AMIDA

Augmented Multi-party Interaction with Distance Access

`http://www.amidaproject.org/`

Integrated Project IST–033812

Funded under 6th FWP (Sixth Framework Programme)

Action Line: IST-2005-2.5.7 Multimodal interfaces

Deliverable D5.2: Report on multimodal content abstraction

Due date: 30/11/2007 **Submission date:** 07/12/2007

Project start date: 1/10/2006 **Duration:** 36 months

Lead Contractor: Tilman Becker **Revision:** 1

Project co-funded by the European Commission in the 6th Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

D5.2: Report on multimodal content abstraction

Abstract:

WP5's main objectives in the AMIDA project are (i) to provide WP6 and WP7 with components for inclusion in tools for remote meeting assistance and meeting browsing by (ii) extending and developing additional models and algorithms for multimodal structure and content analysis, and to provide a quantitative understanding of meeting structure while (iii) ensuring measurable quality by extending component evaluation schemes to AMIDA.

This deliverable describes results from the first 12 months of AMIDA in a range of fields, some of which are new while others extend work from AMI to the remote meeting scenario and/or to real-time and on-line algorithms. The list includes: decision detection, subjectivity recognition, dialog act recognition, summarization, topic segmentation, addressee classification, argumentation, dominance modeling, speech indexing and retrieval and automatic video editing.

Contents

1 Introduction

WP5 is concerned with understanding the multimodal structure of meetings and the analysis of the content of meetings. The ultimate objective of this work is to provide WP6 and WP7 with components for inclusion in tools for remote meeting assistance and meeting browsing. To ensure measurable quality, the component evaluation schemes developed in AMI are applied and extended to AMIDA where applicable.

This deliverable reports in detail on all areas that are covered in WP5. Some areas are new work in AMIDA while others extend work from AMI with new or refined approaches, extend the models to the remote meeting scenario, and move towards real-time and on-line algorithms. The areas covered fall into three broad categories: (i) understanding and classification of meeting structure emcompasses work in decision detection (sec. 2), subjectivity recognition (sec. 3), dialog acts (sec. 4), topic segmentation (sec. 7), addressing (sec. 8), argumenation (sec. 9), dominance modeling (sec. 10), and also automatic video editing (sec. 12) ; (ii) indexing and retrieval which concentrates on spoken terms (sec. 11); and (iii) content abstraction which creates summaries based on the features found in understaning the meeting structure (sec. 5).

Component evaluations are reported in each section and are based on the evaluation regimes developed within AMI. We have also performed an extrinsic evaluation of various types of summaries with a well-defined task and a meeting browser, measuring the effectiveness of our components in the context of a decision audit task (sec. 6).

2 Automatic Decision Detection from Meeting Recordings

2.1 Introduction

Recent advances in multimedia technologies have led to huge archives of audio and video recordings of meetings. Reviewing decisions is an aspect central to our organizational life [Pallotta et al., 2005, Rienks et al., 2005]. For example, it would be helpful for a new engineer assigned to a project to review the major decisions that have been made in previous meetings by watching the recordings. However, while it is straightforward to archive a meeting, finding out what decisions have been made from the recording is still a challenging task. Unless all decisions are recorded in meeting minutes or annotated in the audio-video recordings, it is difficult to locate the decision points using existing browsing and playback utilities alone. Moreover, a recent study [Pallotta et al., 2007] has shown that even when a standard keyword search utility is provided, it is still difficult to recover information about the argumentative process in the discussion (e.g., decision points).

Banerjee and Rudnicky [Banerjee et al., 2005] have demonstrated that it is easier to recover information for user queries if the meeting record includes discourse-level annotations, such as topic segmentation, speaker role, and meeting state[1]. To assist users in revisiting decisions within meeting archives, our goal is thus to automatically annotate decision-related information on the dialogue acts and discussion segments where decisions are made. As the development of such an automatic decision detection component is critical to the re-use of meeting archives [Whittaker et al., 2005], it is expected to lend support to the development of other downstream applications, such as computer-assisted meeting tracking and understanding (e.g., assisting in the fulfilment of the decisions made in meetings) and group decision support systems (e.g., constructing group memory) [Post et al., 2004, Romano and Nunamaker, 2001].

Previous research has developed descriptive models of meeting discussions. Some of this research focuses on modelling the dynamics [Niekrasz et al., 2005], while the other focuses on modelling the content [Marchand-Mailet, 2003, Rienks et al., 2005]. Although automatically extracting these argument models remains a challenging task, researchers have begun to make progress towards this goal [Galley et al., 2004, Gatica-Perez et al., 2005, Hillard et al., 2003a, Hsueh and Moore, 2007a, Purver et al., 2006a, Wrede and Shriberg, 2003a].

In this paper, we present the AMIDA DecisionDetector, which performs automatic decision detection in meeting speech and provides visual aids for users wishing to review decisions. In particular, we are interested in locating decision-related information at two levels of granularity: topic segments and dialogue acts. First, the system detects decision-related topic segments in which meeting participants have reached at least one decision. As shown in Figure 1, this allows users to get an overview of the decisions made in previous meetings by browsing the topics of the decision-related segments (e.g., those shaded in red in Figure 1).

Second, the system detects decision-related dialogue acts (DAs) by looking for DAs which are extract-worthy[2] and reflective of the content of the decision discussions. As

[1] Meeting states include discussion, presentation and briefing.

[2] Extract-worthy DAs are those that should be selected into the extractive summary of a meeting.

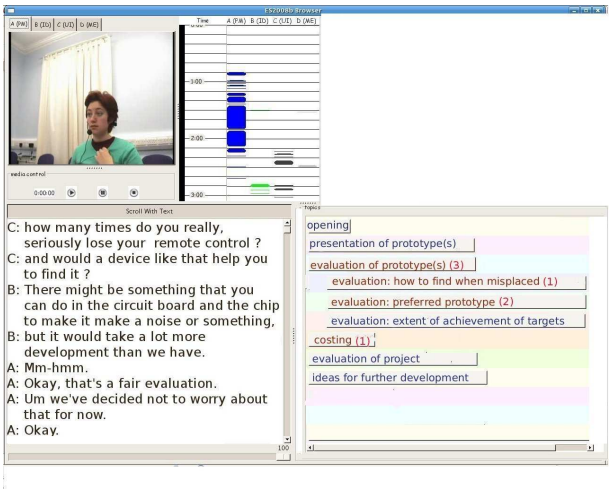


Figure 1: Example application that demonstrates the use of decision-related topic segment information. The bottom right component shows a list of topic segments in an example meeting. The topic segments shaded in red are those that contain at least one decisions. The number shown in the parenthesis following each topic segment label indicates the number of decisions reached within the topic segment.

shown in Figure 2, this allows users to obtain details about the decisions they are particularly interested in by reviewing the relevant decision-related DAs. For example, if a user wants to know more about the design decision relating to "how to find (the remote) when misplaced", they can interpret the decision as "not to worry about designing a function to find the remote when misplaced" by looking at the extract shown in the bottom right component of Figure 2.

2.2 Related Work

Spontaneous face-to-face dialogues in meetings violate many assumptions made by techniques previously developed for broadcast news (e.g., TDT and TRECVID), telephone conversations (e.g., Switchboard) [Godfrey et al., 1992] , and human-computer dialogues (e.g., DARPA Communicator) [Eskenazi et al., 1999] . In order to develop techniques for understanding multiparty dialogues, smart meeting rooms have been built at several institutes to record large corpora of meetings in natural contexts, including CMU [Waibel et al., 2001], LDC [Cieri et al., 2002], NIST [Garofolo et al., 2004], ICSI [Janin et al., 2003], and in the context of the IM2/M4 project [Marchand-Mailet, 2003]. More recently, scenario-based meetings, in which participants are assigned to different roles and given specific tasks, have been recorded in the context of the CALO project (the Y2 Scenario Data) [CALO, 2006] and the AMI project [Carletta et al., 2006].

The availability of meeting corpora has enabled researchers to begin to develop descriptive models of meeting discussions. Some researchers are modelling the dynamics of the meeting, exploiting dialogue models previously proposed for dialogue management. For example, Niekrasz et al. [Niekrasz et al., 2005] use the Issue-Based Information System (IBIS) model [Kunz and Ritte, 1970] to incorporate the history of dialogue moves into the

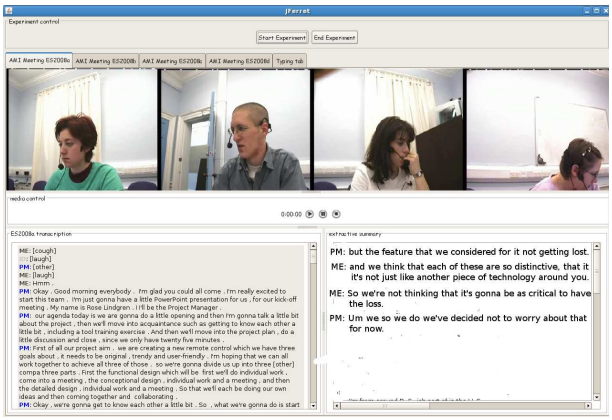


Figure 2: Example application that demonstrates the use of decision-related DA information. The bottom right component shows a set of decision-related DA extracts that are representive of the design decision of "how to find (the remote) when misplaced".

Multi-Modal Discourse (MMD) ontology. Other researchers are modelling the content of the meeting using the type of structures proposed in work on argumentation. For example, Rienks et al. [Rienks et al., 2005] have developed an argument diagramming scheme to visualize the relations (e.g., positive, negative, uncertain) between utterances (e.g., statement, open issue), and Marchand et al. [Marchand-Mailet, 2003] propose a schema to model different argumentation acts (e.g., accept, request, reject) and their organization and synchronization. Decisions are often seen as a by-product of these models.

Automatically extracting these argument models is a challenging task. However, researchers have begun to make progress towards this goal. For example, Gatica et al. [Gatica-Perez et al., 2005] and Wrede and Shriberg [Wrede and Shriberg, 2003a] automatically identify the level of emotion in meeting spurts (e.g., group level of interest, hot spots). Other researchers have developed models for detecting agreement and disagreement in meetings, using models that combine lexical features with prosodic features (e.g., pause, duration, F0, speech rate) [Hillard et al., 2003a] and structural information (e.g., the previous and following speaker) [Galley et al., 2004]. More recently, Purver et al. [Purver et al., 2006a] have tackled the problem of detecting one type of decision, namely action items, which embody the transfer of group responsibility. However, no prior work has addressed the problem of automatically identifying decision-making units more generally in multiparty meetings. Moreover, no previous research has provided a quantitative account of the effects of different feature types on the task of automatic decision detection.

2.3 Methodology

2.3.1 Decision Detection as Classification

Our aim is to develop models for automatically detecting segments of conversation that contain decisions directly from the audio recordings and transcripts of the meetings, and to identify the feature combinations that are most effective for this task.

Meetings can be viewed at different levels of granularity. In this study, we first consider

how to detect the dialogue acts that contain decision-related information (Decision-related DAs). Since it is often difficult to interpret a decision without knowing the current topic of discussion, we are also interested in detecting decision-related segments at a coarser level of granularity: topic segments. The task of automatic decision detection therefore is evaluated at these two levels of granularity: detecting decision-related DAs and detecting decision-related topic segments.

In our study we first empirically identified the features that are most characteristic of decision-making dialogue acts and then computationally integrated the characteristic features to locate the decision-related DAs in meeting archives. To develop computational models to understand multiparty meetings, previous research on automatic meeting understanding and tracking has commonly utilized a classification framework, in which variants of generative and conditional models are computed directly from data. In this study, we use a Maximum Entropy (MaxEnt) classifier to combine the decision characteristic features to predict decision-related DAs and decision-related topic segments.

2.3.2 Manual Decision Annotations

In this study, we use a set of 50 scenario-driven meetings (approximately 37,400 DAs) that have been segmented into dialogue acts and annotated with decision information in the AMI meeting corpus [Carletta et al., 2006]. These meetings are driven by a scenario, wherein four participants play the role of Project Manager, Marketing Expert, Industrial Designer, and User Interface Designer in a design team in a series of four meetings. Participants participated in only one series of 4 meetings. The corpus includes manual transcripts for all meetings as well as individual sound files recorded by close-talking microphones for each participant and cross-talking sound files recorded by an 8-element circular microphone array.

The meeting recordings have been annotated at several levels, including dialogue acts (DAs) and topics. The DA annotation scheme for the AMI corpus consists of 15 dialogue act types, which can be organised into five major groups:

- Information (31.9%): giving and eliciting information, e.g., “Suggestion”.
- Action (9.8%): making or eliciting suggestions or offers, e.g., “Elicit-suggestion”.
- Commenting on the discussion (22.6%): making or eliciting assessments and comments about understanding, e.g., “Assessment”.
- Segmentation (31.8%): not contributing to the content but allowing segmentation of the discourse, e.g., “Backchannel”, “Stall”, and “Fragment”.
- Other (3.9%): a remainder class for utterances which convey an intention, but do not fit into the four previous categories.

Topic segmentation and labels have also been annotated in the AMI meeting corpus. Annotators had the freedom to mark a topic as subordinated (down to two levels) wherever appropriate. In this work, we have flattened the structure into a hierarchy of two layers:

top-level (TOP) and subtopic level (SUB). As the AMI meetings are scenario-driven, annotators are expected to find that most topics recur. Therefore, they are given a standard set of descriptions that can be used as labels for each identified topic segment. In particular, the annotators explicitly identify those parts of the meeting that refer to the meeting process (e.g., opening, closing, agenda/equipment issues), or are simply irrelevant (e.g., chitchat). To capture the common characteristics of these off-topic discussion segments, we have collapsed these segments into one category: functional segments (FUNC). The AMI scenario meetings take, on average, 30 minutes (around 800 DAs) and contain eight top-level topic segments and seven sub-topic segments per meeting. (See Table 1 for a break-down description of different types of segments.)

	ALL	TOP	SUB	FUNC
Average number of segments per meeting	13.65	7.67	7.05	3.54
Average duration per meeting (in minutes)	2.85	3.55	1.94	1.05
Average duration per meeting (in DAs)	71.2	88.84	50.41	22.19

Table 1: *Basic statistics of discourse segmentation annotations in the AMI corpus. ALL segments refer to the combination of TOP and SUB segments.*

Decision-Related Dialogue Acts

It is difficult to determine whether a DA contains information relevant to a decision without knowing what decisions have been made in the meeting. Therefore, in this study decision-related DAs are annotated in a two-phase process. First, annotators are asked to browse through the meeting record and write an abstractive summary about the decisions that have been made in the meeting. In this phase, another group of three annotators are also asked to produce extractive summaries by selecting a subset (around 10%) of DAs which form a summary of this meeting. Annotators are instructed to produce these summaries for an absent project manager.

Finally, this group of annotators are asked to judge whether the DAs in the extractive summary (heceforth called extracted DAs or EDAs) support any of the sentences in the abstractive decision summary; if a EDA is related to any sentence in the decision section of the abstractive summary, a "decision link" from the EDA to the decision sentence in the abstractive summary is added. For those EDAs that do not have any closely related sentence in the abstract, the annotators are not obligated to specify a link. We then label the EDAs that have one or more decision links as decision-related DAs.

In the 50 meetings we used for our experiments, annotators found on average four decisions per meeting and specified around two decision links for each decision sentence in the abstractive summary. Overall, 554 out of 37,400 DAs have been annotated as decision-related DAs, accounting for 1.4% of all DAs in the data set and 12.7% of the original extractive summaries (which consist of the extracted DAs). An earlier analysis established the intercoder reliability of the two-phase process at a kappa ranging from 0.5 to 0.8. In these experiments, for each meeting in the 50-meeting dataset we randomly choose the decision-related DA annotation of one annotator as the source of ground truth data.

Decision-Related Topic Segments

Decision-related topic segments are operationalized as the topic segments that contain one or more decision-related DAs. Overall, 198 out of 623 (31.78%) topic segments in the 50-meeting dataset are decision-related topic segments. As the meetings we use are driven by a scenario, we expect to find that interlocutors are more likely to reach decisions when certain topics from the predetermined agenda are brought up, or when the discussions are related to the decisions made in previous meetings. For example, 80% of the segments labelled as Costing and 58% of those labelled Budget are decision-related topic segments, whereas only 7% of the Existing Product segments and none of the Trend-Watching segments are decision-related topic segments. (See Table 2 for a break-down of different types of decision-related segments.)

	ALL	TOP	SUB	FUNC
Percentage of Decision-related topicsegments per meeting (%)	33%	31%	35%	4%
Average number of decision-related dialogue acts per segment	3.7	4.5	2.76	3.83

Table 2: *Characteristics of topic segments that contain decision-related DAs.*

2.4 Feature Extraction

To provide a qualitative account of the effect of different feature types on the task of automatic decision detection, we have conducted empirical analysis on four major types of features: lexical, prosodic, contextual and topical features.

2.4.1 Lexical Features

Previous research has studied lexical differences (i.e., occurrence counts of N-grams) between various aspects of speech, such as topics [Hsueh and Moore, 2006], speaker gender [Boulis and Ostendorf, 2005], and story-telling conversation [Gordon and Ganesan, 2005]. As we expect that lexical idiosyncracies also exist in decision-related conversations, we generated language models from the decision-related Dialogue Acts in the corpus. Comparison of the language models generated from the decision-related DAs and the rest of the conversations shows that some differences exist between the two models: (1) decision related conversations, whose context ranges from utterances to topic segments, are more likely to contain *we* than *I* and *You*; (2) in decision-related conversations there are more explicit mentions of topical words, such as *advanced chips* and *functional design*; (3) in decision-making conversations, there are fewer negative expressions, such as *I don't think* and *I don't know*. In an exploratory study using unigrams, as well as bigrams and trigrams, we found that using bigrams and trigrams does not improve the accuracy of classifying decision-related DAs, and therefore we include only unigrams in the set of lexical features in the experiments reported in Section 2.5.

2.4.2 Prosodic Features

Functionally, prosodic features, i.e., energy, and fundamental frequency (F0), are indicative of segmentation and saliency. In this study, we follow Shriberg and Stolcke's [Shriberg et al., 2001] direct modelling approach to manifest prosodic features as duration, pause, speech rate, pitch contour, and energy level. We utilize the individual sound files provided in the AMI corpus. To extract prosodic features from the sound files, we use the Snack Sound Toolkit to compute a list of pitch and energy values delimited by frames of 10 ms, using the normalized cross correlation function. Then we apply a piecewise linearisation procedure to remove the outliers and average the linearised values of the units within the time frame of a word. Pitch contour of a dialogue act is approximated by measuring the pitch slope at multiple points within the dialogue act, e.g., the first and last 100 and 200 ms. The rate of speech is calculated as both the number of words spoken per second and the number of syllables per second. We use Festival's speech synthesis front-end to return phonemes and syllabification information. An exploratory study showed the benefits of including immediate prosodic contexts, and thus we also include prosodic features of the immediately preceding and following dialogue acts. Table 3 contains a list of automatically generated prosodic features used in this study.

Type	Feature
Duration	Number of words spoken in current, previous and next DA Duration (in seconds) of current, previous and next DA
Pause	Amount of silence (in seconds) preceding a DA Amount of silence (in seconds) following a DA
Speech rate	Number of words spoken per second in current, previous and next DA Number of syllables per second in current, previous and next DA
Energy	Overall energy level Average energy level in the first, second, third, and fourth quarter of a DA
Pitch	Maximum and minimum F0, overall slope and variance Slope and variance at the first 100 and 200 ms and last 100 and 200 ms, at the first and second half, and at each quarter of the DA

Table 3: *Prosodic features used in this study.*

2.4.3 DA-based Features

From our qualitative analysis, we expect that contextual features specific to the AMI corpus, such as the speaker role (i.e., PM, ME, ID, UID) and meeting type (i.e., kick-off, conceptual design, functional design, detailed design) to be characteristic of the decision-related DAs. Analysis shows that (1) participants assigned to the role of PM produce 42.5% of the decision-related DAs, and (2) participants make relatively fewer decisions in the kick-off meetings. Analysis has also demonstrated a difference in the type, the reflexivity[3] and the number of addressees, between the decision-related DAs and the non-

[3] According to the annotation guidelines, the reflexivity reflects on how the group is carrying on the task. Interlocutors pause to evaluate group performance less often when making decisions.

decision-related DAs. For example, dialogue acts of type ***inform***, ***suggest***, ***elicit assessment*** and ***elicit inform*** are more likely to be decision-related DAs.

We have also found that immediately preceding and following dialogue acts are important for identifying decision-related DAs. For example, ***stalls*** and ***fragments*** are more likely to precede and ***fragments*** more likely to follow a decision-related DA.[4] In contrast, there is a lower chance of seeing ***suggest*** and elicit-type DAs (i.e., ***elicit-inform***, ***elicit-suggestion***, ***elicit-assessment***) in the preceding and following decision-related DAs. A complete list of contextual features used in this study are shown in Table 4.

Position (in words, in seconds and in percentage)
Speaker role
Meeting type
Type of the current dialogue act
Type of the immediate preceding dialogue act
Type of the imeediate following dialogue act

Table 4: *DA-based features used in this study.*

2.4.4 Topical Features

As reported in Section 2.3.2, we find that interlocutors are more likely to reach decisions when certain topics are brought up. Also, we expect decision-making conversations to take place towards the end of a topic segment. Therefore, in this study we include the following features: the label of the current topic segment, the position of the DA in the topic segment (measured in words, in seconds, and in %), the distance of the DA from the previous topic shift (both at the top-level and sub-topic level)(measured in seconds), the duration of the current topic segment (both at the top-level and sub-topic level)(measured in seconds). A complete list of topical features are listed in Table 5.

Topic label
Position in a topic segment (in words, in seconds, and in %)
Distance to the previous topic shift (both at the top-level and sub-topic level) (in seconds)
Duration of the current topic segment (both at the top-level and sub-topic level) (in seconds)

Table 5: *Topical features used in this study.*

[4]STALL is where people start talking before they are ready, or keep speaking when they haven't figured out what to say; FRAGMENT is a segment which is not really speech to be transcribed, or where the speaker did not get far enough to express the intention.

	TRAIN SET						TEST SET					
	Exact Match			Lenient Match			Exact Match			Lenient Match		
Accuracy	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BASELINE-1(PROS)	0.60	0.10	0.17	0.65	0.22	0.32	0.16	0.03	0.05	0.22	0.10	0.13
BASELINE-2(LX1)	0.75	0.72	0.73	0.79	0.87	0.83	0.20	0.20	0.20	0.32	0.44	0.36
DA	0.52	0.01	0.02	0.62	0.02	0.04	0.22	0.01	0.02	0.24	0.01	0.03
TOPIC	0.60	0.09	0.16	0.73	0.13	0.22	0.22	0.05	0.07	0.35	0.08	0.13
ALL-PROS	0.84	0.70	0.76	0.89	0.86	0.87	0.31	0.24	0.27	0.45	0.39	0.41
ALL-LX1	0.72	0.38	0.50	0.81	0.60	0.68	0.28	0.18	0.22	0.46	0.35	0.40
ALL-DA	0.89	0.7826	0.8274	0.92	0.91	0.91	0.25	0.25	0.25	0.40	0.43	0.41
ALL-TOPIC	0.84	0.69	0.76	0.88	0.86	0.87	0.26	0.23	0.24	0.38	0.45	0.41
ALL	0.86	0.75	0.80	0.90	0.90	0.90	0.28	0.25	0.26	0.42	0.47	0.44

Table 6: *Effects of different combinations of features on detecting decision-related DAs from extractive summaries*

2.5 Experiment 1: Detecting Decisions from Extractive Summaries

2.5.1 Decision-related DA Detection

Detecting decision-related DAs is the first step of automatic decision detection. For this purpose, we trained MaxEnt models to classify decision-related DAs in the set of dialogue act extracts, that is, those DAs that have been manually selected as extract-worthy. In Experiment 2, we trained models to classify decision-related DAs directly on entire transcripts, without having to manually annotate the extractive summaries. In this experiment, we focused on detecting decisions from extract-worthy DAs first because we wanted to examine the effects of different features on the task of decision detection in isolation.

We performed a 5-fold cross validation on the set of 50 meetings. In each fold, we trained MaxEnt models from the feature combinations in the training set, wherein each of the extracted dialogue acts has been labelled as either positive (POS) or negative (NEG), i.e., occurring or not occurring in the extract. Then, the models were used to classify unseen instances in the test set as either POS or NEG. In Section2.4, we described the four major types of features used in this study: unigrams (LX1), prosodic (PROS), DA-based (DA), and topical (TOPIC) features. For comparison, we report the naive baseline obtained by training the models on the prosodic features alone, since prosodic features can be generated fully automatically. We also report on another baseline which is obtained on the semi-automatically generated unigram features.[5] The different combinations of features we used for training models can be divided into the following four groups: (A) using prosodic features alone (BASELINE-1) and lexical features alone (BASELINE-2); (B) using DA-based and topical features alone (DA, TOPIC); (C) using all available features except one of the four types of features (ALL-LX1, ALL-PROS, ALL-DA, ALL-TOPIC); and (D) using all available features (ALL).

