic segmentation of conversational speech. In *Proceedings of ICSLP-96*, pages 1005–1008.
Stolcke, Andreas, Elizabeth Shriberg, Rebecca Bates, Mari Ostendorf, Dilek Hakkani, Madeleine Plauche, Gökhan Tür, and Yu Lu. 1998. Automatic detection of sentence boundaries and disfluencies based on recognized words. In *Proceedings of ICSLP-98*, volume 5, pages 2247–2250, Sydney, December.
Teufel, Simone and Marc Moens. 1997. Sentence extraction as a classification task. In *ACL/EACL-97 Workshop on Intelligent and Scalable Text Summarization*, Madrid.
Valenza, Robin, Tony Robinson, Marianne Hickey, and Roger Tucker. 1999. Summarisation of spoken audio through information extraction. In *Proceedings of the ESCA Workshop: Accessing Information in Spoken Audio*, pages 111–116, Cambridge, UK, April.
Wahlster, Wolfgang. 1993. Verbmobil—Translation of face-to-face dialogs. In *Proceedings of MT Summit IV*, Kobe, Japan.
Waibel, Alex, Michael Bett, and Michael Finke. 1998. Meeting browser: Tracking and summarizing meetings. In *Proceedings of the DARPA Broadcast News Workshop*.
Ward, Wayne. 1991. Understanding spontaneous speech: The PHOENIX system. In *Proceedings of ICASSP-91*,

pages 365–367.
Whittaker, Steve, Julia Hirschberg, John Choi, Don Hindle, Fernando Pereira, and Amit Singhal. 1999. SCAN: Designing and evaluating user interfaces to support retrieval from speech archives. In *Proceedings of the 22nd ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 26–33, Berkeley, August.
Zechner, Klaus and Alon Lavie. 2001. Increasing the coherence of spoken dialogue summaries by cross-speaker information linking. In *Proceedings of the NAACL-01 Workshop on Automatic Summarization*, pages 22–31, Pittsburgh, June.
Zechner, Klaus and Alex Waibel. 2000a. DIASUMM: Flexible summarization of spontaneous dialogues in unrestricted domains. In *Proceedings of COLING-2000*, pages 968–974, Saarbrücken, Germany, July/August.
Zechner, Klaus and Alex Waibel. 2000b. Minimizing word error rate in textual summaries of spoken language. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2000)*, pages 186–193, Seattle, April/May.