[5]The reason that it is semi-automatically generated is because the unigram features used here were computed from the manual transcripts.

We report results both on the training set and on the test set in Table 6. The rightmost three columns in the training set results and those in the test set results are the results obtained using a lenient match measure, allowing a window of 20 seconds preceding and following a hypothesized decision-related DA for recognition. The better results show that there is room for ambiguity in the assessment of the exact timing of decision-related DAs. The results show that models trained with all features (ALL), including lexical, prosodic, contextual and topical features, yield substantially better performance than the baseline on the task of detecting decision-related DAs.

Rows 1 - 4 in Table 6 report the performance of models in BASELINE and Group B, which are trained with a single type of feature. Note that decision-related DAs account for only 12.7% of the DA extracts, i.e., the subset of DAs that are extract-worthy. Therefore a random baseline would be generated according to the ratio of 12.7%. We expect the baseline that is obtained on lexical features (LX1) and prosodic features (PROS) alone to be harder to beat than the randomly generated baseline. Among the four single feature classes, lexical features are the most predictive features when used alone. We performed sign tests to determine whether there are statistically significant differences among the other three models, the LX1 baseline, and the ALL model that combines all four feature classes. Results show that none of these feature classes when used alone can outperform the baseline and the ALL model ($p < 0.001$).

To study the relative effect of the different feature types, Rows 5 - 8 in Table 6 report the performance of models in Group C, which are trained with all available features except LX1, PROS, DA and TOPIC features, respectively. The amount of degradation in the overall accuracy (F1) of each of the models in relation to that of the ALL model indicates the contribution of the feature type that has been left out of the model. We also performed sign tests to examine the differences among these models and the ALL model. We find that the ALL model outperforms all of these models ($p < 0.001$).

Overall, we find that combining all of the four feature classes is beneficial to the accuracy of the model. A closer investigation of the precision and recall of these leave-one-out models shows that (1) taking away any of these feature classes greatly degrades recall, and (2) taking away LX1 and PROS slightly improves precision, while taking off TOPIC and DA slightly degrades precision. The comparison of recall shows that these feature classes are complementary to each other on the task of recalling decision-related DAs – each of the four feature classes contributes to some part of the recall performance. Among them, lexical and prosodic features contribute the most, followed by DA-based and topical features. The mixed results for precision stem from the fact that, on the one hand, some feature classes, such as topical features, are tailored to recognize decision-related DAs in particular types of topic segments. Therefore, combining topical features improves the precision of the models by accurately recognizing the decision-related DAs that occur in those types of topic segments. On the other hand, including lexical and prosodic features is detrimental to the precision of models on the task of detecting decision-related DAs, because there are non DA-based DAs that have similar lexical and prosodic characteristics.

2.5.2 Decision-related Topic Segment Detection

As the task of detecting decision-related topic segments can be viewed as a task of recognizing decision-related DAs in a wider window, the results in Table 7 are better than those reported in Table 6, achieving at best 91% overall accuracy in the training set and 67% in the test set. The model that combines all features (ALL) yields significantly better results than all of the models that are trained with a single feature class except LX1 (BASELINE). Please note that compared to the randomly generated baseline in which a topic segment has only 31.78% chance of containing one or more decisions (c.f. Table 2), the baseline we use here is a higher baseline.

Rows 1-4 suggest that lexical model (LX1), compared to the other models in Group (B) that are trained with one single feature class, are the most predictive in terms of overall accuracy. Sign tests confirm the advantage of using LX1 ($p < 0.05$). Interestingly, the model that is trained with topical features alone (TOPIC) alone yields precision as good as using all of the features. As mentioned in the previous experiment, this result stems from the fact that decisions are more likely 0.8406 0.6858 0.7552 to occur in certain types of topic segments (c.f. Section 2.3.2). In turn, training models with topical features helps eliminate incorrect predictions of decision-related DAs in these types of topic segments. However, the accuracy gain of the TOPIC model on detecting decision-related DAs in certain types of topic segments does not extend to all types of decision-related topic segments. This is shown by the significantly lower recall of the TOPIC model over the baseline ($p < 0.001$).

Finally, Rows 5-8 and Row 9 report the performance of the models in Group (C) and the model that is trained with all available features (ALL) on the task of detecting decision-related topic segments. Calculating how much the overall accuracy of the models in Group C degrades from the ALL model shows that the most predictive features are the lexical features, followed by the prosodic features. Sign tests confirm that the ALL model outperforms the models that leave out lexical and prosodic features ($p < 0.05$). However, the ALL model does not outperform the model that leaves out DA-based and topical features due to the degradation of the recall.

	Decision-related Topic Segment					
	Training set			Test set		
Accuracy	Precision	Recall	F1	Precision	Recall	F1
BASLINE-1(PROS)	0.77	0.35	0.48	0.50	0.35	0.39
BASELINE-2(LX1)	0.78	0.93	0.85	0.56	0.79	0.66
DA	0.82	0.03	0.06	0.40	0.05	0.09
TOPIC	0.85	0.16	0.27	0.58	0.16	0.24
ALL-LX1	0.88	0.68	0.76	0.65	0.56	0.59
ALL-PROS	0.91	0.89	0.90	0.62	0.62	0.62
ALL-DA	0.91	0.95	0.93	0.58	0.73	0.65
ALL-TOPIC	0.87	0.92	0.90	0.59	0.77	0.66
ALL	0.91	0.92	0.91	0.60	0.70	0.65

Table 7: *Effects of different combinations of features on detecting decision-related topic segments from extractive summaries.*

	TRAIN SET						TEST SET					
Decision-Related	Dialogue Act			Topic Segment			Dialogue Act			Topic Segment		
Accuracy	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BASELINE(PROS)	0.65	0.22	0.32	0.77	0.35	0.48	0.22	0.10	0.13	0.50	0.35	0.39
LX1	0.79	0.87	0.83	0.78	0.93	0.85	0.32	0.44	0.36	0.56	0.79	0.66
LX1+BPROS	0.85	0.87	0.86	0.84	0.93	0.88	0.37	0.47	0.41	0.59	0.76	0.66
LX1+DA	0.85	0.81	0.83	0.86	0.90	0.88	0.41	0.38	0.39	0.63	0.72	0.67
LX1+TOPIC	0.90	0.88	0.89	0.90	0.93	0.91	0.37	0.38	0.37	0.59	0.69	0.63
ALL	0.90	0.90	0.90	0.91	0.92	0.91	0.42	0.47	0.44	0.60	0.70	0.65

Table 8: *Effects of combining lexical and other features on detecting decision-related DAs and decision-related topic segments from extractive summaries.*

2.5.3 Effects of Combining Lexical Features with Other Feature Classes

As the model that is trained with lexical features alone (LX1) alone yields overall accuracy as good as using all of the features, we are interested in knowing whether it is essential to combine lexical features with other types of features. Table 8 further shows that combining prosodic, DA-based, and topical features with LX1 (LX1+BPROS) can improve the precision of the model but not the recall. This result stems from the fact that those decision-characteristic words, such as content words, are also quite likely to appear in many other dialogue acts that are not directly related to decisions. In turn, combining other decision-characteristic features into the model helps eliminate incorrect predictions of decision related DAs in these other non-decision related DAs. However, this effect does not improve the recall of decision-related topic segments. This is because most of the eliminated non-decision related DA predictions are located in the same major topic segments wherein interlocutors are likely to refer to the same terms.

In sum, we find that models that combine lexical, prosodic, contextual and topical features yield the best results on the task of detecting decision-related dialogue acts, while models that combine lexical with any one of the other feature classes are sufficient for the task of detecting decision-related topic segments.

2.6 Experiment 2: Detecting Decisions from Entire Transcripts

As opposed to Experiment 1, which detects decision-related DAs on only the parts of meetings that have been identified as extract-worthy, in this experiment we trained models to detect decision-related DAs directly from entire transcripts. We expect this task to be much more challenging as the imbalance between positive and negative cases is even more prominent. The proportion of positive cases has gone from 12.7% down to 1.4%. For comparison, we still use the lexical models trained with the unigram lexical features (LX1) as our baseline. As mentioned in Experiment 1, the LX1 baseline raises a bar higher than the randomly generated baseline [6]

Table 9 reports the performance on both the training (40 meetings) and the test set (10

[6]Please note that the LX1 features used here are obtained on manual transcripts; so the lexical models can only be viewed as being trained semi-automatically.

	TRAIN SET						TEST SET					
Decision-Related	Dialogue Act			Topic Segment			Dialogue Act			Topic Segment		
Accuracy	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BASELINE(LX1)	0.40	0.60	0.48	0.55	0.81	0.66	0.26	0.48	0.33	0.48	0.81	0.60
ALL-LX1	0.80	0.13	0.22	0.90	0.17	0.28	0.44	0.09	0.14	0.63	0.21	0.31
ALL-PROS	0.86	0.57	0.68	0.90	0.66	0.76	0.37	0.21	0.27	0.61	0.49	0.53
ALL-DA	0.87	0.62	0.72	0.89	0.72	0.79	0.42	0.32	0.35	0.64	0.56	0.59
ALL-TOPIC	0.82	0.48	0.60	0.89	0.63	0.73	0.29	0.24	0.25	0.59	0.51	0.54
ALL	0.89	0.49	0.62	0.92	0.58	0.70	0.46	0.24	0.31	0.68	0.48	0.56

Table 9: *Effects of different combinations of features on detecting decision-related DAs and topic segments from entire transcripts*

meetings). Because previous work has shown that ambiguity exists in the assessment of the exact timing of decision-related DAs, in Table 9 we reported the results obtained by the lenient match measure. The task of detecting decision-related topic segments can be viewed as that of detecting decision-related DAs in a wider window. The right most three columns of the training set and test set results in Table 6 show the results of detecting decision-related topic segments.

The results demonstrate that, compared to the LX1 baseline, models trained with all features (ALL), including lexical, prosodic, DA-based and topical features, yield notably better precision on the task of decision-related topic segment prediction, 92% on the training set and 68% on the test set. However, in the test set, the overall accuracy (F1 score) of the combined models is relatively worse than the baseline, due to the substantially lower recall rate.

To study the relative effect of the different feature types, Rows 2-5 in the table report the performance of models in Group C, which are trained with all available features except LX1, PROS, DA and TOPIC, respectively. The amount of degradation in the overall accuracy (F1) of each of the models in relation to that of the ALL model indicates the contribution of the feature type that has been left out. For example, we find that the ALL model outperforms all except the model trained by leaving out DA-based features (ALL-DA). A closer investigation of the precision and recall of the ALL-DA model shows that including the DA-based features is detrimental to recall but beneficial for precision. This effect stems from the fact that decisions are more likely (1) to occur in certain types of dialogue acts, such as "Inform", "Suggest", "Elicit-Assessment", and "Elicit-Inform", and (2) to be preceded and followed by segmentation-type dialogue acts, such as "Stall" and "Fragment". Therefore, training models with DA-based features, such as the DA class of the current DA and its immediate context, helps eliminate incorrect predictions of decision-related DAs.

In sum, results suggest the following for the task of detecting decision points from entire transcripts: (1) lexical features are the most predictive in terms of overall accuracy, despite low precision, (2) prosodic features have positive impacts on precision but not on recall, and (3) DA-based and topical features are both beneficial to precision but detrimental to recall.

	TRAIN SET						TEST SET					
Decision-Related	Dialogue Act			Topic Segment			Dialogue Act			Topic Segment		
Accuracy	P	R	F1	P	R	F1	P	R	F1	P	R	F1
EXTRACT (MANUAL-15DA)	0.91	0.79	0.84	0.92	0.85	0.88	0.46	0.48	0.45	0.67	0.68	0.65
EXTRACT (MANUAL-5DA)	0.88	0.88	0.88	0.87	0.92	0.89	0.45	0.56	0.49	0.64	0.79	0.70
EXTRACT (AUTO-5DA)	0.87	0.89	0.88	0.86	0.91	0.88	0.41	0.49	0.44	0.62	0.71	0.64
ALLTRAN (MANUAL-15DA)	0.90	0.53	0.67	0.92	0.62	0.73	0.43	0.28	0.33	0.68	0.46	0.54
ALLTRAN (MANUAL-5DA)	0.89	0.57	0.69	0.88	0.66	0.75	0.44	0.25	0.31	0.65	0.48	0.54
ALLTRAN (AUTO-5DA)	0.89	0.61	0.73	0.91	0.70	0.79	0.43	0.31	0.35	0.61	0.51	0.55

Table 10: *Effects of different versions of DA class features on detecting decision-related DAs and topic segments. The first three rows (EXTRACT) are the results obtained on extractive summaries. The last three rows (ALLTRAN) are the reesults obtained on entire transcripts.*

2.7 Experiment 3: Exploring Automatically Generated DA Class Features in Automatic Decision Detection

As our ultimate goal is to operate AMI DecisionDetector in an automatic fashion, we evaluate the impact of the automatically generated DA class features on the task of detecting decision-related DAs and topic segments. We have utilized the 5-class DA predictions (AUTO-5DA) generated in [Dielmann and Renals, 2007b]. To understand whether the automatically generated features caused any degradation, we train models which combine all available lexical, prosodic and topical features with the AUTO-5DA features. We then evaluate the AUTO-5DA model against other models which combine the other features with the two types of manually annotated dialogue act class features: MANUAL-5DA and MANUAL-15DA. The results reported here are obtained by operating AMI DecisionDetector on the part of meetings that have been manually annotated as extract-worthy. This is because we want to focus on analyzing the impacts of the automatic DA features on the task of decision detection, rather than on that of extractive summarization.

Please note that because some of the test meetings we used in previous experiments are used as development set in [Dielmann and Renals, 2007b], the results reported here are obtained with a set of 50 meetings slightly different those used in previous experiments. Therefore a cross-table comparison of these results should be avoided.

Results in Table 10 show that our strategy that groups 15 DA classes into five major classes is beneficial to the models on the task of decision detection. On the task of detecting decision-related DAs from extractive summaries, it improves the recall of predicting decision-related topic segments by 16%. Although replacing the manual 5-class DA features with the automatically generated version degrades the overall accuracy, the model trained with the 5 automatically predicted DA classes (AUTO-5DA) still compares favorably with that trained with the 15 manually annotated DA classes (MANUAL-15DA).

However, when our system is operated on entire transcripts instead of extractive summaries, the advantage of the grouping strategy (from MANUAL 15-DA to MANUAL-5DA) does not extist. Neither is there any significant difference between the performance of MANUAL 5-DA and AUTO-5DA. As in Experiment 2, we have observed that DA-

based featues are less predictive when predicting on entire transcripts. One possible explanation is that DA-based features in general are not good at the dual task of disamibiguating extract-worthy DAs and decision-related ones simultaneously.

2.8 Conclusions and Future Work

In this paper, we present AMI DecisionDetector, a system which performs automatic decision detection in meeting speech and provides visual aids for users who wish to review decisions. We have examined how our computational models perform when detecting decisionsfrom the extractive summaries. To avoid the costly requirement of operating on extractive summaries, we have also examined how our computational models perform when detecting decisions from complete meeting transcripts. The models on the task of predicting decision-related discussions are evaluated at two levels of granularity: dialogue acts and topic segments. To further overcome the problem of imbalanced class distribution (i.e., only 2% are positive cases in complete transcripts), we have leveraged a variety of knowledge sources (e.g., words, prosody, DA-based contexts, topic annotations). The framework we applied here can also be used to recover information for other aspects in the argumentation process, such as problems and action items.

The results suggest that including knowledge sources beyond words greatly improves both the precision and the recall of models on the task of recognizing decision-related DAs from extractive summaries. However, for the task of recognizing decisions directly from entire transcripts, these additional knowledge sources tend to degrade the recall of the decision detection models and in turn their overall accuracy. As a result, even though the model that combines all the available knowledge sources performs substantially better in terms of precision, achieving 92% and 68% on the task of detecting decision-related topic segments in the training set and test set respectively, it still yields worse results in terms of overall accuracy.

We have also provided a quantitative account of the merits of different feature classes on both the task of detecting from extractive summaries and that of detecting from entire transcripts. Some of the findings are consistent in the two tasks. For example, among features that are extracted from the widely ranging knowledge sources, lexical features are the most indispensable. Also, DA-based features can improve the precision of models but degrade the recall.

However, there are also other findings that no longer hold true when our system is operated on complete transcripts instead of on a selective set of dialogue acts. For example, topical features have been shown to exhibit a distinctive advantage for locating decision topic segments from extractive summaries; However, this is not the case when identifying decision points in entire transcripts. In addition, when operated on entire transcripts, the model trained with lexical features alone outperforms the combined model in its recall rate. This is possibly because when attempting to detect decisions from the whole transcripts, the system needs to simultaneously disambiguate the extract-worthy and decision-related dialogue acts. Therefore, features that are good at disambiguating both will stand out, and features that fail in the extract-worthy DA detection task will be shown as weak features to the final performance of decision-related DA detection.

Another drawback of our previous approach is that many of the features used in this study

require human intervention, such as manual transcriptions, annotated DA segmentations and labels, and other types of meeting-specific features (e.g., speaker role). However, these semi-automatic and manual features are not always available. Therefore, in this work we tested whether our system is robust to the noise introduced by the automatically generated versions of these features. An exploratory study has shown that the performance of our approach does not degrade considerably after replacing the reference words with the ASR words, despite word recognition errors. Our further investigation on the impacts of using an automatically generated version of the DA class features (as reported in [Dielmann and Renals, 2007b]) shows that it is possible to include these automatic features in the model directly. It will not degrade the performance more than including the manually annotated 15-class DA features in the first place.

Also, our approach which automatically extracts decision-related DAs as summaries has some liabilities. First, the unconnected DAs in the extract result in semantic gaps that require contextualization to bridge. Second, anaphora and unexpected topic shifts between these extracted DAs also require context to resolve. Previously, we have attempted to provide such contexts by indicating the topic of the current discussion. However, a preliminary study has shown that the segment boundaries of decision-related discussions coincide with that of the topic segments less than 50% of the time. Last but not least, although it is our intuition that the decision-related DA extracts will assist users in finding and absorbing information in the meeting archives more efficiently and effectively, this assumption has yet to be tested with human subjects.

Therefore, in our future work, we have planned the following to address the shortcomings of our previous approach. We are now planning to conduct an extrinsic decision audit task-based evaluation on the utility of displaying decision-related DA information (as exemplified in Figure 2) to the users. We have also annotated decision-related discussion segmentation, which can be used to train computational models to find contexts that are needed for the interpretation of the identified decision points. Moreover, as we would like to disambiguate which sentence in the abstractive decision summary of a meeting is the most relevant to each of the identified decision points, the decision discussion segmentation annotations can also form a foundation for the development of the disambiguation model.

There are still some other problems we will not address in this research though. For example, as suggested by the mixed results obtained by the model that is trained without the DA-based features, the two-phase decision annotation procedure (as described in Section 2.3.2) may have caused annotators to select dialogue acts that serve different functional roles in a decision-making process in the set of decision-related DAs. One example is given in the dialogue demonstrated in Figure 2 (see Figure 3 for a more detailed view): The annotators have marked dialogue act (1), (5), (8), and (11) as the decision-related DAs related to this decision: *"There will be no feature to help find the remote when it is misplaced"*. Among the four decision-related DAs, (1) describes the topic of what this decision is about; (5) and (8) describe the arguments that support the decision-making process; (11) indicates the level of agreement or disagreement for this decision. Yet these decision-related DAs which play different functional roles in the decision-related process may each have their own characteristic features. Training one model to recognize decision-related DAs of all functional roles may have degraded the performance on the

(1) A: but um the feature that we considered for it not getting lost.
(2) B: Right. Well
(3) B: were talking about that a little bit
(4) B: when we got that email
(5) B: and we think that each of these are so distinctive, that it it's not just like another piece of technology around your house.
(6) B: It's gonna be somewhere that it can be seen.
(7) A: Mm-hmm.
(8) B: So we're we're not thinking that it's gonna be as critical to have the loss
(9) D: But if it's like under covers or like in a couch you still can't see it.
...
(10) A: Okay , that's a fair evaluation.
(11) A: Um we so we do we've decided not to worry about that for now.

Figure 3: Example decision-making discussion

classification tasks. Although it is out of scope of this research, we expect that developing models for detecting decision-related DAs that play different functional roles requires a larger scale study to discover the anatomy of argumentative discussions in general.

3 Subjectivity Recognition

subjective content in meeting data. We also present new techniques that we have developed for general subjectivity and sentiment classification, which in we will be applying to the classification of subjectivity in meeting data over the next few months.

Subjective content ranges from individual statements of opinions to entire documents or speeches that present a viewpoint or evaluation. In the context of meetings, recognising subjective content involves not only identifying when something subjective is being said, but also determining the type of subjective content (e.g., a positive sentiment), what the subjectivity is about, and possibly who is the *source* of the subjectivity (e.g., the speaker or someone else being quoted).

In the past few years, there has been a limited amount of work on recognising subjective content in multiparty conversations. [Wrede and Shriberg, 2003b] worked on recognizing meeting hotspots, which are a fairly coarse type of subjective content. [Hillard et al., 2003b], [Galley et al., 2004], and [Hahn et al., 2006] have worked on recognizing agreements and disagreements in meetings. Most recently, [Somasundaran et al., 2007] worked to recognize utterances that express sentiment and arguing, and [Neiberg et al., 2006] investigated the classification of positive, negative and neutral emotions in meetings.

In textual discourse, on the other hand, there has been a surge of research in the recognition of subjective content. Annotation schemes have been proposed for marking opinions and other types of subjective content, and corpora with detailed annotation of subjective content have been produced (e.g., [Wiebe et al., 2005]). Researchers have worked on automatically identifying subjective sentences (e.g., [Wiebe et al., 1999], [Riloff and Wiebe, 2003], and [Yu and Hatzivassiloglou, 2003]), recognizing the sentiment of phrases or sentences (e.g., [Morinaga et al., 2002], [Yu and Hatzivassiloglou, 2003], [Hu and Liu, 2004], [Popescu and Etzioni, 2005], and [Wilson et al., 2005b]), recognizing expressions of opinions in context (e.g., [Choi et al., 2006] and [Breck et al., 2007]), and identifying who an opinion is attributed to (e.g., [Bethard et al., 2004], [Kim and Hovy, 2004], and [Choi et al., 2005]). Other researchers have worked on identifying subjective documents (e.g., [Wiebe et al., 2004, Yu and Hatzivassiloglou, 2003]) and whether documents such as reviews are positive or negative (e.g., [Pang et al., 2002, Turney, 2002]) There has also been a great deal of focus on automatically acquiring *a priori* subjective information about words and phrases, information which then is applied to automatically recognizing subjective content. This research includes learning words and phrases that are indicative of subjective language (e.g., [Wiebe, 2000], [Riloff et al., 2003], [Kim and Hovy, 2005], [Esuli and Sebastiani, 2006]) as well as learning the polarity (semantic orientation) of words and phrases (e.g., [Hatzivassiloglou and McKeown, 1997], [Turney and Littman, 2003], [Esuli and Sebastiani, 2005], and [Takamura et al., 2005]).

In Section 3.1 we present our initial experiments in recognising subjective content in the AMI meeting data. Sections 3.2 and 3.3 present the novel techniques that we have developed for classifying document sentiment and for recognising subjective sentences.

Subjective Utterances
positive subjective
negative subjective
positive and negative subjective
uncertainty
other subjective
subjective fragment
Objective Polar Utterances
positive objective
negative objective
Subjective Questions
positive subjective question
negative subjective question
general subjective question

Table 11: AMIDA Subjectivity Annotation Types

3.1 Classifying Subjective Utterances

Monolingual text and multiparty conversation are very different types of discourse. However, given the depth of research and readily available resources for recognising subjective content in text, exploring the recognition of subjective content in conversation using approaches that work well for text is an obvious first track to pursue. Thus, for these initial experiments in classifying the subjectivity of utterances, we have two goals. The first is to evaluate how well existing subjectivity classifiers trained on text perform on conversational data. Our second goal is to establish a baseline for the performance we can expect from classifiers trained on annotated meeting data. These experiments will give us insights into the challenges of recognising subjective content in multiparty conversation, and suggest directions to pursue in our ongoing research.

3.1.1 Data

A total of 20 AMI scenario meetings (5 series of 4 meetings) were annotated with the AMIDA subjectivity annotation scheme, described in detail in the AMIDA State-of-the-Art Report: *Recognizing Subjective Content in Text and Conversation*. There are three main categories of annotations in the AMIDA scheme: *subjective utterances*, *objective polar utterances*, and *subjective questions*. Table 3.1.1 lists the annotation types in each category.

Although the annotators choose what spans in the transcript to mark for their subjectivity annotations, for the experiments in this section we use dialogue act segments as the unit of classification. The spans for the subjectivity annotations do not necessarily correspond to dialogue act segments. However, in an annotation study, we found that the subjectivity annotations cross segment boundaries relatively infrequently. This plus the fact that dialogue act segments can be identified automatically makes dialogue act segments a rea-

sonable choice for the unit of classification. Segment-level intercoder agreement is 0.56 kappa (79%) for marking subjective segments, 0.58 kappa (84%) for marking positive-subjective segments, and 0.62 (92%) for marking negative-subjective segments.

For the experiments, the 20 annotated meetings were divided into development and test sets of 10 meetings each. The develoment set contains the first meeting of each meeting series plus one other meeting from each series. The remaining two meetings from each series then go into the test set. The test set is divided into 5 folds, with each fold containing two meetings from the same series.

3.1.2 Subjectivity Classifiers

The existing subjectivity classifiers (developed for text) that we evaluate come from OpinionFinder [Wilson et al., 2005a]. OpinionFinder is a suite of classifiers for identifying subjective sentences and various types of private state expressions. The two subjective sentence classifiers in OpinionFinder are from [Wiebe and Riloff, 2005]. The higher precision classifier (HP-subj) looks for words and phrases from a lexicon to identify subjective sentences with high precision but low recall. The higher accuracy classifier (HA-subj) also uses information from the lexicon, but combines this knowledge with other features to obtain a higher accuracy classifier that is more evenly balanced between precision and recall. HA-subj is a naive Bayes classifier that was trained on a large, automatically constructed training set of subjective and objective sentences. The subjective sentences in this set were identified from a large pool of unannotated data by the HP-subj classifier; the objective sentences were similarly identified using a high-precision, objective sentence classifier. When evaluated on the MPQA Opinion Corpus [Wiebe et al., 2005], a corpus of new articles annotated for opinions and attributions, HA-subj has a 74% accuracy, with a 78.4% precision and a 73.2% recall, and HP-subj has a 91.7% precision with a recall of only 30.9%.

For training classifiers on the annotated meeting data, we use BoosTexter MH [Schapire and Singer, 2000] with 1000 rounds of boosting. We chose boosting for these intial experiments because it has been applied with success to recognising various types of subjective content in previous work. The first classifier (BAG) is a bag-of-words classifier that uses only the words in each segment as features. The second baseline classifier (BAG+LEX), uses bag-of-words features as well as two features defined based on the entries in OpinionFinder's lexicon. Every word in OpinionFinder's lexicon is tagged according to its reliability as a subjectivity clue: strongly subjective or weakly subjective. For the BAG+LEX, the two additional features represent the count of strongly subjective and weakly subjective clues from the lexicon that appear in a segment. The third baseline classifier (BAG+POL) again uses bag-of-words features, but in addition it uses a feature that represents the count of *in context* positive and negative words identified by OpinionFinder's polarity classifier. OpinionFinder's polarity classifier is a modified version of the classifier from [Wilson et al., 2005b]. It identifies when words from OpinionFinder's lexicon are actually being used to express a positive or negative sentiment in context. For BAG+LEX and BAG+POL, we excluded the words *okay*, *yeah*, and *yes* from OpinionFinder's lexicon because the distribution and use of these terms in speech as compared to text is extremely different.

		Subjective			Not Subjective		
	Acc	Rec	Prec	F	Rec	Prec	F
HP-subj	65.8	14.8	66.7	24.2	95.7	65.6	77.8
HA-subj	64.4	40.0	51.9	45.2	78.0	68.9	73.2
BAG	70.1	44.6	64.0	52.6	85.2	72.2	78.1
BAG+LEX	70.4	44.9	64.4	52.9	85.4	72.3	78.3
BAG+POL	70.4	45.2	64.6	53.2	85.4	72.4	78.4

Table 12: Initial Subjectivity Classification Results

3.1.3 Results

Table 3.1.3 gives the results for HP-subj, HA-subj, and the three classifiers trained on the annotated data. The first column in the table gives the overall accuracy of the classifier followed by subjective segment recall, precision and F-measure, and not-subjective segment recall, precision and F-measure. The results listed in the table are averages over the five folds in the test set.

Although all the classifiers outperform a simple classifier that always chooses the most frequent class (not-subjective, accuracy=62.9), none of the classifiers perform particularly well, indicating the very challenging nature of the task. For identifying subjective segments, the highest precisions are in the 60s and highest recalls are around 45. Nevertheless, even these initial results are informative. Perhaps unsurprisingly, the classifiers trained on the AMI data using very basic features outperform the classifiers that were trained on text. Ideally we would like to exploit the existing data and resources developed for text as we tackle the problem of recognising subjectivity in multiparty conversation. However, if even the most simple classifiers trained on the annotated meeting outperform existing trained-on-text classifiers and if existing subjectivity lexicons provide little improvement, we will need to think carefully about what, if any resources from subjectivity recognition in text it will be possible to exploit. At the least, classifiers will need to be retrained. It also raises the question of whether the types of features used to identify subjective content in text are even appropriate for recognising subjective content in conversation. One hope that we had for the HP-subj classifier was that it would be possible to use this classifier to automatically build up a collection of training data from unannotated meeting data in the same way it was used to build up the collection of training data used to train HA-subj. Although HP-subj is able to identify subjective sentences with high precision in text, the same is not true when it is applied to meeting data. Out of the classifiers in Table 3.1.3, HP-subj does has the highest precision for identifying subjective segments. However, a precision of 66.7 is still not very high, and it is not high enough to be used to automatically build up training data for a supervised algorithm to learn from. Finally, the classifier with the highest performance, if only by a slim margin, is the BAG+POL classifier. Recall that this classifier uses the output of OpinionFinder's word-level polarity classifier as one of its features. This suggests that methods for identifying in-context subjective words and phrases in conversation may be worthwhile to pursue.

In addition to subjective/not-subjective segment classification, we also trained our classifiers to classify segments as positive subjective or not-positive subjective, as well as

negative subjective or not-negative subjective. On average in the data, only 24.6% of segments are positive subjective and only 9.7% are negative subjective. Given these skewed distributions, the positive and negative subjective classifiers predictibly performed worse than the more general subjectivity classifiers. The best positive-subjective baseline classifier has a positive-subjective F-measure of 36.7. The best negative-subjective baseline has a negative-subjective F-measure of only 27.0

3.2 Interpolated Information Diffusion Kernels for Global Sentiment Classification

In this section we describe our experiments applying *information diffusion kernels* to global sentiment classification[7]. Information diffusion kernels provide similarity metrics in non-Euclidean information spaces, and for document classification they have been found to produce state-of-the-art results. The goal of global sentiment classification is to classify entire documents or large segments of text or speech as being, for example, positive, negative or neutral on average. We use textual documents for the data in our experiments, but our approach is easily applicable to other types of data as well, such as meeting topic segments or even individual utterances.

In our experiments, we compare information diffusion kernels with the more standard radial basis function (RBF) kernels, and in doing so, we also address the question of how best to represent texts containing sentiment: as binary vectors or as term frequency vectors. We also investigate which types of linguistic information are most useful for sentiment classification, focusing in particular on the role of unigrams and bigrams. Our results show that in fact an interpolation of unigram and bigram information is beneficial.

3.2.1 Data

The data we use in our experiments is the polarity dataset [Pang and Lee, 2004]. The polarity dataset is a collection of 1000 positive and negative movie reviews that has been widely used for research in global sentiment classification.

3.2.2 Information Diffusion Kernels

Our classification framework consists of *support vector machines* (SVM) [Boser et al., 1992] classifiers that linearly separate data by (implicitly) projecting it into a high-dimensional space using *kernels*: similarity functions that compare data represented by feature vectors. Information diffusion kernels ([Lafferty and Lebanon, 2005]) are similarity functions that explicitly take into account geometric properties of data like curvature and angle. They operate in curved L_1-normalized information spaces with only a local Euclidian structure, and they measure distance between datapoints along arcs connecting points. This contrasts with the standard vector space model of [Salton et al., 1975], where the distance between points is a Euclidean distance in high-dimensional spaces, irrespective of the geometry of these spaces. L_1-normalized information spaces for textual objects emerge

[7] This work was first reported in [Raaijmakers, 2007a].

by simply normalizing word frequences in documents represented by word sequences $w_1, \ldots, w_n$:

$$L_1(w_1, \ldots, w_n) = \frac{\mid w_1 \mid}{\sum_i^n \mid w_i \mid}, \ldots, \frac{\mid w_n \mid}{\sum_i^n \mid w_i \mid} \tag{1}$$

It is well-known that applying L_1-normalization corresponds to embedding data into the multinomial manifold: an infinitely differentiable manifold that corresponds to the parameter space of the multinomial distribution. An infinitely differentiable manifold is called a *Riemannian manifold* when equipped with a distance metric measuring the distance between two arbitrary points. Riemannian manifolds are generalizations of Euclidean 2D-geometry to arbitrary dimensions.

[Lafferty and Lebanon, 2005] and [Lebanon, 2006] propose the following multinomial information diffusion kernel (x, y two feature vectors) and show that it produces state-of-the-art results for document classification:

$$K_t^{ID}(x, y) = (4\pi t)^{-\frac{n}{2}} \exp\left(-\tfrac{1}{t} \arccos^2\left(\sum_{i=1}^{n} \sqrt{x_i y_i} \right) \right) \tag{2}$$

This is a one-parameter kernel, n being the dimension of the data. In the context of support vector machines, for sufficiently small $t \in [0, \epsilon)$, this kernel is positive definite, guaranteeing a unique solution to the convex problem the kernel machine has to solve [Lafferty and Lebanon, 2005]. Interestingly, the so-called Bhattacharyya distance occurs as a subterm in (2):

$$B(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)} \tag{3}$$

with p, q being two probability distributions over the same event space X. The Bhattacharyya distance is basically Euclidean distance restricted to probability distributions. In kernel (2), the inverse cosine (arccos) is used to measure distance across curves, and the Bhattacharyya kernel introduces a local notion of Euclidean distance.

In [Raaijmakers, 2007b] we proposed a two-parameter variant of (2):

$$K_{n,t}^{ID2}(x, y) = (4\pi t)^{\frac{n}{2}} \exp\left(-\tfrac{1}{t} \arccos^2\left(\sum_{i=1}^{m} \sqrt{x_i y_i} \right) \right) \tag{4}$$

Here, n is a free parameter, part of a positive exponent, and no longer bound to the dimension of the data; the constant m is the original dimension of the data vectors. Kernel K_t^{ID} arises as a special case of $K_{n,t}^{ID2}$, by setting n to $-m$. In the experiments reported in [Raaijmakers, 2007b] we observed clear positive effects on accuracy by adding n as a separate parameter and applied this kernel to a number of language learning tasks: classification problems on data not necessarily containing document-like structure (e.g., part-of-speech tagging and morphological analysis). In the work reported here, we will use a variant of kernel (4), to be described below.

3.2.3 Combining Linguistic Information for Sentiment Mining

Many types of information have been identified as relevant for sentiment classification during the last years: sentiment-bearing unigrams (e.g., [Turney, 2002]), word n-grams

(e.g., [Pang et al., 2002] and [Kennedy and Inkpen, 2006]), valency information of adjectives (e.g., [Ng et al., 2006]), valency-shifting properties of adverbs (e.g., [Andreevskaia et al., 2007] and [Kennedy and Inkpen, 2006]), dependency relations (e.g., [Ng et al., 2006] and [Wilson et al., 2005b]). [Pang and Lee, 2004] also found that filtering out objective sentences is also helpful.

In this section, we concentrate on using unigram and bigram information only, represented by frequency counts. These information sources are the natural ingredients for information diffusion kernels, once properly represented as normalized frequencies. RBF kernels, on the other hand, are frequently applied to either index vectors (where every feature flags the presence (yes/no) of a certain unigram of bigram), raw frequencies, or term weighting representations such as *tf.idf*.

Normalization of unigram and bigram frequencies will produce two separate multinomial probability distributions that need to be reconciled during the classification process. There are two options for normalization. The first is to view all unigram and bigrams strings as coming from the same distribution, ignoring intrinsic differences between the two distributions such as variance. Notice that during normalization the bigram sparseness may affect the unigram statistics. The second option is to interpolate the two information sources by assigning weights (Lagrange multipliers) to both unigram and bigram contributions, estimating these weights with an estimation procedure. Our hypothesis is that a good balance between these information sources will lead to accuracy gains. Below we describe a general hyperparameter estimation procedure, needed independently to optimize our sentiment classifiers, and we demonstrate how to use this algorithm to interpolate unigram and bigram information as part of this optimization process.

Hyperparameter Estimation Classifiers are complex, parameterized decision functions, and the parameters that influence the learning process are called hyperparameters. For SVMs, hyperparameters include the choice of kernel function, the regularization parameter C, and kernel parameters such as the degree of a polynomial kernel. Hyperparameters need to be accurately estimated; the wrong settings can greatly influence results.

In [Raaijmakers, 2007c] we proposed an elitist version of the well-known *cross-entropy (CE) method* for optimization. The CE algorithm is an iterative optimization procedure that performs parametrized sampling of a search space. It is particularly useful for sampling rare events: events that occur with low frequency, but which are useful from an objective point of view. Once the parameter vector along which the CE algorithm performs sampling is isomorphic to the type of solution to the optimization problem, the CE algorithm effectively becomes a search algorithm in a rare event space.

In [Raaijmakers, 2007c] it was demonstrated that, using the notion of elitism from the field of genetic algorithms, it is possible to define a cross-entropy-style search algorithm for hyperparameter optimization of classifiers. This algorithm adapts, at a certain time tick t, a hyperparameter j in a hyperparameter vector $\hat{v}$ with the following update formula.

$$\hat{v}_{t,j} = \frac{\sum_{i=1}^{n} I_{\{S(X_i^t) \geq \gamma^t\}} W(X_i^t; E^t) X_{ij}^t}{\sum_{i=1}^{n} I_{\{S(X_i^t) \geq \gamma^t\}} W(X_i^t; E^t)} \tag{5}$$

Here, the $I_{\{S(X_i^t) \geq \gamma_t\}}$ term is a performance indicator function. Every X_i is a random solution drawn from the hyperparameter space, conditioned on the current hyperparameter vector

and a width parameter μ. Every X_{ij} corresponds to a particular hyperparameter value, restricted to lie in the interval determined by the width parameter and the current solution derived by the algorithm:

$$X_{ij} \in \{[\hat{v}_{t,j} * (1 - \mu), \hat{v}_{t,j} * (1 + \mu)]\} \tag{6}$$

That is, every hyperparameter value X_{ij} in the candidate vector X_i is at most μ away from $\hat{v}_{t,j}$. The performance function $I()$ in 5 measures whether a certain X_i performs at least as well as the best score γ obtained so far, as measured by an independent objective function (like classifier accuracy). The $W()$ term embodies a *change of measure*: a steering mechanism to steer the search process into a certain 'good' direction. In our case, we steer search into the direction of the best ('elitist') result encountered during the whole optimization process, $E^t = \text{argmax}_{\hat{v}_{i=1,\ldots,t}} \gamma^i$. We measure (normalized) Euclidean distance between this best result and the candidate solution, and use this information (which lies in the interval $[0, 1]$) to steer search:

$$W(X_i^t; E^t) = 1 - \frac{\sqrt{\sum_{j=1}^{m}(X_{ij}^t - E_j^t)^2}}{\sqrt{\sum_{j=1}^{m}(X_{ij}^t)^2}\sqrt{\sum_{j=1}^{m}(E_j^t)^2}} \tag{7}$$

Kernel interpolation The elitist hyperparameter estimation algorithm can be used to determine the optimal weights for unigram and bigram contributions to sentiment classification, once we make these weights hyperparameters of a kernel-based classifier, combining unigram and bigram information in a parameterizable way. While there are many ways to combine this information in one kernel, the following composit kernel combines them simply through a weighted sum. We factor out the contributions of both unigrams and bigrams by two separate Bhattacharyya subkernels. The weights λ_1 and λ_2 are interpolation weights that express the relative importance of the two information sources.

$$K_{n,t,\lambda_1,\lambda_2}^{ID2}(x, y, i, j) = (4\pi t)^{\frac{n}{2}} \exp \left(-\frac{1}{t} \arccos^2 \left(\frac{\lambda_1 \left[\sum_{k=1}^{i} \sqrt{x_i y_i} \right] + \lambda_2 \left[\sum_{i+1}^{j} \sqrt{x_i y_i} \right]}{2} \right) \right) \tag{8}$$

Every movie review is indexed for unigrams and bigrams. The result is a sparse feature vector, consisting of separately normalized frequencies of unigrams and bigrams. The two subscript parameters i and j indicate the highest index positions of the unigrams and bigrams in the vocabulary. This information is used by the kernel to factorize the unigram and bigram parts: any feature with an index higher than the maximum unigram index will be considered a bigram feature. By treating the interpolation weights as hyperparameters, we can use the elitist hyperparameter estimation algorithm to estimate these weights jointly with the 'normal' kernel hyperparameters n and t, which seems only natural to do. The *hyperkernel* (8) was implemented as an extension to the LIBSVM support vector machine toolkit ([Chang and Lin, 2001]).

3.2.4 Experiments

We set out to investigate the following questions. First, should we use all unigrams and bigrams in the training data or a dedicated selection only? Second, should we interpolate unigram and bigram information or just combine these two information sources, pretending they arise from the same distribution? Third, should we use normalized frequency information or just mere 'binary' presence (yes/no) of a certain vocabulary term? Table 13 lists the set of experiments we performed to find answers to these questions. In order to apply the information diffusion kernels to the binary representations, we set all term frequencies to 1. We normalized these binary vectors by assigning every feature a value $1/n$, with n the number of terms from the vocabulary that were actually found in the review. For the interpolation experiments, we also investigated the effects of setting one of the weights λ_1 and λ_2 in kernel (8) to zero; setting $\lambda_1 = 0$ eliminates the unigram contribution, and $\lambda_2 = 0$ eliminates the bigram contribution.

Interpolation	
E1	all terms, binary features
E2	all terms, normalized frequencies
E3	selected terms, binary features
E4	selected terms, normalized frequencies
No interpolation	
E5	all terms, binary features
E6	all terms, normalized frequencies
E7	selected terms, binary features
E8	selected terms, normalized frequencies
RBF	
E9	all terms, binary features
E10	all terms, normalized frequencies
E11	selected terms, binary features
E12	selected terms, normalized frequencies

Table 13: Experimental conditions.

Term selection Following [Ng et al., 2006], we ranked all unigrams and bigrams t in the data based on their weighted log-likelihood ratio (WLLR) scores, for the two classes $c \in \{+1, -1\}$:

$$WLLR(t, c) = P(t \mid c) \ \log \frac{P(t \mid c)}{P(t \mid \neg c)} \tag{9}$$

For both classes, we selected the 5000 highest ranked unigrams and 5000 highest ranked bigrams from the training data as index vocabulary.

Class	unigram
–1	bad, worst, stupid, boring, ridiculous, awful
+1	great, best, well, perfect, wonderful

Class	bigram
–1	this mess, the worst, worst movie a stupid, is terrible, waste of
+1	the best, is excellent, most powerful very effective, a great

Table 14: Highly WLLR-ranked unigrams and bigrams across the two classes.

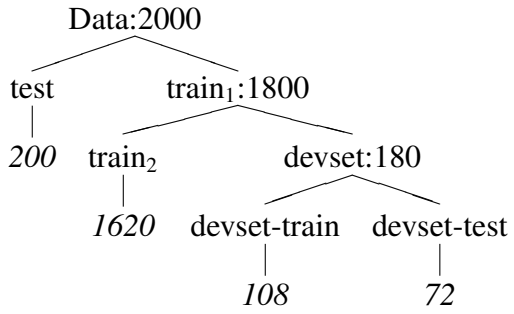


Figure 4: Sample data partitioning for 10CV. Actual training of the classifier took place on the 'train$_2$' partitions containing 1620 cases. Hyperparameter estimation took place on the 'devset-train' and 'devset-test' partitionings. The trained classifier was tested on the 'test' partitions containing 200 cases each.

Experimental setup For every experiment, we performed 10-fold cross-validation. We permuted the data, and split it into 10 training and test partitions. Every training partition T (90% of the data) was subsequently split into a secondary 90% training partition (90%× 90%=81% of the data), and a 10% development data partition. Finally, every development data partition was subsequently split into a 60% development training partition and a 40% development test partition. The two development data partitionings were used by the hyperparameter estimation algorithm to derive the hyperparameters for the kernel machine, for each fold separately. After the hyperparameter estimation algorithm found these hyperparameters, we trained on the 81% training data partitionings, and tested on the corresponding 10% test partitioning. Figure 4 illustrates the partitioning of a particular fold. Notice that the development data is kept separate from the training data for each fold. Because of this our results are not strictly comparable to 10-fold cross-validation results previously reported for this data that do not employ development data.

We measured the significance of results using a paired t-test, at $p \leq 0.05$ and $p \leq .1$.

3.2.5 Results

Table 15 list the results for our various experiments. Results for the interpolated kernel first of all underline the importance of the unigram contribution. The hyperparameter

Experimental condition	Interpolation		
	$\lambda_1 \neq 0, \lambda_2 \neq 0$	$\lambda_1 = 0$	$\lambda_2 = 0$
E1	87.5	81.9	86.7
E2	87.3	79.75	86.2
E3	**88.0**	81.85	86.7
E4	86.4	80.5	84.0
	No interpolation		
E5			82.65
E6			80.5
E7			86.0
E8			86.8
	RBF		
E9			86.8
E10			84.25
E11			85.8
E12			81.1

Table 15: Generalization accuracy (10CV), for the various experimental conditions (see table 13 for explanation of the labels).

	E1	E3
E8	+ ($p \leq .1$)	+ ($p \leq .1$)
E9	+ ($p \leq .1$)	+ ($p \leq .1$)
E11	+ ($p \leq .05$)	+ ($p \leq .05$)

Table 16: Pairwise significance results, measured with a paired t-test.

estimation algorithm predominantly assigned higher weights to the unigram subkernel across the various folds. The importance of unigrams becomes clear when manually deactivating the unigram subkernel, that is, setting $\lambda_1 = 0$. The drops in accuracy are more dramatic than when eliminating the bigram contribution, sometimes even amounting to over 6%. This is in line with the findings of [Pang et al., 2002].

The combined result of using both unigram and bigram kernels, for the WLLR selected terms, with normalized binary vectors, produces the best results, showing that there is indeed useful, supplementary information in the bigrams. This classifier (E3) significantly outperformed the RBF baseline classifier (see Table 16). Normalized binary vectors with WLLR-selected terms perform better[8] than normalized frequency vectors.

[Ng et al., 2006] observed beneficiary effects of using the WLLR-ranked subset of unigrams and bigrams for a linear kernel with binary features. We observed that using all terms is beneficiary for the RBF kernel (classifier E9) only. For the interpolated kernel, the difference between WLLR selected terms and all terms vanishes: the difference between E1 and E3 is statistically significant only at $p \leq .35$, and the difference between E2 and E4 is significant only at $p \leq .42$.

Most importantly, our results specifically indicate that the combination of unigrams and bigrams yields better performance than using only one of these resources. Interpolation of unigrams and bigrams performs better than plain, unweighted combination. Further, RBF kernels are outperformed by information diffusion kernels, underlining the usability of these kernels.

3.2.6 Conclusions

In this section, we investigated the use of unigram and bigram information for sentiment classification of reviews. We found that a composite kernel consisting of a separate unigram and bigram kernel, which were interpolated automatically with a cross-entropy-style hyperparameter estimation procedure, produced the best results. These results rank among the best reported using unigrams and bigrams on the polarity dataset [Pang and Lee, 2004] and are still open to improvement, by incorporating other sources of useful information reported in the literature, such as valency information of adjectives. We intend to expand this work into the direction of incorporating these sources of information by properly embedding them into L_1-space, and weighting them with our hyperparameter estimation algorithm.

Being a general classification approach, kernel interpolation is amenable to multiclass classification of sentiment, and more generally subjectivity, and it should be useful for predicting more fine-grained sentiment and other types of subjective content as well. In subsequent work we will address the recognition of the subjectivity and sentiment of sentences in text and segments in multiparty conversation using this method.

[8] E3 is significantly better than E4 at $p \leq .13$.

3.3 A Shallow Approach to Subjectivity Classification

In this section we present a shallow linguistic approach to subjectivity classification. Using SVMs with information diffusion kernels, we show that a data representation based on counting character *n*-grams improves over results previously attained for sentence-level subjective sentence classification using deeper linguistic representations. Specifically, we compare two types of string-based representations: key substring groups and character *n*-grams. Although the experiments we report are on text, the results are very encouraging and suggest that this approach is a good one to pursue for subjectivity classification in conversation, where the deeper linguistic knowledge often used in subjectivity recognition (e.g., parse structure and dependency information) is just not available.

3.4 Shallow Linguistic Representations

Recent research in text mining has provided new evidence that shallow linguistic representations are able to capture important linguistic aspects of utterances, while being far easier to compute and presupposing less linguistic theory than deeper linguistic structure (e.g., syntactic or semantic information). For instance, [Giuliano et al., 2006] show that shallow linguistic features consisting of words, lemmas and orthographic equivalence classes (capitalized words, numerals, etc.) are quite useful for relation extraction in the biomedical domain, and outperform approaches that incorporate syntactic and semantic information. In [Stamatatos, 2006], character *n*-gram models are used to succesfully predict authorship, using ensemble classifiers. [Li and Roth, 2001] observe that shallow parsers provide for better performance and robustness when confronted with new and low quality texts than their 'deep linguistic' counterparts. [Kanaris and Stamatatos, 2007] proposes the use of character *n*-grams for webpage genre identification. In our experiments in subjective sentence classification, we experiment with two types of shallow linguistic representations, key substring groupings and character *n*-grams.

3.4.1 Key Substring Groupings

String-based feature representations often suffer from a high level of specificity, and finding a suitable level of abstraction from raw feature values that still captures important distributional properties of the underlying data is an area of active research in the machine learning community. [Zhang and Lee, 2006] recently proposed a *key substring group* representation for document classification that significantly outperforms approaches based on deep linguistic analysis. Given a text corpus, a suffix tree is formed that compresses the entire corpus into a tree labelled with set-valued nodes. These nodes are containers for substrings that have exactly the same distribution in the given corpus: a key substring group is a set of substrings that share the same path label in the tree that stores them. By definition of the tree data structure, all strings in a key substring group have exactly the same frequency. Since these substrings are equivalent from a distributional point of view, they can be safely replaced by a single arbitrary symbol, which constitutes another form of value abstraction based on distributional string equivalence. Using the key substring group representation, [Zhang and Lee, 2006] report significant improvement over state-of-the-art results for both authorship classification and document topic classification.

3.4.2 Character *n*-grams

In statistical language modeling, language models assign probabilities to sequences of tokens $t_1 \dots t_N$ according to a product of local probabilities:

$$P(t_1 \dots t_N) = \prod_{i=1}^{N} P(t_i \mid t_1 \dots t_{i-1}) \tag{10}$$

which, in the case of an n-gram representation ($n = 1...N$) becomes

$$P(t_1 \dots t_N) = \prod_{i=1}^{N} P(t_i \mid t_{i-n+1} \dots t_{i-1}) \tag{11}$$

Maximum likelihood (ML) estimates for n-grams are determined by relative frequencies of cooccurrence:

$$r(t_i \mid t_{i-n+1} \dots t_{i-1}) = \frac{\mid t_{i-n+1} \dots t_i \mid}{\mid t_{i-n+1} \dots t_{i-1} \mid} \tag{12}$$

These ML estimates can be made much more accurate when the amount of data increases. In general, n-gram models converge to the ML estimate with enough data, under the usual smoothing schemes that reserve a small probability mass to assign non-zero probabilities to unseen events. Clearly, the more data these models see, the less probability mass is "wasted" on unseen events. Generating more data by using longer n-grams (5-grams, 6-grams, etc.) is not a realistic option, as this will lead to sparse event spaces: the chances of observing a certain n-gram decrease with increasing values of n. An easy way to generate more data is to exploit the subword level and use character n-grams. For instance, for the following sentence

This car really rocks. (13)

subword character bigrams and trigrams are

- `th`, `hi`, `is`, `ca`, `ar`, `re`, `ea`, `al`, `ll`, `ly` , `ro`, `oc`, `ck`, `ks`, `thi`, `his`, `car`, `rea`, `eal`, `all`, `lly`, `roc`, `ock`, `cks`.

Any n-character word produces $n - 1$ character bigrams and $n - 2$ character trigrams. This means that for any document D containing w words with average length l, the rough expansion factor is $w \cdot (l - 1) + w \cdot (l - 2) = 2wl - 3w$ for character bigrams and trigrams.

In discriminative (non-generative) models of machine learning, n-gram information usually is used in the form of a *bag of words*, or, better put, a bag of n-grams. This model assumes a feature space consists of a set of frequency counts and treats every feature in this feature space as an element of a multiset. The bag $\{a, b, a, b, b\}$ can be interpreted as a frequency table $a : 2, b : 3$, and is open to more advanced counting methods, such as TF-IDF (see, e.g., [Joachims, 1998]).

Intuitively, more items in the bag make the bag more informative, as it makes for a more descriptive event space and presumably a better model. So, more data might be benificiary for discriminative models as well. Yet, this is an intricate issue, as words that are not strongly predicative of a certain class might generate noise, leading to conflation problems with other classes. This is the reason why in many applications (e.g. [Ng et al., 2006]

and [Raaijmakers, 2007a]), term selection is applied in order to create a strongly class-predictive index vocabulary.

The bag-of-words approach ignores sequentiality altogether. Nonetheless, it is clear that there is useful information in the sequential structure of documents. For sentiment mining, specific combinations of terms appear particularly informative, most notably valence shifting combinations like 'not good.' A simple way of restoring sequentiality for a character-based bag-of-words approach is to employ *n*-grams on the subword level that cross word boundaries and therefore reach the superword level. For instance, a bigram and trigram representation for sentence (13) that ignores word boundaries produces

- th, hi, is, s<sp>, <sp>c, ca, ar, r<sp>, <sp>r, re, ea, al, ll, ly, y<sp>, <sp>r, ro, oc, ck, ks, thi, his, is<sp>, s<sp>c, <sp>ca, car, ar<sp>, r<sp>r, <sp>re, rea, eal, all, lly, ly<sp>, y<sp>r, <sp>ro, roc, ock, cks

with <sp> a whitespace indicator. These *n*-grams capture transitions between consecutive words, and thus encode phrasal effects on the character level. Notice that the amount of string data increases significantly (39 vs. 24 character *n*-grams). For w words, the expansion factor for bigram and trigram superword character *n*-grams is $2(w-1)+3(w-1) = 5w-5$ extra strings. We shall refer to these *n*-grams as *superword character n-grams* (supergrams, for short) and to word-internal character *n*-grams as *subword character n-grams* (or subgrams). As character *n*-grams do not encode positional information, value abstraction arises naturally from overlap of *n*-grams;

3.5 Data and Experiments

For the experiments in this section, we use the MPQA Opinion Corpus [Wiebe et al., 2005]. The MPQA Corpus is a collection of news articles that have been annotated for opinions and attributions. Although the annotations are of words and phrases, they can be used to derive sentence-level subjectivity annotations in a straightforward manner (see [Riloff et al., 2006]). We use 9,266 sentences from the MPQA corpus (54% subjective) divided into 3 folds for cross validation, following the exact splits used by [Riloff et al., 2006].

In our experiments, we investigate the following substring-based representations:

- Subword character *n*-grams (bi-, tri- and quadrigrams)
- Superword character *n*-grams (bi-, tri- and quadrigrams)
- Key substring groups
- A mixture of key substring groups and superword character *n*-grams (bi-, tri- and quadrigrams)

The word-internal character *n*-grams are used as a baseline. The key substring group features were generated with standard parameter settings of the software made available by

[Zhang, 2006a]. Due to an inherent memory restriction of this software, we had to (uniformly) limit training set size over all runs and data representations to 5,000 datapoints, which amounts to 81% of the original training data.

Having at our disposition essentially distributional, frequency-based string data, we chose to use SVM classifiers with information diffusion kernels (*multinomial kernels*) (see Section 3.2.2). Specifically, we use a simple, hyperparameter-free multinomial kernel: the negative geodesic kernel NGD [Zhang et al., 2005].

$$K_{NGD}(x, y) = -2 \arccos\left(\sum_{i=1}^{n} \sqrt{x_i y_i}\right) \tag{14}$$

Notice that this kernel combines a *local*, Euclidean notion of similarity with a geodesic notion of similarity: the vector product expresses cosine similarity, and the inverse cosine the measurement of distance along a curve. The multinomial manifold $\mathbb{P}^n$ is isometric to the positive portion of the n-sphere with radius 2, $\mathbb{S}^n_+$ [Kass, 1989, Lebanon, 2003]:

$$\mathbb{S}^n_+ = \{\phi \in \mathbb{R}^{n+1} : \| \phi \| = 2, \forall i, \phi_i \geq 0\} \tag{15}$$

by a diffeomorphism $F : \mathbb{P}^n \mapsto \mathbb{S}^n_+$:

$$F(x) = (2\sqrt{x_i}, \ldots, 2\sqrt{x_{n+1}}) \tag{16}$$

This allows for measuring distance with a kernel K between two vectors x, y in the much compact space $\mathbb{S}^n_+$:

$$K(F(x), F(y)). \tag{17}$$

The length of the shortest path connecting these two points in hyperspace is a segment of a great circle.

3.6 Results

The results in Table 15 show first of all that superword character n-grams perform the best among the other representations. Even on the basis of using only 81% of the training data, this representation also leads to an improvement over the results reported on this data using unigram, bigram and extraction pattern features (74.9% reported by [Riloff et al., 2006]).

	SUB	SUPER	KSG	KSG + SUPER
Acc	74.57	**82.5**	77.9	81.9
Rec	77.6	**84.9**	80.7	84.4
Prec	75.9	**83.2**	78.9	82.5
F_1	76.7	**84.0**	79.8	83.5

Table 17: Average accuracy, recall, precision and F_1 (three-fold cross-validation) for subgrams, supergrams, key substring group features (KSG) and a combination of KSG and supergrams (best results in bold).

We graphically compare the four different data representations across the three cross-validation folds, using both cost curves and ROC curves. ROC curves illustrate the trade-off between selectivity and sensitivity, by plotting a curve of false positive rate versus true positive rate while varying a sensitivity or threshold parameter. For a discrete classifier (like, in our case, a SVM producing a binary decision), ROC analysis produces one point in ROC space for every test fold. A ROC curve can be drawn by connecting this point to the origin and the upper right corner. Cost curves [Drummond and Holte, 2006] allow for a much more articulate assessment of classifier quality. A cost curve can be seen as a generalization of ROC curves: a point in ROC space corresponds to a line in cost space. Cost curves indicate classifier performance at any operating point in the graph, expressed as the expected cost (or error rate). This makes it easier to compare classifiers than with ROC curves, which, when intersecting, do not easily illustrate when exactly one classifier overall outperforms another. Cost curves, applied to binary classification problems, indicate exactly when one classifier outperforms another by plotting the probability of observing the positive class as a function of error rate. Lower lines indicate better performance. Assuming that the cost of misclassification of both classes (subjective versus objective) is equal, we can read off the exact performance per classifier at the operating point $x = 0.5$: the values on the y-axis reduce to error rate [Drummond and Holte, 2004].

The curves in Figures 6 and 5 confirm the superior performance of superword character n-grams. The combination of key substring group features and superword character n-grams performs worse than superword character n-grams alone. The baseline consisting of word-internal character n-grams performs worst.

3.6.1 Bias and Variance Decomposition of Classification Error

The error of a classifier is often decomposed into bias, variance and noise [Breiman, 1996]. Bias is the systematic, intrinsic error of a classifier, variance is its data dependent error, and noise corresponds to errors (either in features or classes) in the data. Noise is often assumed to be zero [Kohavi and Wolpert, 1996] as reliably estimating noise is often infeasible for large datasets. In our analysis, we used the definition of bias and variance proposed by [Kohavi and Wolpert, 1996]:

$$bias_{KW_x^2} = \frac{1}{2} \sum_{y \in Y} [P_{Y,X}(Y = y \mid X = x) - P_{\mathcal{T}}(\mathcal{L}(\mathcal{T})(x) = y]^2 \tag{18}$$

$$variance_{KW_x} = \frac{1}{2} \left(1 - \sum_{y \in Y} P_{\mathcal{T}}(\mathcal{L}(\mathcal{T})(x) = y)^2 \right) \tag{19}$$

According to this definition, the bias of a classifier at a data point x is the squared difference between the true class observed for x in the training data X, Y (X a feature space and Y a class space), and the class predicted for x in the hypothesis space $\mathcal{T}$, i.e., the output of the classifier $\mathcal{L}$ trained on $\mathcal{T}$ and applied to x. [Kohavi and Wolpert, 1996] measure bias and variance on the basis of the following procedure: each data set is partioned into a training set d and a test set t. The training data d is partitioned into 50 training sets of size $2m$, where m is 100 for data sets less than 1,000 data points, and 250 otherwise. The bias and variance estimates are then derived from training the classifier on the 50 training

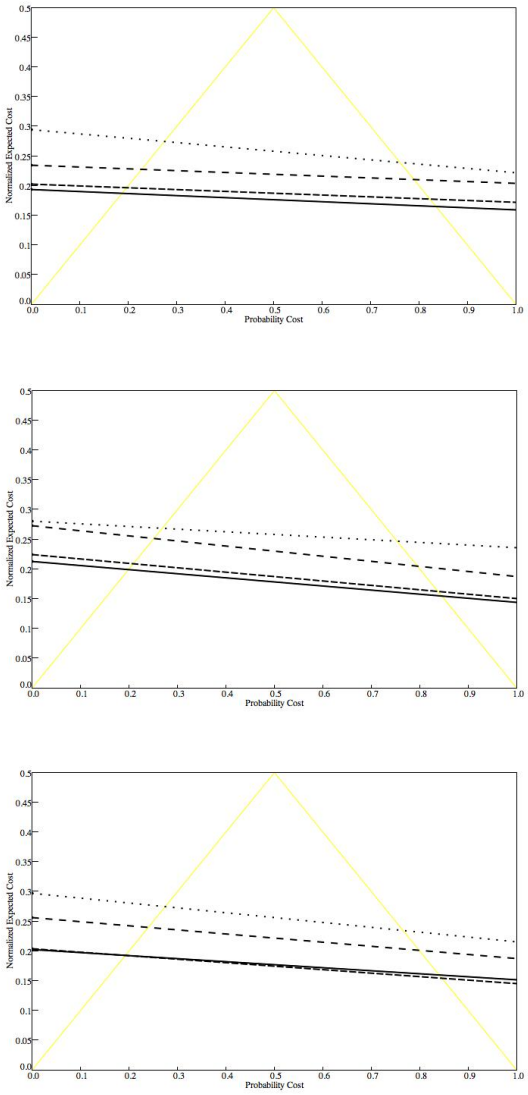


Figure 5: Cost curves for 3 folds, comparing superword character *n*-grams (solid line), subword character *n*-grams (dotted), KSG (long dashed), and KSG+superword character *n*-grams (short dashed).

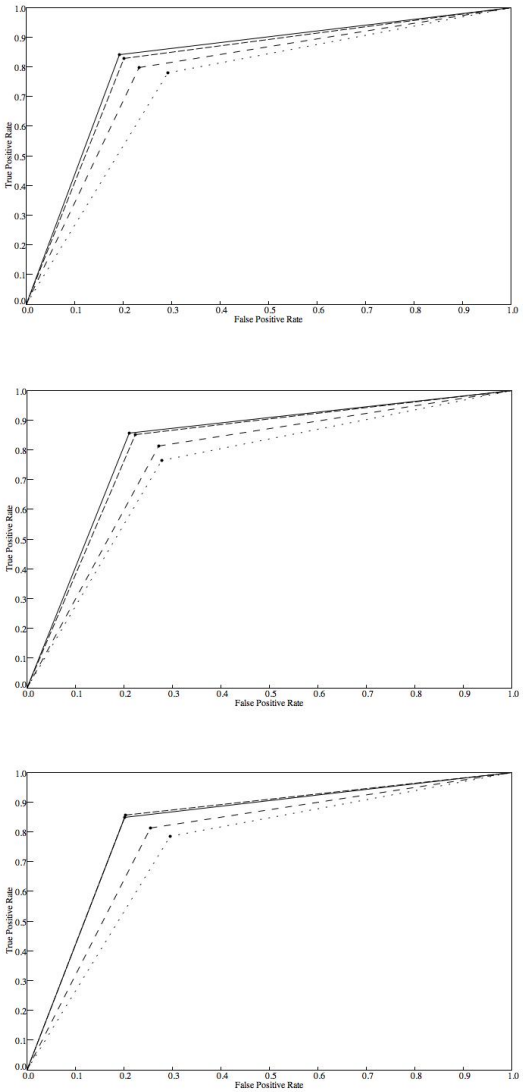


Figure 6: ROC curves for 3 folds, comparing superword character *n*-grams (solid line), subword character *n*-grams (dotted), KSG (long dashed), and KSG+superword character *n*-grams (short dashed).

subsets in turn, and applying it to the test data *t*. As has been noted by [Webb, 2000], this implementation of bias and variance suffers from the fact that training subsets become very small for medium to average size data sets, and the estimates for bias and variance consequently become unreliable. We took to heart the recommendations of [Webb, 2000], and applied 10 runs of 3-fold cross-validation, in order to get better estimates of bias and variance. From Table 18, we see that superword character *n*-grams yield lower bias than word-internal character *n*-grams and key substring groups. The combination of key substring groups and superword character *n*-grams has a slightly lower bias than superword character *n*-grams alone, but this small difference is hard to interpret, and, as noted, overall performance of this combination is lower than the performance of superword character *n*-grams alone.

SUB	SUPER	KSG	KSG + SUPER
39.8	35.9	37.9	35.8

Table 18: Bias decomposition of the classification error, using subgrams, supergrams, key substring group features (KSG) and a combination of KSG and supergrams (variance=100-bias).

3.7 Plans and Ongoing Work

This chapter presents our preliminary experiments in automatically recognising subjective content in meetings. In these experiments we explored only the most straightforward features and methods for classifying subjective utterances. Over the next few months, our continued work on subjectivity recognition will focus in two directions. First, we will explore the use of more sophisticated lexical features, as well as features capturing prosodic and visual information. We will evaluate the lexical features on the ASR as well as the reference transcripts, to evaluate how much performance degration is caused by ASR and how much the non-lexical features can compensate. Second, we will investigate how well the new approaches described in Sections 3.2 and 3.3 perform for classifying subjective utterances in meeting data. In addition to exploring shallow character *n*-grams features, which depend on the ASR transcription, we will also explore the utility of phoneme *n*-grams for subjectivity recognition.

4 Dialogue Acts

4.1 Introduction

The concept of dialogue acts (DAs) is based on the speech acts described in [Austin, 1962] and [Searle, 1969]. The idea is that speaking is acting on several levels, from the mere production of sound, over the expression of propositional content to the expression of the speaker's intention and the desired influence on the listener. Dialogue acts are labels for utterances which roughly categorise the speaker's intention.

As such, they are useful for various purposes in a dialogue or meeting processing situation. DAs are used as elements in a structural model of a meeting. A simple example would be a browser which highlights all points where a suggestion or offer was recognised. Often, however, DA labels serve as elementary units to recognise higher levels of structure in a discourse. To generate abstractive summaries, for example, content is extracted from utterances, and integrated in a discourse memory depending on the DAs of the utterances.

The dialogue act recognition process consists of two subtasks: segmentation and classification (tagging). The first step is to subdivide the sequence of transcribed words in terms of DA segments. The goal is to segment the text into utterances that have the same (or at least approximately similar) temporal boundaries to the annotated DA units. The second step is to classify each segment as one of the DA classes from the adopted DA annotation scheme. These two steps may be performed either sequentially (segmentation followed by classification) or jointly (both tasks carried out simultaneously by an integrated system). Although most of the work on automatic DA processing has been focused on the tagging task, assuming knowledge of the reference DA segmentation; novel integrated DA recognition frameworks are growing in popularity.

4.1.1 The AMI & AMIDA Dialogue Act Tag Set

For AMIDA, the most comprehensive and suitable corpus resource is the AMI corpus that was build in the predecessor project AMI because it is based on very similar meeting situations. The AMI meeting corpus [Carletta et al., 2005a] is a multimodal collection of annotated meeting recordings. It consists of about 100 hours of meetings collected in three instrumented meeting rooms. About two thirds of the corpus consists of meetings elicited using a scenario in which four meeting participants, playing different roles on a team, take a product development project from beginning to completion. The scenario portion of the corpus consists of a number of meeting series, with four meeting per series. Each series of four meetings involves the same four participant roles, and comprises project kick-off, functional design, conceptual design, and detailed design meetings. The aim of the corpus collection was to obtain a multimodal record of the complete communicative interaction between the meeting participants. To this end, the meeting rooms were instrumented with a set of synchronised recording devices, including lapel and headset microphones for each participant, an 8-element circular microphone array, six video cameras (four close-up and two room-view), capture devices for the whiteboard and data projector, and digital pens to capture the handwritten notes of each participant. The cor-

pus has been manually annotated at several levels, including orthographic transcriptions, various linguistic phenomena including head and hand movements, and focus of attention[9]. Most of the scenario data in the AMI corpus, over 100,000 utterances, have been annotated for dialogue acts. The AMI dialogue act scheme[10] consists of 15 dialogue act types (table 19), which are organised in six major groups:

- Information exchange: giving and eliciting information
- Possible actions: making or eliciting suggestions or offers
- Commenting on the discussion: making or eliciting assessments and comments about understanding
- Social acts: expressing positive or negative feelings towards individuals or the group
- Other: a remainder class for utterances which convey an intention, but do not fit into the four previous categories
- Backchannel, Stall and Fragment: classes for utterances without content, which allow complete segmentation of the material

Group	Dialogue Act		Frequency	
Segmentation	fra	Fragment	14348	14.0%
	bck	Backchannel	11251	11.0%
	stl	Stall	6933	6.8%
Information	inf	Inform	28891	28.3%
	el.inf	Elicit Inform	3703	3.6%
Actions	sug	Suggest	8114	7.9%
	off	Offer	1288	1.3%
	el.sug	Elicit Offer or Suggestion	602	0.6%
Discussion	ass	Assessment	19020	18.6%
	und	Comment about Understanding	1931	1.9%
	el.ass	Elicit Assessment	1942	1.9%
	el.und	Elicit Comment about Understanding	169	0.2%
Social	be.pos	Be Positive	1936	1.9%
	be.neg	Be Negative	77	0.1%
Other	oth	Other	1993	2.0%
Total			102198	100.0%

Table 19: The AMI Dialogue act scheme, and the DA distribution in the annotated scenario meetings.

Each DA unit is assigned to a single class, corresponding to the speaker's intent for the utterance. The distribution of the DA classes, shown in table 19, is rather imbalanced,

[9]The annotated corpus is freely available from `http://corpus.amiproject.org`

[10]Guidelines for Dialogue Act and Addressee Annotation V1.0, Oct 13, 2005. `http://mmm.idiap.ch/private/ami/annotation/dialogue_acts_manual_1.0.pdf`

with over 60% of DAs corresponding to one of the three most frequent classes (inform, fragment or assess). Over half the DA classes account for less than 10% of the observed DAs. This annotation scheme is different to the one used for the ICSI corpus (section 4.1.3), thus it is not possible to test a DA recognition system developed on the AMI data on the ICSI corpus or vice-versa.

4.1.2 Training, development and test sets

Out of the 100 hours of meeting recordings in the AMI corpus, roughly 72 hours are 'scenario' meetings, concerned with the development of a remote control. The scenario meetings are organised in 35 series of (normally) four meetings, which have been split into designated training, development and evaluation sets.

25 series of meetings have been assigned to the training set, five to the development and five to the test set (table 20). For the purpose of cross-validation (CV), a split into ten parts was defined (table 21). This split can be used for ten-fold or five-fold CV. Note that for ten-fold CV, the size of the parts differ - each part contains either three or four series of meetings. Splitting a series of meetings over two parts might introduce unwanted speaker modelling effects, as a model learned on the first half of a series might be evaluated on the other half.

The assignment into training, dev and test data is independent of the annotations. In practice, most annotations are not available on all meetings, therefore systems use only those meetings which contain all required annotations. Also, the split definition does not apply to annotations which were only performed on a very small number of meetings. For systems requiring those annotations, individual training and test sets have to be defined.

Subset	Meetings	#meetings	#series
Training set	ES2002, ES2005-2010, ES2012-2016 IS1000-1007 TS3005 TS3008-3012	98	25
Development set	ES2003, ES2011, IS1008, TS3004, TS3006	20	5
Evaluation set	ES2004, ES2014, IS1009, TS3003, TS3007	20	5
All scenario data		138	35

Table 20: The split of the AMI scenario data into training, development and evaluation sets.

4.1.3 The ICSI Meeting Corpus and DA Tag Set

The ICSI Meeting Corpus [Janin et al., 2003] consists of 75 multi-party meetings recorded with multiple microphones: one head-mounted microphone per participant and four table-top microphones. Each meeting lasts about one hour and involves an average of six participants, resulting in about 72 hours of multichannel audio data. The corpus contains human-to-human interactions recorded from naturally occurring meetings. Moreover, having different meeting topics and meeting types, the data set is heterogeneous both in terms of content and structure.

Training	Dev	Test	Fold no.	
ES2002, IS1000	TS3004	ES2004	1	1
ES2007, IS1001, TS3005			2	
ES2005, IS1002	TS3006	ES2014	3	2
ES2006, IS1003, TS3008			4	
TS3009, IS1004	ES2003	IS1009	5	3
ES2008, ES2013, TS3010			6	
TS3011, IS1006	ES2011	TS3003	7	4
ES2010, IS1007, ES2015			8	
ES2009, TS3012	IS1008	TS3007	9	5
ES2012, IS1005, ES2016			10	

Table 21: Split of the scenario data into ten folds for cross-validation (rows), and the relation to the training, development and test sets (columns). For ten-fold cross-validation, each row describes a part, for five-fold, every pair of lines describe a part.

Orthographic transcriptions are available for the entire corpus, and each meeting has been manually segmented and annotated in terms of Dialogue Acts, using the ICSI MRDA scheme [Shriberg et al., 2004]. The MRDA scheme is based on a hierarchy of DA types and sub-types (11 generic tags and 39 specific sub-tags), and allows multiple sub-categorisations for a single DA segment. This extremely rich annotation scheme results in more than a thousand unique DAs, although many are observed infrequently. To reduce the number of sparsely observed categories, we have adopted a reduced set of five broad DA categories [Ang et al., 2005a, Zimmermann et al., 2005a]. Unique DAs were manually grouped into five generic categories: statements, questions, backchannels, fillers and disruptions. The distribution of these categories across the corpus is shown in table 22. Note that statements are the most frequently occurring segments, and also the longest, having an average length of 2.3 seconds (9 words). All the other categories (except backchannels which usually last only a tenth of a second) share an average length of 1.6 seconds (6 words). An average meeting contains about 1500 DA segments.

Dialogue Act	% of total DA segments	% of corpus length
Statement	58.2	74.5
Disruption	12.9	10.1
Backchannel	12.3	0.9
Filler	10.3	8.7
Question	6.2	5.8

Table 22: Distribution of DAs by % of the total number of DA segments and by % of corpus length.

The corpus has been subdivided into a training set (51 meetings, ca. 80.000 DAs), a development set (11 meetings, 13.500 DAs) and a test set (11 meetings, 15.000 DAs). This leaves out 2 of the 75 meetings, which were left out due to their different nature. All our experiments were conducted on this subdivision proposed by [Ang et al., 2005a] in order to have directly comparable results.

4.1.4 The Dialogue Act Recognition Task

The DA recognition task comprises two related sub-tasks: segmentation, and classification or tagging. These tasks may be performed jointly or sequentially. In a sequential approach the conversation is first segmented into unlabelled DA segments, then each detected segment is tagged with a DA label. The joint approach performs both tasks concurrently, detecting DA segment boundaries and assigning labels in a single step. The joint approach is able to examine multiple segmentation and classification hypotheses in parallel, whereas only the most likely segmentation is supplied to the DA classifier in a sequential approach. The joint approach is potentially capable of greater accuracy, since it is able to explore a wider search space, but the optimization problem can be more challenging. In a sequential system the two sub-tasks can be optimised independently. Note that an integrated system may be used as a segmenter by ignoring its classifications. For purposes of comparison, often it may also be used as a classifier, by forcing a human DA segmentation onto it.

Most previous work concerned with DA modelling has focused on tagging presegmented DAs, rather than the overall recognition task which includes segmentation and tagging. Indeed, automatic linguistic segmentation [Stolcke and Shriberg, 1996, Shriberg et al., 2000, Baron et al., 2002] is often regarded as a research problem itself.

4.1.5 Features

Although the tasks of dialogue act segmentation and classification are related, different types of features are employed for each of them. Table 23 lists some of the features used for DA classification in previous work, while Table 24 lists those that were used for the task of DA segmentation. The most common features used for automatic DA segmentation and classification can be subdivided in:

Lexical features Usually a language model based on words: DA specific ngrams of words, polygrams, factored language models, part-of-speech ngrams, etc. Some systems also rely on selected cue words/phrases and specific lexical or grammatical patterns. The number of words contained by the current DA segment (sentence length) is also a lexical related feature frequently adopted for DA classification. In order to evaluate fully automatic DA tagging and recognition systems, automatic ASR transcriptions are required. Inaccuracies of the automatically recognised speech have an adverse effect on lexical derived features. Therefore it is worth evaluating the full system both on manual and automatic transcriptions in order to estimate the overall degradation of performances caused by the ASR output.

Prosodic features represent a wide group of acoustic related features like: F0 and pitch slopes, the duration of words, unvoiced pauses, speech rate, features derived from spectral coefficients, etc.

Context features describe the relation between the current and the surrounding utterances, e.g. to indicate temporal overlap between speakers.

A discourse model (or discourse grammar) is based on the DA types of the preceding or surrounding segments. It is important to note whether this history is maintained on the actual output of the DA classifier, or on the hand-annotated DAs. For a realistic evaluation, the actual classification results should be used; however, generating the history from annotated DAs gives an estimation of the potential usefulness of this kind of features. Note that DA type related features can obviously not be used by a stand-alone DA segmentation program.

Two important aspects related to the feature extraction process are source and scope of the extracted features. Even if all the information required for feature extraction should come from fully automatic approaches, several systems are trained on features relying on manually labelled data. Moreover many systems are frequently evaluated using features based on manual annotations (i.e: lexical features estimated using the reference orthographic transcriptions), either because data from an automatic system is not available yet, or to assess the potential usefulness of a new feature family. Automatic DA processing is often a component block of a larger infrastructure , specific constraints imposed by the applicative domain have a deep influence on the feature scope. For example, in a meeting browsing application designed to offer its facilities online during an undergoing meeting, the DA recognition process will have access only to the past conversations. Note also that in this application the DA processing should operate in real-time relying on a less accurate ASR transcription. In a post-processing application (i.e: offline meeting corpus browser), the whole discourse is available, allowing the use of features which look ahead in the time.

Feature / Article	[Ang et al., 2005a]	[Rosset and Lamel, 2004]	[Fernandez and Picard, 2002]	[Rotaru, 2002]	[Lendvai et al., 2003]	[Andernach, 1996]	[Reithinger and Klesen, 1997]	[Venkataraman et al., 2002]	[Venkataraman et al., 2003]	[Keizer and Akker, 2005]	[Venkataraman et al., 2005]	[Jurafsky et al., 1998]	[Zimmermann et al., 2005a]	[Zimmermann et al., 2005b]	[Warnke et al., 1997]	[Katrenko, 2004]	[Webb et al., 2005]	[Ji and Bilmes, 2005]	[Surendran and Levow, 2006]	[Liu, 2006]	[Dielmann and Renals, 2007b]	[Verbree et al., 2006c]
Sentence length	✓									✓	✓									✓	✓	✓
First two words	✓	✓									✓									✓		
Last two words	✓										✓									✓		
Number of utterances		✓																				
Bigrams of words in segment				✓																		
Bigram of first two words																				✓		
Utterance type						✓																
Presence/absence Wh-words						✓																
Subject Type						✓																
Specific cue words/phrases						✓						✓				✓						✓
First verb type						✓																
Second verb type						✓																
Question mark						✓																✓
Sparse bag of ngrams																			✓			
Specific patterns										✓												
Grammar pattern										✓		✓										
Polygrams of words							✓								✓							
Factored Language Model																		✓			✓	
Part Of Speech ngrams																						✓
Ngrams of words								✓	✓		✓		✓	✓			✓	✓		✓		✓
First word of next segment	✓										✓									✓		
Speaker (turn) change		✓							✓		✓								✓	✓		
Words in last 10 DA's					✓																	
Pitch			✓		✓				✓			✓			✓				✓		✓	
Energy			✓		✓							✓							✓		✓	
Duration			✓		✓				✓			✓			✓				✓		✓	
Pauses					✓				✓			✓			✓						✓	
Rate of speech					✓														✓			
Ngrams of previous DA's								✓	✓		✓			✓				✓	✓		✓	✓
Previous DA hyp. / posteriors		✓								✓												
Next DA										✓												
Previous 10 DAs (from ref.)					✓																	

Table 23: Features used for DA-classification in different studies

Feature / Article	[Dielmann and Renals, 2007a]	[Kolar et al., 2006b]	[Stolcke and Shriberg, 1996]	[Lendvai and Geertzen, 2007a]	[Zimmermann et al., 2006a]	[Dielmann and Renals, 2007b]	[Ang et al., 2005a]	[Zimmermann et al., 2006a]
Segmentation only		✓	✓					
Surrounding Words					✓			
Ngrams of words		✓	✓				✓	✓
Part Of Speech ngrams			✓					
Tokenized Words				✓				
Bag of Words				✓				
Word length	✓							
Word relevance	✓							
Factored Language Model						✓		
Disfluencies				✓				
Repeats		✓						
Overlapping Speech				✓				
Pauses	✓	✓		✓	✓	✓	✓	
Pitch	✓	✓				✓		
Duration		✓				✓		
Energy	✓	✓				✓		

Table 24: Features used for DA-segmentation in different studies.

4.1.6 Metrics and Evaluation

Each of the segmentation, classification and the joint segmentation and classification tasks, has its own set of performance metrics. If performance evaluation is straightforward for the DA tagging task, the same cannot be said about DA segmentation or recognition tasks. Several evaluation metrics have been proposed, but the debate on this topic is still open. In our experiments we have adopted all the performance metrics proposed by [Ang et al., 2005a] and subsequently extended by [Zimmermann et al., 2005a], also the NIST-SU error metric introduced in [NIST website, 2003].

Classification metrics The performance of DA classification using manually annotated segments is usually measured in terms of accuracy, which is the percentage of correctly classified segments, or classification error rate, which is the percentage of incorrect classifications. For a more detailed evaluation, occurrences and correct classifications of each DA type are counted separately:

$$\begin{aligned} correct_{DA} &= \textit{the number of times DA was correctly classified} \\ annotated_{DA} &= \textit{the number of occurrences of DA in the annotated test data} \\ tagged_{DA} &= \textit{the number of times DA was classified} \end{aligned}$$

Based on these counts, we define the recall and precision measures for each DA type, as well as the accuracy and mean precision for the whole test set:

$$\begin{aligned} Recall_{DA} &= \frac{correct_{DA}}{annotated_{DA}} \\ Precision_{DA} &= \frac{correct_{DA}}{tagged_{DA}} \\ Accuracy &= \frac{\sum_{DA} correct_{DA}}{\sum_{DA} annotated_{DA}} \\ Precision &= \frac{\sum_{DA} Precision_{DA} * annotated_{DA}}{\sum_{DA} annotated_{DA}} \end{aligned}$$

Segmentation metrics Figure 7 illustrates the performance metrics used in the experiments described below. NIST-SU, recall, precision, f-measure and boundary are based on boundaries. Each word is followed by a potential boundary position, and segmentation is a binary classification into boundaries and non-boundaries. There are four possible outcomes: boundaries may be correctly identified (true positives, tp) or missed (false negatives, fn), non-boundary positions may be correctly identified (true negatives, tn) or a false boundary may be hypothesised (false positives, fp). The sum $tp + tn + fp + fn$ is equal to the number of words. The occurrences of these four events are counted. The

Reference	S|Q.Q.Q.Q|S.S.S|B|S.S|
System	S|Q|S|Q.Q|D.D.D|S.S|S|
NIST-SU	.c.e.e...c.....c.e.e.c
Boundary	.c.e.e.c.c.c.c.c.e.e.c
Recall	.c.......c.....c.e...c
Precision	.c.e.e...c.....c...e.c
DSER	c|...e...|..c..|e|.e.|
Strict	c|e.e.e.e|c.c.c|e|e.e|

Metric	Counts	Reference	Rate
NIST-SU	3 FP, 1 miss	5 boundaries	80%
Boundary	3 FP, 1 miss	11 (non-)boundaries	27%
Recall	4 correct	5 boundaries	80%
Precision	4 correct	7 hypothesised boundaries	57%
F-Measure	-	-	67%
DSER	3 match errors	5 reference DAs	60%
Strict	7 match errors	11 reference words	63%

Figure 7: Metrics for segmentation based on boundaries (NIST-SU, Recall, Precision, F-Measure and Boundary) and on segments (DSER and Strict). The symbol '|' is used to indicate boundaries between consecutive DAs and '.' stands for non-boundaries between words. The letters S, Q, D, and B represent single words of the DAs. Correctly hypothesised boundaries are marked with a letter c while e is used to label false positives and missed boundaries.

boundary-based metrics take different combinations of these counts into consideration:

$$
\begin{aligned}
NIST-SU &= \frac{fp+fn}{tp+fn} \\
Boundary &= \frac{fp+fn}{tp+tn+fp+fn} \\
Recall &= \frac{tp}{tp+fn} \\
Precision &= \frac{tp}{tp+fp}
\end{aligned}
$$

The F-measure is the harmonic mean of the computed precision and recall given the reference sentence boundaries and the boundaries hypothesised by the segmentation system: $F = 2 \times Recall \times Precision/(Recall + Precision)$. The other two segmentation metrics, DA segment error rate (DSER) and Strict, are based on segments. DSER is the fraction of reference segments which have not been correctly recognised, meaning that either of the boundaries is incorrect. Strict is a variant of DSER in which each DA segment is weighted with its length (number of words).

Joint segmentation and classification metrics The DA recognition task is more challenging, since the limited accuracy of automatic segmentation and classification are

Reference	S|Q.Q.Q.Q|S.S.S|B|S.S|
System	S|Q|S|Q.Q|D.D.D|S.S|S|
NIST	.c.e.e...c.....e.e.e.c
Strict	c.e.e.e.e.e.e.e.e.e.e.
Lenient	c.c.e.c.c.e.e.e.e.c.c.
DER/Recall	c|...e...|..e..|e|.e.|
Precision	c|e|e|.e.|..e..|.e.|e|

Metric	Counts	Reference	Rate
NIST	3 FP, 1 miss, 1 subst.	5 boundaries	100%
Strict	10 words	11 words	91%
Lenient	5 words	11 words	45%
DER	4 erroneous dialog acts	5 dialog acts	80%
Recall	1 correct dialog act	5 dialog acts	20%
Precision	1 correct dialog act	7 dialog acts	14%
F-Measure	-	-	17%

Figure 8: Metrics for joint segmentation and classification: the boundary based NIST error rate, the word based strict and lenient metrics, as well as the DA error rate (DER). The DA based recall, precision, and corresponding F-measure are illustrated in the lower part of the table. The symbol '|' is used to indicate boundaries between consecutive DAs and '.' stands for non-boundaries between words. The letters S, Q, D, and B represent single words of the same DA segment; S, Q, D, and B also represent the dictionary of 4 possible DA labels. Correctly recognised elements are marked with a letter c while e is used to mark errors.

combined together. Note that a direct comparison between DA recognition and classification results is difficult. However the DA classification performance can be interpreted as an upper boundary for the whole recognition process, which would be reached if automatic segmentation was perfect.

A set of metrics, in analogy to the segmentation metrics of section 4.1.6, can be defined for the recognition task. Figure 8 illustrates a set of performance metrics for joint segmentation and classification of DAs. In contrast to the NIST error metric for segmentation, the hypothesised DA label is taken into account as well, leading not only to false positives and misses but also to substitutions. While the strict error metric requires correct DA boundaries the lenient metric completely ignores segmentation errors. As the DER can also be defined via a DA based recall, DA based precision can be defined as well, leading to a DA based F-measure: $F = 2 \times Recall \times Precision/(Recall + Precision)$. Note that recall, precision and F-measure are based on dialogue act segments, not on DA boundaries as it was for the segmentation metrics.

While higher values for Recall, Precision and the F-measure indicate higher performances, the remaining metrics are error metrics, thus higher values imply lower performances. It is important to note that these metrics and all evaluations presented in this chapter are intrinsic, being purely based on the comparison between human annotation and classifier/recogniser output. Knowledge of the discourse structure could be beneficial in several

applicative domains; thus the automatically classified/recognised DAs often form the input of further processing stages. However the effects of DA segmentation errors and DA misclassifications on the overall system performances depend on how the DA recogniser output was used. These effects are not taken into account by the metrics defined in table 7 and 8, and are not examined here. Ideally, the users of a DA segmenter/classifier should separately investigate the effects of different DA recognition errors. Given such analysis, the most appropriate metric can be identified, and the DA recognition system can be optimised for this specific application.

Evaluation on speech recogniser output The reference DA annotation is produced on top of the manually transcribed word sequence. When the reference orthographic transcription is replaced by the ASR output, the DA tags need to be applied to a different word sequence, owing to ASR errors. Since a manual re-annotation of the ASR output would be extremely expensive, the evaluation scheme proposed by [Ang et al., 2005a] is often adopted: ASR words are mapped into the manually annotated segments according to their midpoint $0.5 * (word\_start\_time + word\_end\_time)$, thus inheriting their reference DA labels.

The systems described in this section have been evaluated on manually written reference transcriptions, and on preliminary output from an automatic speech recognition (ASR) system [Hain et al., 2006].

Since only the manual transcriptions have been annotated for dialogue acts, these annotations were aligned with the ASR output in order to train and evaluate systems on automatically recognised words. An ASR word was assigned to a dialogue act segment if the mid-point of the word lies within the boundaries of the dialogue act [Ang et al., 2005a].

Insertions and deletions Since the proposed alignment method is segment-based, insertions and deletions of single words are ignored. However, insertions and deletions of entire DA segments occur if the recogniser finds words outside of the boundaries of any annotated dialogue act, or if no words are recognised within the boundaries of an annotated DA.

ASR data is available for 101585 dialogue acts; alignment results in 91537 annotated dialogue act segments with recognised words, and 9968 DA segments without words. Although this is a large fraction, the information loss is likely to be less severe, as 66% of the deleted segments contain only laughs, coughs and other non-speech noises; 70% are of type Fragment and have no function in the discourse. While 49.2% of the segments of type Fragment are deleted, the loss on all other types is less severe, between 1% and 7%. Only 14% of the deleted segments are non-Fragments containing more than one word.

For the evaluations on AMI ASR data presented in this chapter, inserted words were ignored in the accuracy metrics. The deleted DA segments, however, were considered in different ways:

Include deletions as misclassifications Using the currently available ASR output, there is no indication that a dialogue act has taken place unless words from it were recognised. Therefore, deleted segments as errors can be included as errors.

Classify deletions Later versions of the ASR system may, however, provide the information that a participant spoke, even when no words were recognised. Therefore, we also included these segments as ordinary dialogue acts without words. They can be classified using non-lexical features like the duration, overlap with previous DAs, or the classes of the previous DAs. Classifiers which are limited to lexical features can choose the most frequent class. Also, this type of evaluation allows a closer comparison to results on manual transcriptions.

Exclude deletions Deletions can be excluded from the accuracy metrics, which shows the possible performance of the classifier on ASR words more clearly.

Impact of ASR on DA classification Classification accuracy on recognised words is approximately 10% (absolute) lower than on reference transcriptions. Note that these results have been obtained using preliminary ASR data, which contain some known errors; for future releases of the ASR output, we expect better classification, as well as a lower number of insertions and deletions.

4.1.7 Related Work

[Stolcke et al., 2000] provide a good introduction to dialogue act modelling in conversational telephone speech, a domain with some similarities to multi-party meetings. Dialogue acts may be modelled using a generative hidden Markov model [Nagata and Morimoto, 1993], in which observable feature streams are generated by hidden state DA sequences. Most DA recognisers are based on statistical language models evaluated from transcribed words, or on prosodic features extracted directly from audio recordings. Various language models have been tried, including factored language models [Bilmes and Kirchhoff, 2003], although any kind of trainable language model can be adopted. Prosodic features provide a large range of opportunities, with entities such as duration, pitch, energy, rate of speech and pauses being measured using different approaches and techniques [Shriberg et al., 1998, Hastie et al., 2002]. Other features, such as speaker sex, have also been usefully integrated into the processing framework.

[Ang et al., 2005a] addressed the automatic dialog act recognition problem using a sequential approach, in which DA segmentation was followed by classification of the candidate segments. Promising results were achieved by integrating a boundary detector based on *vocal pauses* with a hidden-event language model HE-LM (a language model including dialogue act boundaries as pseudo-words). The dialogue act classification task was carried out using a maximum entropy classifier, together with a relevant set of textual and prosodic features. This system segmented and and tagged DAs in the ICSI Meeting Corpus, with relatively good levels of accuracy. However results comparing manual with automatic ASR transcriptions indicated that the ASR error rate resulted in a substantial reduction in accuracy.

Using the same experimental setup, [Zimmermann et al., 2005a] proposed an integrated framework to perform joint DA segmentation and classification. Two lexical based approaches were investigated, based on an extended HE-LM (able to predict not only the DA boundaries but also the DA type), and a HMM part of speech inspired approach.

Both these approaches provided slightly lower accuracy when compared with the two-step framework [Ang et al., 2005a], but this may be accounted by the lack of prosodic features.

[Ji and Bilmes, 2005] propose a switching-DBN based implementation of the HMM approach outlined above, which they applied to dialogue act tagging on ICSI meeting data. They also investigated a conditional model, in which the words of the current sentence generate the current dialog act (instead of having dialogue acts which generate sequence of words). Since this work used only lexical features, and a large number of DA categories (62), a direct comparison with the results provided by [Ang et al., 2005a] is not possible.

[Venkataraman et al., 2003] proposed an approach to bootstrap a HMM-based dialogue act tagger from a small amount of labeled data followed by an iterative retraining on unlabeled data. This procedure enables a tagger to be trained on an annotated corpus, then adapted using similar, but unlabeled, data. The proposed tagger makes use of the standard HMM framework, together with dialogue act specific language models (3-grams) and a decision tree based prosodic model. The authors also advance the idea of a completely unsupervised DA tagger in which DA classes are directly inferred from data.

4.1.8 Structure of this Chapter

The remaining sections describe several systems for joint segmentation and classification and the separate tasks, employing different modelling approaches and corpora.

4.2 Segmentation

4.2.1 Abstract

The DA Recognition process consists of two main steps, the segmentation and the tagging. In this section the realization of the first step is discussed. The task of the segmentation is to subdivide a sequence of transcribed words (here from the AMICorpus) in terms of DA segments. As for the built classifier, it must be trained for the purpose to detect in unseen data temporal boundaries approximately similar to the annotated DA units. All experiments for segmentation are realized using the WEKA Machine learning toolkit [Witten and Frank, 2005]. WEKA is an open source library written in Java [11] making available a collection of machine learning algorithms for data mining tasks. For the segmentation task, the underlying implementation achieves a modular setup of the experiment environment that facilitates to communicate with the AMICorpus to infer requested data on demand. In order to use the learning machines by WEKA, information out of the Corpus gets converted and preprocessed by the interface tool. In the segmentation work the Bayes Net classifier by WEKA has been employed.

[11] WEKA is developed under GNU GPL and publicly available at http://www.cs.waikato.ac.nz/ml/weka

4.2.2 Bayes Net Classifier

A Bayesian network over a set of variables $U = x_1, ..., x_n$ is a network structure, that is a DAG (directed acyclic graph) over U and a set of probability tables $B_p = p(u|pa(u))$, where $u \in U$ and $pa(u)$ is the set of parents of u. So a Bayesian Network represents a probability distribution :

$$P(U) = \prod_{u \in U} p(u|pa(u)) \tag{20}$$

A Bayesian network can then be channeled into a classifier by simply calculating the $argmax_y P(y|\mathsf{x})$, where y is the class variable to be classsified given a set of variables x. Since those variables are known, $P(y|\mathsf{x})$ can be rewritten to $P(U)$. More information on the Bayesian Net classifier provided by WEKA is avalaible in [Remco, 2007].

4.2.3 Methodology

For the segment classification it is crucial to identify valuable features. The availability of information depends on the richness of data information registered in the corpus. The AMICorpus assigns to each word element the start signal and end signal on the timeline. Further there is a distinction between words spoken from the different speakers, identified by their roles. Every meeting consists of the following four speaker-roles :

- "PM" : The Project Manager (272929 words)
- "ME" : The Marketing Expert (193850 words)
- "ID" : The Industrial Designer (186177 words)
- "UI" : The User Interface Designer (175016 words)

The design of the experiments is based on the assumption that the respective speaker-roles undertake the task of representing expected functions in the meetings. For instance as a consequence of the charging imposed on the "Project Manager" to lead meetings, his performance reflects that he is the most active member. That observation holds throughout all meetings independently of the specific person playing the role. Currently the training and the evaluation of the classifier happens on the distinct speaker-role. In return the data extracted from the AMICorpus has been splitted for each speaker-role respectively. Hence the training of the classifier results in four different classifiers that are tested on the appropriate speaker-role dependent data.

The effects on the performance of the classifier give indication of the utility that a defined feature yields. The setup of the implementation enables easily to extend the information extraction out of the AMICorpus in order to specify additional features. It is worth to mention, that the Bayes Net learner by WEKA cannot handle with String data. Therefore words must be transformed to numeric values, where each number is assigned by a specific word in the data. If the application of the *Current Word* attribute is intended to use all words, then the number of attributes gets increased by the number of all occuring distinct

words in the given data. This latter case would affect the building time of the classifier seriously. Optional preprocessing methods on the input data required to model a classifier are included in the WEKA toolkit as well. In addition WEKA offers different evaluation techniques to make predictions on the usefulness of features without training and evaluating a classifier. In this work the validity of employing a specific feature is organized by the calculated Information Gain [12] on that given feature. The detailed specification of the Information Gain calculation is given in the next section 21. The ranking of the feature evaluation represents the background for the attribute selection we apply prior to the training of the classifier. In 10 the ranking of the used features are listed according to their informativeness for segment border detection. Different features static and dynamic as well have been integrated into the classifier. Static features are derivable from the given data, whereas dynamic features must be updated during evaluation of each instance (word element) taking into account the value of the classified previous instance. The list of all defined features employed to evaluate the classifier is displayed in 9.

[12] Other techniques for evaluating features like Chi^2, Gain Ratio and Symmetrical Uncertainty are also available, but as [Fung et al., 2007] has shown, these result in a similar ranking

$st(w)$:= start time of word in ms
$et(w)$:= end time of word in ms
w_n := word, that is subject to $n \in N$
$\mathcal{W}_n(w)$:= all (equal) words given in the Corpus
$\mathcal{S}(w)$:= number of syllable in w

#11 *Current Word* : w_n

#12 *Next Word* : w_{n+1}

#1 *Pause Duration* : $st(w_n) - et(w_{n-1})$

#2 *Duration of Word* : $et(w_n) - st(w_n)$

#3 *Mean Duration of Word* :

$$\frac{\sum_{i=0}^{n}(et(\mathcal{W}_i(w_n)) - st(\mathcal{W}_i(w_n)))}{n}$$

#4 *Relative Duration of Word* : **#2 - #3**

#5 *Distance to the last Segment in Words** : n

#6 *Distance to last Segment in ms** : $et(w_n) - st(w_0)$

#7 *Number of Words in the previous Segment**

#9 *Speech Flow Past* : $\frac{et(w_n) - st(w_{n-4})}{\sum_{i=-4}^{n} \mathcal{S}(w_i)}$

#10 *Speech Flow Future* : $\frac{et(w_{n+4}) - st(w_n)}{\sum_{i=n}^{n+4} \mathcal{S}(w_i)}$

#8 *Speech Flow* : **#9 - #10**

#13 *Relative Position of Word within Segment**

Figure 9: List of features, where dynamic ones are marked by the asterisk. #13 corresponds to the counter of 5-block words in [Dielmann and Renals, 2007c].

Rank	"PM"	"ME"	"ID"	"UI"
1.	#1	#1	#1	#1
2.	#3	#3	#3	#3
3.	#9	#9	#9	#9
4.	#6	#6	#6	#6
5.	#4	#8	#8	#11 (yeah)
6.	#8	#4	#11 (yeah)	#8
7.	#7	#7	#7	#4
8.	#11 (yeah)	#11 (yeah)	#4	#7
9.	#11 (okay)	#10	#10	#10
10.	#11 (so)	#5	#5	#5
11.	#10	#11 (so)	#11 (so)	#11 (so)
12.	#5	#11 (mm-hmm)	#11 (but)	#11 (but)
13.	#11 (mm-hmm)	#11 (mm)	#11 (okay)	#11 (okay)
14.	#11 (and)	#11 (okay)	#11 (and)	#11 (and)
15.	#11 (but)	#11 (but)	#11 (mm)	#11 (yeah)
16.	#11 (mm)	#11 (and)	#11 (yeah)	#11 (mm)
17.	#11 (um)	#11 (oh)	#11 (mm-hmm)	#11 (mm-hmm)
18.	#11 (oh)	#12 (yeah)	#11 (oh)	#11(oh)
19.	#11 (i)	#11 (i)	#11 (yes)	#11 (well)
20.	#11 (to)	#11 (to)	#12 (i)	#12 (i)
21.	#11 (the)	#11 (no)	#11 (the)	#11 (i)
22.	#12 (yeah)	#11 (um)	#11 (um)	#11 (um)
23.	#12 (okay)	#12 (i)	#11 (hmm)	#11 (to)
24.	#12 (to)	#12 (mm-hmm)	#11 (i)	#11 (the)
25.	#2	#13	#11 (to)	#11 (no)
26.	#11 (because)	#11 (the)	#11 (because)	#12 (okay)
27.	#11 (a)	#12 (mm)	#11 (well)	#12 (to)
28.	#11 (well)	#11 (a)	#11 (no)	#11 (of)
29.	#12 (i)	#11 (of)	#12 (to)	#11 (a)
30.	#11 (have)	#12 (okay)	#11 (a)	#11 (because)
...	...	...	...	...
100.	#11 (our)	#12 (up)	#12 (so)	#12 (maybe)

Figure 10: The Ranking of the Features after Selection through Info Gain. The most indicative attribute extracted from the AMICorpus turns out to be the pause feature independent of the speaker-role. Generally the first places in the ranking of best attributes do not differ significantly. Note that almost all "non-word" features appear before the first word-related feature is ranked.

4.2.4 Results

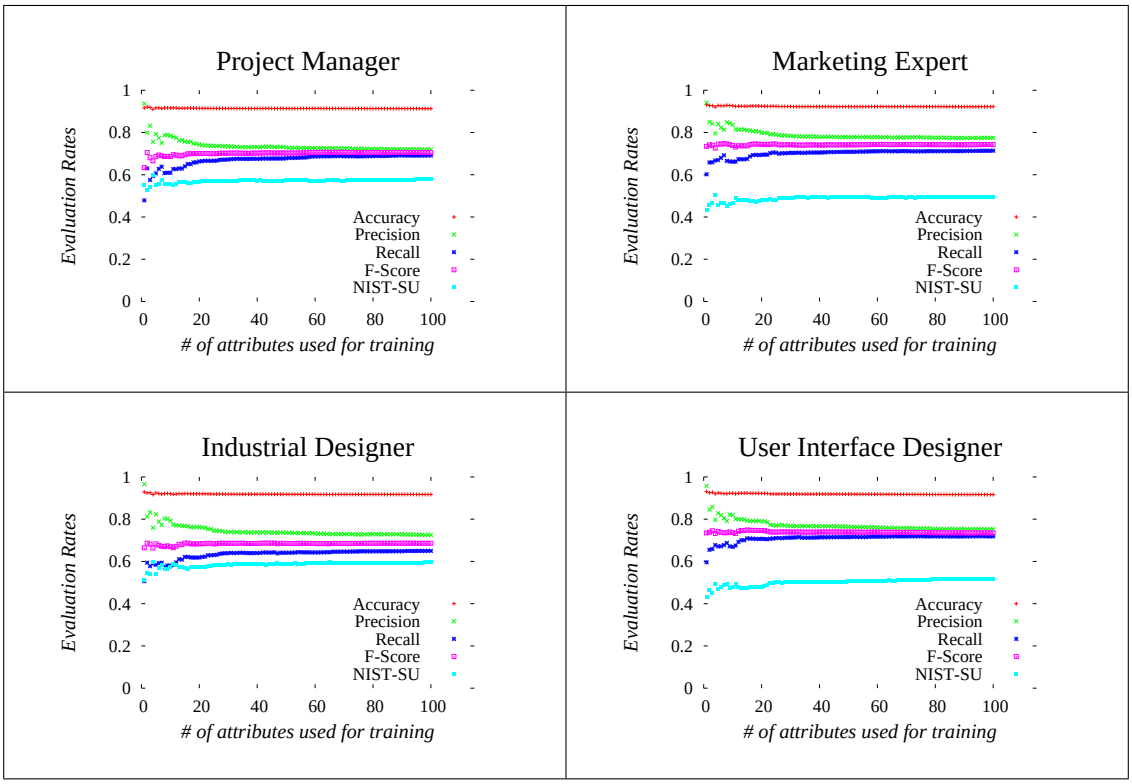


Figure 11: Performance flow of the experiments running through the 100 "best" attributes.

The experiments on segmentation are monitored on attribute selection given by the ranking of the features. At last the evaluation of the classifier containing the selected features permits to state which feature configuration obtains the best result. The four graphs above demonstrate the evaluation of 100 experiments for each speaker-role. For each of the experiments we succesively included one additional attribute following the order of the ranking, starting with the best feature (*Pause Duration*) and ending up by involving also the feature at the hundredth position. The analysis of the curve shows that all segmentation metrics converge with an asymtotic value the more attributes we apply to the classifier. Whereas comparing the evaluations among the classifiers with less than 10 attributes we notice more variance. On the next page the concrete results of the experiment of the classifier producing the best F-Score referring to the employed feature configuration is shown for each speaker-role.

	PM	ME	ID	UI
num FEATS	69	7	43	15
F-Score	0.7064	0.7478	0.6869	0.7491
NIST SU	0.5724	0.4667	0.5854	0.4750
Accuracy	0.9137	0.9265	0.9185	0.9234
Precision	0.7251	0.8135	0.7382	0.7939
Recall	0.6886	0.6919	0.6423	0.7091
num INS (train)	155649	104218	108024	97587
train time (*min*)	14,30	2,25	6,30	1,82
num INS (test)	34013	24516	22726	18117
test time (*s*)	11	2	5	2

Figure 12: Metric Measures based on the best F-Score speaker-role dependent. The average of the metric values are 0,7226 (F-Score), 0,5249 (NIST SU), 0,9205 (Accuracy), 0,7677 (Precision) and 0,683 (Recall).

Given the evaluation of experiments considering many different kind of feature configuration, we identify speaker dependently those with the best F-Score performance. The table above gives in the first row the information about how many of the best features are incorporated in the classifier that yields the best F-Score indicated in the second row. Note that NIST SU represents an error rate, where a decline in the rate value corresponds to an improved performance. Concerning the NIST SU, accuracy, precision and recall rates their values are adequately matched with respect to the F-Score. The last four rows deliver information of the size of the train and test data, further the time to model the classifier with the train data and respectively the time to evaluate the classifier on the test data.

4.2.5 Future Work

Besides experimenting with the Bayes Net classifier, we suggest to take other machine learning algorithms into account for the purpose of comparing the achieved rates of segmentation in this work. An additional goal is to acquire an adequate combination of the role-specific data to have only one classifier to deal with.

An other issue to be investigated is to make other predictions of the feature's utility on the level of attribute selection. Instead of evaluating a single feature exclusively based on its interaction with the class attribute, we want also to consider its triggering effect on the class attribute in combination with the other features in order to find out the optimal feature subset. Further analysis on the features data type, i. e., binning of numeric features is a subject in the future work as well.

Finally still not all information assumed to be a potential indicator for segment border detection are available in the corpus. Currently the effort lies in the motivation to supplement the word data with prosodic information, part of speech and also including multimodal features like movement and focus of attention.

4.3 Classification

This section describes dialogue act classification on AMIDA data, based on the ground truth segmentation. The idea is to contiune the work on dialogue act classification by doing *feature evaluation* as well as *classifier evaluation*. The system is implemented with the help of the freely available WEKA toolkit[13] which is explained in the previous section. A wrapper for the Maximum Entropy classification algorithm by the Stanford NLP group was implemented to add it to the WEKA classifier library. Former work ([Alexandersson et al., 2006]) showing good results with the MaxEntStanford classifier motivated this.

4.3.1 Methodology

The implemented system is designed to easy evaluate different features as well as different classification algorithms. It first trains the choosen classifier on a given training set with choosen features and then evaluates the trained classifier on a defined evaluation-set with the same features.
A variety of features are implemented which can roughly be divided into **static** and **dynamic** features.

- static features
  - **lexical features** N-grams, presence of "or", presence of "and".
  - **length and duration** The (discretized) length and duration values of the segment.
  - **temporal relation** Several features which indicate the temporal relation between the current segment and the previous one.
  - **speaker-related features** The speaker-role (PM, ID, ME, UI) of the current speaker and the speaker-change.
- dynamic features
  - **dialogue act history** The DA type of the last N dialogue acts splitted by speaker-change. If the previous DA type/s is/are unknown a dummy (*Missing*) value is inserted.
  - **dialogue act future** The DA type of the next N dialogue acts splitted by speaker-change. If the next DA type/s is/are unknown a dummy (*Missing*) value is inserted.

4.3.2 Feature Evaluation

[Hall, 1998] stated that some machine learning algorithms can be slowed down or their performance can be adveresly affected by *irrelevant* or *redundant* data. Therefore a careful selection of which features have relevant information and which one should not be fed

[13] WEKA is a Machine learning toolkit which is developed under GNU GPL and publicly available at http://www.cs.waikato.ac.nz/ml/weka

into the algorithm has to be performed before using/training a classifier. This section explains which types of feature evaluation methods were used and gives a short introduction in how every method works.

Information Gain Feature Selection The Information Gain (IG) of a given attribute x_i is a frequently employed technique for feature evaluation in the field of machine learning [Yang and Pedersen, 1997]. It measures the "goodness" of an attribute by knowing the presence or absence of an attribute in a segment (see equation 21).

$$\begin{aligned} IG(x_i) &= -\sum_{j=1}^{k} P(c_j)log(P(c_j)) \\ &\quad +P(x_i)\sum_{j=1}^{k} P(c_j|x_i)log(P(c_j|x_i)) \\ &\quad +P(\overline{x_i})\sum_{j=1}^{k} P(c_j|\overline{x_i})log(P(c_j|\overline{x_i})) \end{aligned} \tag{21}$$

Let $\{c_j\}_{j=1}^{k}$ denote the targeted classes.

Chi Square Feature Selection The *Chi-Squared* (or χ^2) feature selection algorithm evaluates features individually by measuring their chi-squared statistic w.r.t. the classes [Liu et al., 2002]. The χ^2 value of an attribute is defined in equation 22 where N is the number of examples in the dataset, I is the number of intervals, N_{ij} is the number of samples of the C_i class within the jth interval and M_{ij} is the number of samples in the jth interval.

$$\chi^2 = \sum_{i=1}^{C}\sum_{j=1}^{I}\frac{(N_{ij}-E_{ij})^2}{E_{ij}} \tag{22}$$

$$E_{ij} = M_{Ij} * C_i/N \tag{23}$$

The larger the χ^2 value, the more informative the corresponding feature is.

Correlation-based Feature Subset Selection The *Correlation-based Feature Subset Selection* algorithm (CFS) was developed by [Hall, 1998] and - like the majority of feature selection programs - uses a search algorithm to evaluate the merit of feature subsets. The measurement that calculates the quality of a feature subset takes into account the usefulness of individual features for prediction the class label along with the level of intercorrelation among them by maximising the first and minimizing the second value. The formal definition of the based hypothesis is shown in equation 24.

$$G_s = \frac{k\overline{r_{ci}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}} \quad (24)$$

$$r_{xy} = 2.0 * [\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}] \quad (25)$$

Where k is the number of features in the subset; $\overline{r_{ci}}$ is the mean feature correlation with the class and $\overline{r_{ii}}$ is the average feature intercorrelation.

Evaluation by hand An additional feature evaluation was done by hand. Which means that a classifier was trained and evaluated on every single feature to check if the classification algorithm gets any information out of the feature to classify the segments.

4.3.3 Classifier Evaluation

This section describes the classifiers that are evaluated so far. The evaluation is done on the whole feature set as well as on the best-feature-subset that was found in the feature evaluation to compare the performances.

MaxEntStanford The *Maximum Entropy* classifier is freely available from the Stanford NLP group. It has already been used in the AMI project ([Alexandersson et al., 2006]) and was already applied to the ICSI meeting corpus .

We use a conditional model of the probability of a class C (the DA type) given the feature-set derived from the segment X (see equation 26). The f_i are indicator functions defined on X and C. Each f_i is assigned a weight λ_i. The weights are the model parameters which are estimated from the training material, i.e., feature-sets whose classes are known.

$$P(X|C) = \frac{e^{\sum_i \lambda_i f_i(X,C)}}{\sum_{C'} p e^{\sum_i \lambda_i f_i(X,C')}} \quad (26)$$

In the maxent toolkit used here, there is an f_i for each feature/class pair (X_i, C), and λ_i indicates how strongly the presence of this feature suggests that the utterance in qeustion is of the class C.
A potential drawback is that maxent models do not capture correlations between features: if the co-occurence of two features has a particular meaning, it is up to the developer to generate an additional feature which represents the co-occurrence.

4.3.4 Results

The evaluation is based on the whole corpus as defined earlier in this document. Only "real" words (no noise or punctuation) were considered. As a result, empty segments that only contain noise words or punctuation symbols were deleted from the corpus.

Feature Evaluation The feature evaluation is based on the whole corpus merged from the TRAIN-, TEST- and DEVEL-sets.

Table 25 shows the features that have been choosen as the best 15 features from the whole feature set calculated by the IG-ranking algorithm. Compared with the top-15 features computed with the Chi-squared-ranking algorithm (see table 26, one can see that most of the features were choosen by both algorithms. In fact both algorithms valued 1181 features with values greater than zero (have useful information for the classification algorithm).

RANK	FEATURE	IG-VALUE
1	wordlengthOfSegment	0.47968
2	durationOfSegment	0.358112
3	intraNextDATxype1	0.162882
4	intraPrevDAType1	0.162836
5	speakerChange	0.141348
6	"the"	0.110307
7	"yeah"	0.10656
8	interNextDAType1	0.099369
9	DAhistory1	0.09702
10	DAfuture1	0.096961
11	tempRelContainment	0.093046
12	interPrevDAType1	0.089979
13	tempRelNoPause	0.085322
14	tempRelOverlap	0.076215
15	intraPrevDAType2	0.070828

Table 25: Feature Ranking of Information Gain algorithm

As the subset evaluation of more than 10000 features would not be computable, the n-gram features were omitted in the feature-subset evaluation process.
The CFS measuring computed a subset of 9 features out of the 31 features. The merit of the subset was valued with 0.178. As the correlation between the featues and the n-gram data is not evaluated, this information should be considered carefully. But one can see, that some features like "wordlengthOfSegment" and "durationOfSegment" which are in the subset were also measured with high values in the IG- and χ^2-feature evaluation.

Table 28 shows the results of the by-hand feature evaluation. A train and test cycle has been performed on every feature. If the classifier was able to gain some information out of this single feature, it was labeled as a worthwile features, while features where no information could be gained, were labeled as non-worthwile features.

Classifier Evaluation As mentioned before, empty segments were deleted from the train-corpus as well as from the test- and devel-corpus. That leads to an accuracy rate that is about 2% worse the accuracy rate that can be reached with the whole corpus. But it is good to do this step as it is a more target-oriented decision. In the targetted meeting-assisting environment, no empty segments occur because of the classified segments.

Rank	Feature	χ^2-Value
1	wordlengthOfSegment	66417.66943
2	durationOfSegment	52069.66221
3	intraPrevDAType1	32393.27846
4	intraNextDAType1	32376.17412
5	DAhistory1	19616.28099
6	DAfuture1	19578.14342
7	interNextDAType1	19027.69581
8	speakerChange	18969.68827
9	"yeah"	16720.29684
10	interPrevDAType1	16246.70873
11	"i'll"	16002.9321
12	tempRelContainment	15973.17576
13	"the"	14344.49521
14	"mm-hmm"	13977.05457
15	"sorry"	13788.69887

Table 26: Feature Ranking of Chi-Squared algorithm

Best feature-subset
wordlengthOfSegment
durationOfSegment
tempRelContainment
speakerChange
speakerRole
intraPRevDAType1
interPrevDAType1
intraNextDAType1
interNextDAType1

Table 27: Feature Ranking of CFS-subset measuring

Table 29 shows a comparision of the accuracy and average F-measure values gained by the MaxEnt classifier on all features (all), on the features that were measured with a positive value by the IG-ranking algorithm (+) and the feature-subset that was evaluated by hand (*).
The results show that taking all features first raises the train- and testtime and second decreases the performance of the classifier. Moreover the feature-subset calculated by the IG-ranking is not the best subset, as can be seen by comparing the values to the *-experiment.

Transcript vs. ASR A further target-oriented evaluation step has been enforced. As in real meeting scenarios, only data from automatic speech recognizer (ASR) is available and this data contains word-errors, a train- and test-cycle should be done on ASR-data. In fact, the accuracy degrades with about 8% while using ASR data instead of transcribed

Worthwile	Non-Worthwile
wordlengthOfSegment	tempRelPause
durationOfSegment	tempRelNoPause
DAhistory3	tempRelOverlap
DAhistory2	tempRelContainment
DAhistory1	speakerChangePrev
intraPrevDAType2	intraNextDAType3
intraPrevDAType1	intraNextDAType2
interPrevDAType3	intraNextDAType1
interPrevDAType2	absenceOfAND
interPrevDAType1	absenceOfOR
N-gram ($N = 1$)	N-gram ($N > 1$)
DAfuture3	
DAfuture2	
DAfuture1	
interNextDAType2	
interNextDAType1	
speakerRole	
speakerChange	

Table 28: Feature Ranking of by-hand measuring

Algorithm	Traintime [sec]	Testtime [sec]	Accuracy [%]	F-measure [%]
MaxEnt(all)	7317	315	65.59	45.29
MaxEnt+	5685	37	65.54	45.43
MaxEnt*	7730	245	65.67	45.75

Table 29: Evaluation results of different classification algorithms

data (see 30.

	ASR		REF	
Algorithm	Accuracy [%]	F-Measure [%]	Accuracy [%]	F-measure [%]
MaxEnt*	57.60	45.28	65.67	45.75

Table 30: Comparing ASR and Reference transcriptions

In the whole experiments, transcribed (gold-standard) DA-history and -future features were used. The system was implemented to use classified history and future values for training as well as for evaluating. If this is done, a degradation of 5% has been observed.

4.3.5 Future work

In the future additional prosodic features are planned to implement and evaluate, like pitch, energy, speech velocity. Also a comparision to other classifiers like Naïve Bayes or Decision Trees is aspired. Furthermore we plan to train an *Ensemble* classifier which can

easily generated with the WEKA toolkit. This classifier consists of many sub-classifier which all classify the given segment and a meta-measurement calculates the endresult.
An additional step is to seperate classifier- vs. inter-annotator confusion and get rid of them. Inter-annotator confusion was studied by Rieks op den Akker and could interfere the classifier in the training process.
Furthermore, the training and evaluation on classified segments is planned. This should be done with the automatic segmentation tool by Christian Schulz as it is also based on the WEKA toolkit.

4.4 Joint Segmentation and Classification

The DA recognition task comprises two related sub-tasks: segmentation, and classification or tagging. These tasks may be performed jointly or sequentially. In a sequential approach the conversation is first segmented into unlabelled DA segments, then each detected segment is tagged with a DA label. The joint approach performs both tasks concurrently, detecting DA segment boundaries and assigning labels in a single step. The joint approach is able to examine multiple segmentation and classification hypotheses in parallel, whereas only the most likely segmentation is supplied to the DA classifier in a sequential approach. The joint approach is potentially capable of greater accuracy, since it is able to explore a wider search space, but the optimization problem can be more challenging. In a sequential system the two sub-tasks can be optimised independently.

We present an approach to DA recognition that takes advantage of both techniques by employing a joint generative infrastructure (section 4.4.1) followed by a discriminative classifier (section 4.4.7). Both system components make use of supervised learning from manually annotated data, using the 15 AMI DA class annotation scheme. The joint recognition is coordinated by a switching DBN which integrates a discourse language model, six lexical and prosodic features , and two factored language models trained on the orthographic transcriptions. The recognised sequence of DA units is then re-classified using a conditional random field DA tagger trained using the lexical content and a set of discrete features.

4.4.1 The joint DA recognition system

We have developed a joint approach to DA recognition based on a switching DBN generative model. The observed features that are generated by this model are the words spoken by the meeting participants, together with a set of word-based prosodic features related to timing, intonation and energy. The mapping from DA labels to word sequences was modelled using a factored language model (FLM) and an interpolated FLM. The probability of observing a certain sequence of DA labels (discourse model) was represented through a simple trigram language model over DAs. The set of continuous word-based prosodic features was integrated into the recogniser using a Gaussian mixture model (GMM). The overall recognition process is actively controlled by a switching DBN which integrates information derived from words, prosodic features and language models.

We have used two sets of features in the DA recognition system: the transcription of the spoken words obtained using an ASR system and the continuous prosodic features.

Section 4.4.2 outlines the use of an automatic speech recogniser to produce a transcription, and section 4.4.3 outlines the extraction of the prosodic features. Sections 4.4.4 and 4.4.5 discuss the factored language models and the switching DBN model that underlie the DA recognition system.

4.4.2 Automatic transcriptions

Fully automatic DA recognition requires speech recognition. The AMI corpus has been manually transcribed at the word level, as well as being processed by an ASR system, thus enabling us to assess the robustness of the DA recognition system to speech recognition errors. Automatic transcriptions of the AMI meeting corpus provided by the WP4 were obtained using the AMI-ASR system [Hain et al., 2007]. This LVCSR system is based on decision tree clustered crossword triphone hidden Markov models, and a trigram language model. To recognize the complete corpus, a five-fold cross-validation was employed using equal splits of the corpus. Two transcription versions were generated: a fully-automatic one achieved by applying the full system on automatically segmented audio files; and a semi-automatic transcription obtained from a manual segmentation into utterances. The manual system also used a simpler ASR system, in which speaker adaptation was not used. In both cases the system operated on signals recorded from the close-talking microphones.

The automatic DA recognition experiments reported in section 4.4.6 compared both transcription versions. The speaker adapted "automatic segmentation" ASR output offers an overall improvement in terms of WER compared with the "reference segmentation" ASR output. However entire utterances may be deleted by the automatic acoustic segmentation, and consequently whole DA segments are irredeemably lost. Moreover, the word boundary times of the "manual segmentation" ASR output, are more accurate, compared with the reference orthographic transcription, since they cannot cross the manually annotated utterance boundaries. Accurately timed word boundaries are desirable for the extraction of prosodic features at the word level and are also required to evaluate segmentation into DAs.

Although both ASR versions offer valuable insights during the evaluation of our system, the "automatic segmentation" ASR output represents the main test condition since it does not implicate any manual intervention.

4.4.3 Prosodic features

Six continuous prosodic features were extracted for each word, using the audio signal and the transcription (figure 13): mean and variance of the the fundamental frequency (F0), mean energy, word duration, pause duration, and word relevance. For the reference transcription the times of word boundaries were obtained using a forced alignment against the audio. For the ASR transcriptions, the word boundary timings were output as part of the recognition process. The F0 tracks were estimated using ESPS get_f0 [Talkin, 1995], and the mean and variance were computed. The mean pitch was also normalised by speaker and by the average pitch for that term, with the objective of having a speaker independent measure able to highlight content words with a significant pitch shift. A

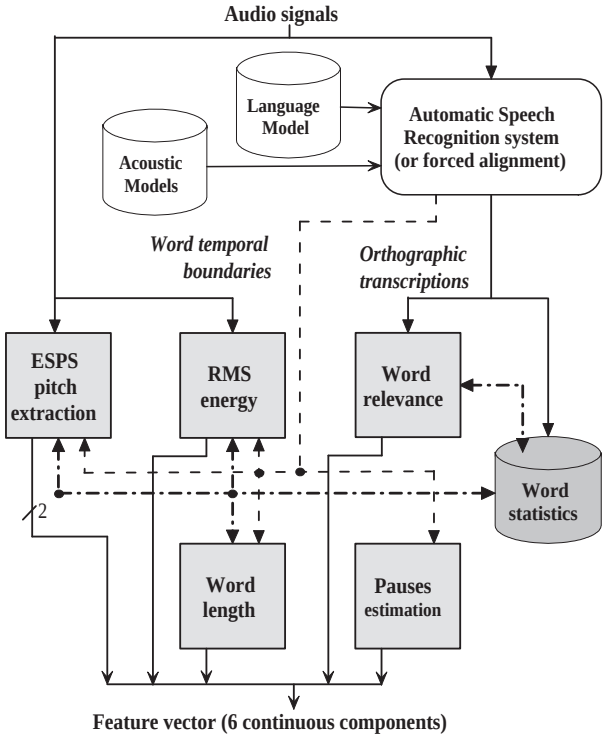


Figure 13: Data-flow of the automatic speech transcription and feature extraction process.

similar normalisation technique was applied during RMS energy estimation with the aim of compensating for different channel gains and to highlight emphasised words. Word duration was "term normalised" in order to highlight words which last more (or less) than the usual occurrences of that term. Inter-word pauses were also estimated from the word boundary times. Pauses are often associated with speaker turn alternations and other relevant changes in the conversational process such as topic shifts, and it is known that they provide a valuable cue for DA segmentation [Ang et al., 2005a, Zimmermann et al., 2006a]. Word relevance was estimated as the ratio between local term frequency within the current conversation and absolute term frequency across the whole meetings collection, thus assigning high scores to globally infrequent terms which occur frequently in the current conversation.

4.4.4 Interpolated Factored Language Models

Conventional language models construct a joint probability distribution over word sequences, $P(w_1, \ldots, w_n)$, which is factorised as a product of the conditional probabilities $P(w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-k})$. This concept can be generalised by replacing words $w_1, \ldots, w_n$ with bundles of factors $\mathbf{v_1}, \ldots, \mathbf{v_n}$, to construct a factored language model (FLM) [Bilmes and Kirchhoff, 2003]. Each factor bundle, $\mathbf{v}_t \equiv \left\{v_t^0, v_t^1, \ldots, v_t^k\right\}$, is a vector whose components are factors such as word identity, part of speech tag, word stem, and enclosing dialogue act label. Conventional LMs can be interpreted as a special case of FLMs with a single factor, the actual words: $\mathbf{v}_t \equiv w_t$. Word identities are usually included in the collection of factors employed in an FLM. The smoothing and discounting techniques used for

conventional LMs may be applied to FLMs, with the added flexibility of choosing which factor to drop when constructing simpler models for interpolation or backoff. Moreover, it is possible to drop more than one factor at a time and to follow multiple concurrent backoff paths using *generalised parallel backoff* [Bilmes and Kirchhoff, 2003]. FLMs have an increased number of degree of freedom, compared with conventional LMs, and it is possible to choose the factor set, the number of backoff steps, the backoff topology, and the discounting method associated to each backoff step.

We use FLMs to map word sequences into DA units, and we are primarily interested in evaluating these models in terms of DA labelling accuracy, rather than perplexity. It is possible to select the optimal FLM topology automatically [Duh and Kirchhoff, 2004], and we experimented with a simple search algorithm that randomly sampled the search space. The resulting models tended to employ a large number of factors (7 or more), implying many backoff steps. These automatically discovered topologies resulted in a slightly improved DA tagging accuracy (up to 2% absolute) when compared to manually developed FLMs, but the more intricate structure requires a more elaborate DBN infrastructure and substantially increases computational cost. In order to reach a trade-off between simplicity, cost and accuracy, we decided to employ a simpler FLM topology with three factors (and two backoff steps). Although this topology was initially designed by hand, it was also discovered by the automatic search procedure (with an improved set of discounting parameters).

The FLM that we used for the DA recognition task was based on three factors: the orthographic transcription w_t, the dialogue act label d_t associated to each word w_t, and the relative word position n_t in the context of the DA unit. The word sequence probability was modelled using a product of word bigrams conditioned also on word position and DA label, $P(w_t|w_{t-1}, n_t, d_t)$. The model was smoothed using two backoff steps and Kneser-Ney discounting. w_{t-1} was the first term to be dropped leading to a unigram like term, $P(w_t|n_t, d_t)$. In the case of a subsequent backoff the DA label factor d_t was the next term to be dropped, leading to $P(w_t|n_t)$. The FLM was estimated using the training subset of the AMI scenario meeting data (470 000 words and a dictionary of about 9 000 unique terms).

FLMs with the same topology may be interpolated, similarly to word-based n-grams. This enables the construction of combined models, whose component FLMs are trained using different data resources. We built FLMs for DA recognition using two additional corpora of conversational speech (the ICSI meetings corpus and the Fisher corpus of conversational telephone speech), in addition to an FLM built on the target AMI corpus, integrating them into a single interpolated factored language model. The Fisher corpus consists of more than 16 000 English telephone conversations on a wide range of elicited topics, resulting in about 2 000 hours of recorded speech, which has been orthographically transcribed. The AMI meetings corpus has a size of 0.97 million words in total, with about 0.47 million words in our training set of 98 meetings. The ICSI corpus, which is from a similar domain, contains 0.74 million words. The Fisher corpus is much larger, containing 10.62 million words. Building an interpolated FLM from these data sources, enriches the baseline FLM trained on AMI meetings only, by extending the vocabulary and thus reducing the out-of-vocabulary, and by improving the n-gram counts with word sequences that are not observed in the AMI training data-set alone. However, neither the

ICSI or Fisher corpora are annotated using the AMI DA annotation scheme. (The ICSI corpus has been annotated for DAs, but using a different and incompatible scheme.) In the absence of useful DA annotations, both the ICSI and FISHER corpora were duplicated 15 times when training the FLMs, labeling every sentence with all the 15 possible DA labels in the AMI DA annotation scheme. FLMs trained on artificially duplicated data are obviously not discriminative in a DA classification task, but they are able to enhance the dictionary and n-gram counts of the resulting interpolated FLM.

As will be discussed in section 4.4.6 the use of an interpolated FLM provides an improvement in DA segmentation at the price of slightly reduced DA classification accuracy. We report on experiments with a hybrid approach in which the baseline FLM trained on the AMI data is combined with an interpolated FLM at the sequence decoding level by maximising the product of the joint probabilities associated to the two concurrent FLMs.

4.4.5 Switching DBN architecture

In a DA recognition system, segmentation and classification are strongly related—the output of the DA classifier is dependent on the optimal placement of the DA unit boundaries, and the placement of the DA boundaries depends on the labels assigned to the DAs. In our approach, we treat the segmentation and classification problems jointly and the process is coordinated by a switching DBN model [Bilmes, 2000], implemented using the Graphical Model ToolKit (GMTK) [Bilmes and Zweig, 2002].

Figure 14 depicts the switching DBN model [Dielmann and Renals, 2007b]. The transcribed words are represented as the sequence of discrete observable nodes $W_0, \ldots, W_{t-1}$, W_t. The FLM and interpolated FLM outlined in the previous section are depicted using dotted arcs. The relative position of each word W_t into the current DA unit DA_t^0 is represented by the discrete node N_t. N_t relies on a bounded word counter C_t, which is incremented at every word encountered in the current DA unit. After each block of 5 words, C_t is reset to zero and N_t is incremented, thus indicating to which "block of five words" the current word W_t belongs to:

$$\begin{array}{lll} \text{if } C_{t-1} < 4: & C_t := C_{t-1} + 1 & \\ \text{if } C_{t-1} = 4: & C_t := 0 & N_t := N_{t-1} + 1 \end{array} \tag{27}$$

The final length of an automatically detected DA unit is not known a priori, and is only available at the end of the DA recognition process, therefore it is impractical to estimate word position features normalized for sentence length.

The DA recognition history is represented by the current and the two previous DA labelling hypotheses, DA_t^0, DA_t^1 and DA_t^2. This history is needed by the DA boundary detector, the hidden binary variable E_t. E_t is the principal switching variable in the model, switching from zero to one when a boundary between two disjoint DA units is detected. In the absence of a DA boundary ($E_{t-1} = 0$) the DBN assumes the *intra-DA* topology shown in figure 14A; when a boundary is likely to be present ($E_{t-1} = 1$) the model adopts the alternative *inter-DA* topology depicted in figure 14B.

The dependency of the observable prosodic feature vectors Y_t on E_t is modelled using a

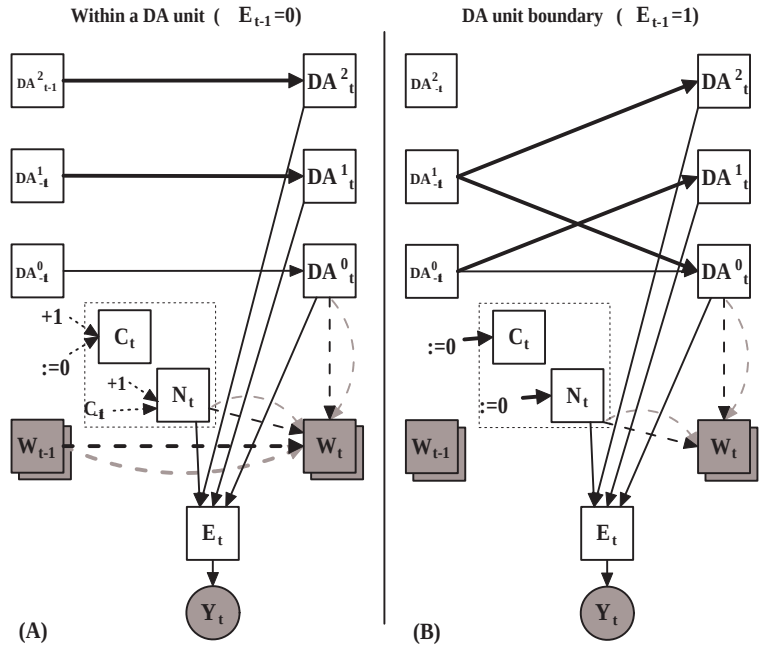


Figure 14: Switching dynamic Bayesian network model for the joint dialogue act recognition task: (A) *Intra-DA* topology adopted within a DA unit; (B) *Inter-DA* topology used at DA boundaries. The model switches between the two operating conditions (topologies) according to the state of the DA boundary detector node E. Square nodes represent discrete random variables, round nodes are continuous variables. Shaded nodes represent observable features, unshaded nodes are hidden variables. Plain arcs visually encode statistical dependences between random variables and dotted arcs highlight the dependences implied by FLMs.

Gaussian mixture model (GMM) with n components:

$$P(Y_t = y \mid E_t = i) = \sum_{j=1}^{n} C(i,j) N(y; \mu_{i,j}, \Sigma_{i,j}) \tag{28}$$

where $N(y; \mu_{i,j}, \Sigma_{i,j})$ is a Gaussian density with mean $\mu_{i,j}$ and covariance $\Sigma_{i,j}$, evaluated at y. $C(i,j)$ is the conditional prior weight of each mixture component j, and the optimal number of mixture components n for each state $i = [0,1]$ is automatically selected during training [Bilmes and Zweig, 2002]. The GMM relates the six-dimensional prosodic features to the two discrete states of E_t, thus helping to predict the DA segmentation.

The cardinalities of the discrete random variables reflect the function they serve in the model, thus: $|E_t| = 2$, $|C_t| = 5$, $|DA_t^0| = |DA_t^1| = |DA_t^2| = 15$, and W_t has as many states as the number of words in the dictionary. Since the vast majority of the DA units have fewer than 75 words, the word block counter cardinality has been constrained to $|N_t| = 15$.

The intra DA topology used within a DA unit (figure 14A) accumulates the joint probability for a sequence of k words $W_{t-k},\ldots,W_t$ as the product of a FLM and a weighted interpolated FLM given the current DA label hypothesis DA_t^0 and the deterministic counter nodes N_t and C_t. The absence of a DA boundary implies that the DA recognition history remains unaltered, hence the content of DA_{t-1}^1 needs to be cloned into DA_t^1 and similarly

$DA_t^2 := DA_{t-1}^2$. Since the word sequence $W_{t-k},\ldots,W_t$ has been generated by the same DA unit with label DA_t^0, and no DA boundaries have been spotted between time $t-k$ and time t, it follows that $DA_{t-k}^j = \ldots = DA_{t-1}^j = DA_t^j$ for $j = [0,2]$.

If a DA boundary is hypothesised ($E_{t-1} = 1$), then the model switches to the inter DA topology (figure 14B), which integrates the probability from the 3-gram discourse LM into the overall recognition process and starts the evaluation of a new DA unit, reinitializing the counter nodes: $C_t = 0$, $N_t = 0$. The DA recognition history is updated and a new set of DA classification hypotheses DA_t^0, for the next DA unit beginning with W_t, is generated following the 3-gram discourse language model $P(DA_t^0 \mid DA_{t-1}^1, DA_{t-1}^2)$.

When $t = 0$ a slightly modified intra DA topology ($E_{-1} = 0$) needs to be adopted: having both the DA recognition history and the counter nodes forcefully initialised to zero ($DA_0^1 = DA_0^2 = 0$, $C_0 = 0$, $N_0 = 0$).

Segmentation and classification are carried out concurrently. The classification process accounts for the joint probability of the transcription $W_{t-k},\ldots,W_t$ accumulated by the two concurrent FLMs given the current classification hypothesis DA_t^0, the probability of DA_t^0 given the two previously recognised DA units, and the segmentation hypothesis (a DA unit starting at time $t-k$ and ending at time t). Several alternative segmentation hypotheses are generated, with the probability of each segmentation combining the likelihood of generating the observed prosodic feature vectors Y_t and the likelihood of the DA unit generating the observed words $W_{t-k},\ldots,W_t$. A pruned Viterbi decoding is used to find the most likely sequence of labeled DA segments[14].

Since this approach cannot generate a DA segmentation without an associated DA labeling hypothesis, the segmentation accuracy is assessed by ignoring the recognised DA labels. Classification of the DA units for a reference segmentation can be achieved by constraining the state of the boundary detector nodes E.

4.4.6 Joint DA recognition of AMI meetings

We have used the switching DBN model for tagging, segmentation, and recognition of DAs in the AMI meeting corpus, using three language model configurations described in section 4.4.4: FLM, interpolated FLM, and a hybrid in which the interpolated FLM is focused on segmentation and the baseline FLM is focused on tagging. Each of these systems was run on three transcription conditions: manual reference transcription, ASR with manual utterance segmentation, and ASR with automatic utterance segmentation.

Error rates for the DA tagging, segmentation and recognition tasks, using the three system configurations and the three transcription conditions are shown in table 31. The three system configurations are as follows:

- *FLM*: simple FLM trained only on the AMI training set;
- *iFLM*: weighted interpolated FLM trained on AMI, ICSI and FISHER conversational data;

[14]The decoding runtime for this model is about 10 times slower than realtime on a 3Ghz P4 equipped with 1Gb of RAM.

- *Hybrid*: *iFLM* and *FLM* combined at the decoding level.

These three systems were each run on three transcription conditions, described in section 4.4.2:

- *Manual* Hand transcription;
- *ASR_AS* ASR with automatic segmentation: fully automatic system from ASR pre-processing up to DA segmentation and recognition (12.8% of DAs lost due to ASR deletions);
- *ASR_MS* ASR with manual segmentation: non-speaker adapted ASR with manual utterance segmentation (5.8% of DAs lost due to ASR deletions).

Although *ASR_MS* has a higher word error rate, the manual segmentation results in fewer complete DAs being deleted. Most of the deleted DA segments are very short, typically backchannels or fragments.

The *FLM* system has a classification error rate of about 10% absolute lower than the *iFLM* system for the tagging task, which uses a predefined segmentation. This is to be expected, since the additional data sources used in the *iFLM* system, the Fisher and ICSI corpora, do not have DA tags corresponding to the AMI scheme (section 4.4.4). Thus although these additional data sources extend the vocabulary and n-gram counts, they are unable to provide information to help discriminate between DA classes. The trigram discourse model contributes to these results by about 7.0% absolute: DA tagging experiments using the *FLM* system without the discourse trigram, resulted in classification error rates of 47.7%, 57.5% and 59.7% respectively for the *manual*, *ASR_MS* and *ASR_AS* transcriptions.

Precision and recall of DA tagging is shown by class in figure 15. This graph indicates that DA tagging accuracy is influenced by the imbalanced distribution of DA labels. Not surprisingly the classifier performs better on the two most frequent classes, *inform* and *backchannel*. However very infrequent classes such as *be-positive* and *offer* have good recall and precision scores, suggesting that even if rare they can be well modelled and discriminated.

For the DA segmentation task, table 31 indicates that the *iFLM* system results in much lower errors, by a factor of three, compared with the the basic *FLM* approach. In this case the reduced discrimination of the *iFLM* system is outweighed by the extended dictionary and larger language model, obtained from the additional ICSI and Fisher corpora.

Since DA recognition needs both accurate segmentation and classification, we combined the *FLM* and *iFLM*, resulting in a hybrid approach which combines the two models at the decoding level. The segmentation error rates of the *hybrid* system are slightly higher than those provided by the *iFLM* approach, and the tagging error rate is slightly higher than the *FLM* approach, but on the joint recognition task, which involves both classification and segmentation, the *hybrid* provides the lowest errors.

Compared with the reference transcription, the automatically produced transcriptions, *ASR_AS* and *ASR_MS*, result in increased error rates for DA tagging, segmentation and recognition. For tagging, the *ASR_AS* system results in an increased error of about 11% absolute, similar to that recorded on the ICSI tagging task [Dielmann and Renals, 2007a].

Task	Metric	Reference transcription			ASR manual segmentation			ASR automatic segmentation		
		FLM	iFLM	Hybrid	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
TAG.	100 - %Correct	**40.9**	51.4	42.8	**50.7**	61.2	53.0	**52.7**	61.9	54.8
S	NIST-SU	70.7	**20.4**	25.6	77.6	**26.5**	34.1	102.6	**30.7**	34.0
E	DSER	78.0	**12.8**	17.0	85.5	**17.0**	22.8	94.2	**23.2**	25.8
G	Strict	74.4	**28.5**	36.9	81.8	**29.4**	39.5	91.5	**26.9**	33.7
M.	Boundary	10.8	**3.1**	3.9	12.8	**4.4**	5.6	16.7	**5.0**	5.5
R	NIST-SU	93.1	73.6	**71.3**	98.3	85.3	**85.9**	114.8	84.0	**81.2**
E	DER	85.5	57.0	**51.9**	91.7	67.0	**62.5**	96.5	68.6	**64.1**
C.	Strict	83.2	64.4	**62.1**	89.2	70.7	**68.5**	94.5	68.3	**64.7**
	Lenient	40.9	51.8	**42.2**	43.8	59.0	**48.3**	43.4	57.1	**46.9**

Table 31: DA tagging, segmentation and recognition error rates (%) on the AMI meeting corpus; results are reported on 3 different FLM setups (baseline FLM, interpolated FLM, and hybrid FLM+iFLM) both on reference manual transcriptions and on 2 ASR outputs

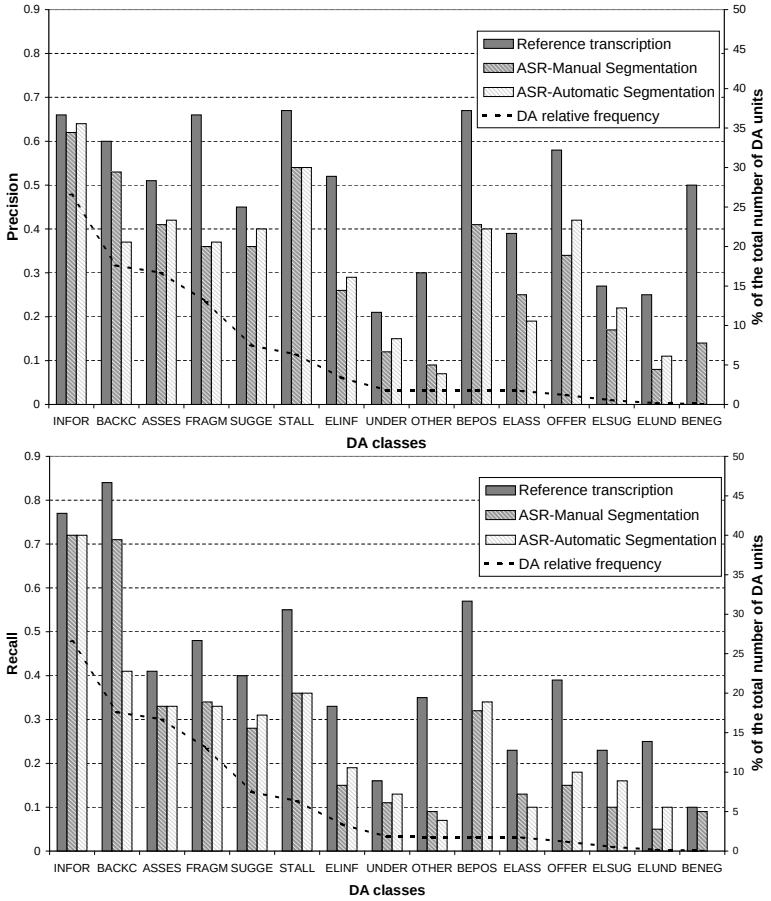


Figure 15: DA class based precision/recall metrics for the automatic DA tagging task on reference orthographic annotation and two versions of the ASR output. The 15 classes are sorted by their relative frequency in the AMI corpus.

Since the automatic DA segmentation strongly relies on the lexical content, a similar degradation can also be observed on DA segmentation metrics. The *iFLM* and *Hybrid* test conditions are less severely affected, suggesting that the larger language model results in a greater tolerance toward ASR inaccuracies. The full DA recognition task, representing a trade off between segmentation and classification, leads to an increase in the NIST-SU recognition metric by about 10% on *iFLM* and *Hybrid* setups and by 20% on the baseline *FLM* experiment.

However, the 12% of segments that are deleted in the *ASR_AS* transcription have an effect on the DA recognition results. In order to quantify this degradation, we compared the *ASR_AS* with the *ASR_MS* transcriptions which have an increased overall WER, but a reduced number of utterance deletions. Despite its higher WER, *ASR_MS* performs slightly better than *ASR_AS* on the isolated DA tagging task, although the lenient metric suggests that the situation is actually inverted when the DA classification is carried out as part of the joint DA recognition. Because of the lower number of deleted segments, *ASR_MS* outperforms *ASR_AS* on the DA segmentation sub-task using both the *FLM* and *iFLM* systems. A similar discourse applies to the overall recognition performances on the baseline *FLM* setup. Thanks to the more ASR tolerant interpolated FLM and to the improved *ASR_AS* transcription quality, which leads to better dynamic classification performances (Lenient

Recognition metrics	Reference transcription	ASR manual segmentation	ASR automatic segmentation
NIST-SU	59.2 (71.3)	70.3 (85.9)	71.3 (81.2)
DER	46.7 (51.9)	56.1 (62.5)	59.7 (64.1)
Strict	54.2 (62.1)	59.3 (68.5)	57.4 (64.7)
Lenient	36.0 (42.2)	40.6 (48.3)	40.5 (46.9)

Table 32: DA recognition error rates (%) of a CRF based re-classification system; Best prior recognition performances using the *hybrid* approach have been reported in brackets

metric), *ASR_AS* offers a slightly improved DA recognition over *ASR_MS* on both *iFLM* and *Hybrid* setups.

4.4.7 Discriminative re-classification of joint recognition output

The use of static discriminative classifiers to re-rank the output of sequential generative models has proven to be an effective technique in domains such as probabilistic parsing [Collins and Koo, 2005] and statistical machine translation [Shen et al., 2004]. Discriminative approaches have also been used to correct (or validate) the ASR transcription produced by a generative HMM system. Support Vector Machines trained on features related to the acoustics are used in [Venkataramani et al., 2007] to disambiguate confusable word pairs. In another application of static reranking of LVCSR n-best hypotheses, additional phonetic, lexical, syntactic and semantic knowledge were used to discriminate between multiple recognition hypotheses [Balakrishna et al., 2006].

This is an attractive approach for several reasons. First, since it is a post-processing method it may be applied to any preexisting system leaving it unaltered. Second, directly discriminant approaches explicitly optimize an error rate criterion, while exploiting temporal boundaries and recognition candidates estimated by the generative model. Finally, it is possible to add features to the joint recognition system, with the possibility of lower computational overhead.

We have applied discriminative re-ranking to automatic DA recognition, postprocessing the output of the *iFLM* system with a static discriminative classifier based on Conditional Random Fields [Lafferty et al., 2001a]. CRF are undirected graphical models frequently used with a simplified *linear chain* topology (first-order CRF) which can be interpreted as a generalisation of HMMs. Since CRFs are trained to maximise the conditional likelihood of a given training sequence, rather than the joint likelihood, they offer improved discrimination and a better support of correlated features. Moreover during CRF decoding the classification decision is taken globally over the entire sequence and not locally on a single observation.

The linear chain CRF has been used to associate DA labels with the best segmentations provided by the switching DBN. The prosodic features that we used in the generative model (with the exception of F0 variance) were discretised and used in conjunction with the lexical information during the CRF re-labeling process, implemented with CRF++[15].

[15] http://crfpp.sourceforge.net/

Table 32 reports the recognition performances after discriminative re-classification. The improvement is consistent on all the transcription conditions and on all the evaluation metrics, with reduction of 5–12% absolute.

4.4.8 Summary

We have presented a framework for the joint recognition of dialogue acts on AMI meetings. The system that we have presented employs a generative probabilistic approach implemented through the integration of a heterogeneous set of technologies: six continuous prosodic features extracted from the lexical and prosodic content facilitate the segmentation process; a trigram discourse language model estimated from observed sequences of DAs; a factored language model interpolated using multiple conversational data resources, used in conjunction with a plain FLM trained solely on in-domain data; and a switching DBN model with two alternating topologies, which coordinates the joint DA segmentation and classification task by integrating the available resources.

Three experimental systems were investigated based on a simple FLM, an interpolated FLM, or hybrid using both. The simple FLM trained only on data from the target AMI corpus offers the most accurate DA classification. However the interpolated FLM, thanks to its richer dictionary and language model, reduces the number of segmentation errors by a factor of 2–3, at the cost of a slightly degraded DA classification accuracy. A hybrid approach, using both FLMs, allows a trade off between segmentation and classification, to improve the overall recognition accuracy. Experiments using each of the three systems on hand-transcribed and two kinds of automatically transcribed data, showed that these systems generalise well to automatic imperfect transcriptions. A further improvement in the recognition accuracy, of 5–12%, was obtained using a discriminative DA re-classification process based on conditional random fields.

The degradation when moving from manual transcriptions to the output of a speech recogniser is less than 15% absolute for most tasks and metrics. These experiments indicate that it is possible to perform automatic segmentation into DA units with a relatively low error rate. However the operations of tagging and recognition into fifteen imbalanced DA categories have a relatively high error rate, even after discriminative reclassification, indicating that this remains a challenging task.

4.5 Evaluation and Classification

In this section we consider the dialogue act annotations in the AMI meeting corpus more closely. These human made annotations on the human made transcriptions of the recorded speech are used for training machine classifiers to segment speech into segments and classify them as either a meaningfull dialogue act utterance or some other type of vocal signal. We look at segmentation as well as at dialogue act classification because these tasks are strongly related. A DA segment is a sequence of subsequent tokens (in the hand transcribed speech or produced by ASR) that form a unit, because they express a single speakers intention. These notions are rather vague and that is the reason for the method that is followed in the development of automatic machine classifiers. If we had a set of rules that tells us how to segment speech into DA segments and how to tell what type they

are, we wouldn't need human annotators to build a corpus, then see what "features" can be used as usefull indicators for the various classes we want to distinguish, and then build classifiers that uses these features on new data to perform the task that it has learned from the annotated training data. Thus, although human annotators were trained to segment and label the transcribed speech into DA segments, they did not follow a set of explicit rules. They are supposed to get the "idea", pointed at by a number of example phenomena in which the idea is present. The annotation procedure is a first "specification" of what a DA segment is and how the types of dialogue acts are distinguished. The second form in which the notion of DA segment and DA types is implemented is the human annotated corpus, in which the annotators express their concepts in interaction with the observed phenomena that they have labeled. Human annotators have diverging concepts however, which is partly due to the vagueness of the ideas and partly to the ambiguity of the phenomena. They therefore make arbitrary decisions. And they sometimes make errors. Analysing human annotations and comparing them gives us insight in the intrinsical problems of the task, the ambiguities, and the types of ambiguities. If our annotated corpus consist of parts that were produced by different annotators, as the AMI corpus of DA annotations does, we then know what types of inconsistencies or variations we have in the annotated corpus.

The third implementation is the machine classifier. How good is this implementation of the concept that it is meant to be implemented? Can we rely on its output? Did it grasp the idea from the examples in the training data, expressed in the mapping of ensembles of feature values onto the classes to be distinguished?

"How good *is* our classifier?" If we evaluate the machine classifier we usually do this against a test corpus of human annotations. These annotations contain the same kinds of "errors" and arbitrary decisions as the annotations in the training corpus, that is used to learn the classifier to perform the task. One way to check the consistency between the parts of the corpus that were annotated by different annotators to train the classifier on one part and test it in an other annotators part. If performance measure decreases significantly, we know that the classifier is not very robust for the variations in the annotators implementations of the ideas.

Evaluation metrics of statistical classifiers are usually given in terms of probabilities. We would like to know how reliable the output of our statistical classifier is in a particular instance. Or on a particular class of instances. Therefore detailed error analysis is required.

Analysing the disagreements between human annotators gives an indication of the complexity of the task. Various error and performance metrics for segmentation of talk in conversations have been proposed. We applied these metrics to human annotations to see how good they perform. This gives an answer to the question "how good our classifiers *can* be?"

The question "how good our classifier *should* be?" can only be answered by considering the application to which the classifier offers its functionality. One of the questions is how fast it must do its task, the other is how harmful errors of a certain type are regarding the application.

Analyses of annotated data may also hint at new methods for classification. We will propose a method for DA segmentation and classification that is a combination of a simple

decision rule and a sequential statistical classifier. For the latter we used both a simple Viterbi generative classifier on the hidden event language model and a conditional random field classifier.

4.5.1 Dialogue Acts in the AMI Corpus

The AMI meeting corpus ([McCowan et al., 2005a]) has 15 classes of "dialogue acts" that fall into the following classes:

- Classes for things that aren't really dialogue acts at all, but are present to account for something in the transcription that doesn't really convey a speaker intention: backchannels, stalls and fragments
- Classes for acts that are about information exchange: inform and elicit inform.
- Classes for acts about some action that an individual or group might take: suggest, offer, elicit suggest or offer.
- Classes for acts that are about commenting on previous discussion: assess, comment about understanding, elicit assessment, elicit comment about understanding
- Classes for acts whose primary purpose is to smooth the social functioning of the group: be-positive, be-negative.
- A "bucket" type, OTHER, for acts that do convey a speaker intention, but where the intention doesn't fit any of the other classes.

Table 33 shows the dialogue act types and how often they occur in the corpus.

Dialogue Act Segmentation Instructions The DA annotators worked on the hand-made speech transcriptions, and their task was to segment the transcriptions into dialogue act segments, each of which conveys a speaker intention. Segmentation into DA segments is a tricky thing to do. The manual contains a few rules how the annotators should deal. They are illustrated with examples.

The first rule is: *each segment should contain a single speaker intention*.

- If a speaker, for instance, asks two different questions in a row, without anyone else speaking, each of them is a separate segment.
- If someone says "No, its not", the "its not" is not a separate segment, since it rephrases the same information as the "No".
- Lengthy pauses or conjunctions that introduce whole new clauses such as "so", "because", and some uses of "and", "but", and "or" can be hints that a new segment is starting.

Group	Dialogue Act Type	Frequency.
Info Exchange	Inform	28.3
	Elicit-Inform	3.6
Actions	Suggest	7.9
	Offer	1.3
	Elicit-Offer-Suggest	0.6
Discussion	Comment-about-understanding	1.9
	Assess	18.6
	Elicit-comment-about-understanding	0.2
	Elicit-assessment	1.9
Social Acts	Be-positive	1.9
	Be-negative	0.1
Segmentation	Backchannel	11.0
	Stall	6.8
	Fragment	14.0
Other	Other	2.0

Table 33: Dialogue act types and frequencies of their occurrence in the corpus.

- In case of a (sub-ordinate) conjunction if the first half requires the second half to be complete - neither segment expresses a complete intention - they should be combined into one segment.
- If the speaker changes from talking to one person to talking to someone else or the whole group, or the other way around, there would be two intentions, and therefore two segments, although the speakers intention is the deciding factor.

An example of how to split a speaker turn into DA segments is the following, in which || indicates segment boundaries.

> And then you have the numeric pad in the dark blue at the bottom, || and on the right-hand side you have the access to the menu on the T V , || and on the left-hand side you have the the the ability to turn off the voice recognition. || So this is pretty much what we had on the white board the last time.

Although not mentioned explicitly, the example indicates that the conjunctives are at the start of the second segment.

A second rule is that *all segments only contain transcription from a single speaker*. In theory it is possible for one speaker to start a dialogue act and for another to finish it such as when someone cant think of a word, so someone else fills it in. In these cases, a separate segment for each speaker is marked, giving each of the segments the same type. This is the only case when a single intention is split over more than one act.

The third rule is that *everything in the transcription is covered in a dialogue act segment, with nothing left over*.

In case of doubt, coders were instructed *to use two segments, instead of one*.

Note that the second rule allows to do DA segmentation on the speech of one speaker. Thus after speaker identification and separation of the speech of different speakers we can do segmentation into DA units on the material of one single speaker.

4.5.2 Error metrics and performance measures

There are several error metric and performance measures for segmentation proposed in the literatures. The figure shows a confusion table for the segmentation of the sequence of tokens into dialogue act segments. Each token (a word utterance recognized) is labeled either by a *B* (for begin new segment) or *I* (for internal, i.e. not a new segment begin).

	predicted	
true	B	I
B	tp	fn
I	fp	tn

Table 34: Confusion matrix for segmentation boundaries. fn times a segment boundary was missed (”false negatives”) and fp times a word was identified wrongly as start of a new segment (”false positives”).

Using the terms for the entries of the confusion table 34, the error metrics and performance metrics are defined as follows:

NIST-SU-Error $(fn + fp)/(tp + fn)$

NIST-SU-Boundary $(fn + fp)/(tp + fn + fp + tn)$ ([Ang et al., 2005b]. This is also called BER, Boundary Error Rate ([Kolar et al., 2006a])

Strict The number of words not in a correct segment divided by the total number of words. ([Ang et al., 2005b])

DSER Zimmermann et al. ([Zimmermann et al., 2006b]) proposes the DA Segmentation Error Rate which counts errors on the segment level instead of the word level as in the Strict metric. The total number of incorrectly identified reference segments is divided by the total number of reference segments.

The performance metric are the standard measures:

Accuracy $(tp + tn)/(tp + fn + fp + tn)$ - equals 1−NIST-SU Boundary.

Recall $tp/(tp + fn)$ (R - the more false negatives the lower the recall)

Precision $tp/(tp + fp)$ (P - if precision is high then the number of false positives is low.)

F-measure $F = 2 * R * P/(R + P)$

It depends on the requirements of the application whether we prefer a higher precision (if the system says that a token starts a new segment, we want it to be true), or a higher recall (we want the system not to miss starts of new segments.).

An other measure for annotator agreement is percent pairwise agreement; this equals $tp/(tp+fn+fp)$, the number of agreed segments divided by number of segments identified by at least one of the annotators.

4.5.3 How good can our classifier be?

How good is an accuracy of 80%? A reference to the state of the art, saying that this method is better than that, does say something but not what is achievable. To measure "goodness" we need to know what is the best possible. What is the best possible is hard to see directly, but several methods are used to obtain an indication of the "intrinsic" complexity of a classification task, and thus of what is achievable. One way is by looking at how trained human annotators perform the task. The assumption is that the more their annotations differ, the more complex the task is.

When we measure the performance of the machine classifier, we do that by comparing the output on an human annotated test set, that we consider as "ground truth". From an outside point of view, having complete knowledge of what is and what is not the case, we could observe that the result of the classifier, being trained by human annotators, agrees with the "ground truth", but nevertheless, both are wrong. In a more democratic world, we don't have insight in Truth, and we rely on statistics. Notice that this is a good thing, since the users of the classifiers fall under the same democratic rights, and the same laws of statistics, and they make the same kind of "errors" (to use an undemocratic qualification), as the human annotators.

Comparing annotators we can go either way: consider one of them as the norm, and measure the other against this norm, or, treat all annotators in the same way. The first method is usually done for measuring machine performance against that of human annotators, after all "of all things the measure is man" ([Steidl et al., 2005]). The second methods is performed when analysing human annotations. Notice that in the latter case we actually don't measure one annotator against another, but we measure pairs or sets of annotations, and what we aim at is inference about the quality of the annotations or the complexity of the task, not about how good or bad one human annotator is; we just measure, how much they agree. Kappa statistics are mostly used for this, since it takes chance agreement into account.

Viera treats her classifier as a human annotator ([Vieira, 2002] and uses kappa statistics to measure the agreement with other (real) annotators. The same method is followed by Rienks ([Rienks, 2007]. Steidl et al. ([Steidl et al., 2005]) also treats machine classifiers and human annotators alike. The way they compare "decoders" (either human or machine) is as follows. Assume all items have been coded by N annotators into K classes. Let item I be coded with label k $k(I)$ times. Then $(k(I)/N)$ is a normalized distribution of the labeling of item I, a "soft label". If we replace one decoder by another we get a new distribution. These distributions are compared using the entropy $H(I)$ of the distribution. The more the entropy differs the more "unexpected" the new decoder has coded the item. We can also consider the output of one human annotator as ground truth and measure the

performance of the other annotators as if it was output of a machine decoder. That is what we do. For each annotator pair (A_1, A_2) we calculate the performance of the human "classifier" A_2 on the segmentation task, and we take the annotation of A_1 as "ground truth". Two tables comprise the results.

The entries of table 35 contain recall and precision measures for boundary class *B*. The recall (precision) value for the pair of annotators (A_1, A_2) equals the precision (recall) value for the pair (A_2, A_1), i.e. when switching roles of the two annotations. The table contains recall as well as precision for both pairs.

Row dha column dha-c contains recall where dha is the truth value row dha-c column dha contains precision where dha is the truth value.

	dha	dha-c	mar	s95	s95-c	vka
dha	–	0.91	0.93	0.84	0.92	0.91
dha-c	0.93	–	0.94	0.84	0.92	0.91
mar	0.77	0.79	–	0.76	0.80	0.86
s95	0.81	0.83	0.89	–	0.85	0.87
s95-c	0.89	0.91	0.94	0.85	–	0.92
vka	0.72	0.74	0.83	0.72	0.75	–

Table 35: Recall values for boundary prediction (see main text for explanation).

Table 36 shows values for Dialog Segmentation Error Rate for pairs of human annotators. We see that DSER values vary from 0.19 (a cross coding) to 0.56.

	dha	dha-c	mar	s95	s95-c	vka
dha	1238	0.19	0.33	0.41	0.23	0.40
dha-c	0.23	1274	0.29	0.40	0.22	0.39
mar	0.45	0.40	1507	0.48	0.39	0.36
s95	0.44	0.41	0.39	1290	0.37	0.46
s95-c	0.26	0.23	0.29	0.37	1281	0.36
vka	0.53	0.51	0.39	0.56	0.48	1565

Table 36: Dialog Segmentation Error (DSER) Metric values for boundary prediction. The main diagonal contains the number of DA segments identified by the annotator. The bottom-left triangle contains DSER values when the row annotation is considered reference segmentation. The upper-right triangle contains the DSER values for the reversed case in which the column annotation is reference annotation.

A qualitative error analysis of the machine classifications reveals in what way these results differ from the ways that annotators disagree on segmentation and/or labeling.

A carefull analyses of the relation between kappa values computed on the human annotations and the performance of the machine classifiers trained, and tested, on these annotations reveals that there are two different types of disagreements between human annotators that both have impact on kappa and on the accuracy of the classifier. There can be systematic disagreement, and there can be noise. It is especially the former type of disagreements that may lead to a false sense of security that the data is good enough in

case kappa and accuracy values of 0.8 or higher are taken for granted. (see [Reidsma and Carletta, 2007]).

4.5.4 A simple decision rule for segmentation

A Simple Segmentation Rule to split up speech into DA segments is the following, which is only based on the lengths of periods of silence.

Simple Segmentation Rule A word (token) is marked as the start of a new segment when it comes after a period of silence that is longer than T seconds, otherwise it is considered to belong to the same segment as the previous words of the same speaker.

Thus a token is labeled B if it is the first token after a period of silence and the time between the start of the token and the end of the previous token that is not a silence token has length more than T sec, otherwise the token is labeled I.

Application of this leads to two types of errors. Intra-utterance silences occur within the same dialogue act as well as between two adjacent dialogue acts of the same speaker (see for example Gail Jeffersons [Jefferson, 1989] study on intra-utterance pauses.). On the other hand, applying the simple rule misses the situations where a speaker produces two utterances, each of which is a DA segment, but without a noticeable period of silence in between.

We calculated the performance of this rule for several values of T.

The following statistics are based on an analysis of a corpus of 152 AMI project design meetings, each with 4 participants. We used the timing of the words in the word layer that were computed using forced alignment. The total number of words is 690.764 (not including word elements that are gaps, or punctuations, but including vocal sounds and disfluency markers), the total number of dialogue acts is 112.702.

Of the total number of 687.934 adjacent token pairs within a dialogue act, 6.432 have a time gap in between that is larger than 0.0 sec. (less than 1 percent.) They can be divided into the following cases:

A) 4060. The pause is between the first and the second token of the dialogue act.

   A1) 3477 [$vocalsound||w$] (w includes punctuations)

B) 183. The pause is between the last en last but one token.

   B1) 46 [$w||w$] (w includes punctuations) of which 21 have the pattern [.$||w$] (typical example Okay . Yeah)

C) 2199. Others.

If we distribute these 6.432 pairs or tokens, over a number of bins according to the length t of the gap between the two tokens we get the following:

1) $0.0 < t \leq 1.0$ - 2804

T	NIST-B	DSER	REC	PREC	F
0.0	0,07(0,02)	0,63(0,08)	0,57(0,08)	0,91(0,04)	0,70(0,07)
0.5	0,08(0,02)	0,63(0,08)	0,56(0,08)	0,93(0,03)	0,70(0,07)
1.0	0,08(0,02)	0,65(0,07)	0,54(0,08)	0,94(0,03)	0,68(0,07)
1.5	0,09(0,02)	0,72(0,06)	0,47(0,07)	0,96(0,02)	0,62(0,06)
2.0	0,10(0,02)	0,76(0,05)	0,41(0,06)	0,98(0,02)	0,58(0,06)

Table 37: Error metrics of the Simple Segmentation Method for various values of T. $N = 121$ meetings; mean and st.dev. between brackets. Columns contain: input parameter time limit T, nist-boundary error metrics, DA Segmentation Error metrics (segment level), recall, precision, and f-measure for Boundary identification (word level).

2) $1.0 < t \leq 2.0$ - 2573

3) $2.0 < t \leq 3.0$ - 627

4) $3.0 < t \leq 4.0$ - 197

5) $4.0 < t \leq 5.0$ - 88

6) $t > 5.0$ - 143 instances.

Although intra dialogue act segment pauses are not very frequent (a little less than one percent of the total of all token pairs within a dialogue act) if we use the simple decision rule with $T = 0$ then we would have 6055 ”false alarms”, where we decide that a new segment starts where it is not. On the other hand, we would miss all boundaries where there is no time lag between two adjacent dialogue acts of the same speaker. This is true in 48.263 cases (out of a total of 112.162 adjacent dialogue pairs).

This simple method for segmentation has a precision of 91.65 and a recall of 56.91. Table 37 shows error and performance metrics for this segmentation method for various values of T. An F-optimal value is obtained for $T = 0$ sec ($F_1 = 70.22$).

Note that this method only uses information about the time gap between two adjacent words of a speaker. No information about the talking of the other conversational partners is used, and no information about the words themselves either.[16]

Table 37 shows that precision is quite high; the problem with the simple strategy is the low recall. In order to improve recall we have to learn when new segments start when the time gap with the previous dialogue act is small, less than or equal to T seconds. For $T = 0.0$ we analysed what words start a new DA segment. In 18.097 pairs of subsequent DA segment with no time between the segments, the most frequent words that start the second DA segment are shown in table 38. The listed words occur at least 100 times as first words of the segment and count altogether for 14.090 of the total of 18.097 pairs (that is 78%).

To complete the picture we need to see how often these words occur within a DA segment. For example ”and” occurs many times as a noun phrase coordinator (”fruit and

[16]Punctuations (elements of the word layers with punct=”true”; these are comma, end of sentence markers) all have duration zero (start and end time are equal). They have been been filtered out.

and	1927	2927	okay	398	627	i'm	166	426
so	1551	1343	it	290	4324	what	159	1279
i	1349	3534	the	266	9463	well	158	633
um	1291	2324	or	257	981	oh	140	139
but	945	469	like	243	2218	this	139	1458
yeah	677	1578	because	241	170	right	132	460
vocalsound	453	2556	if	225	1284	maybe	128	495
you	432	4051	that	223	4196	just	118	1638
we	423	3833	'cause	214	118	is	112	2417
it's	420	1585	that's	202	963	no	107	363
uh	409	2943	which	195	291	do	100	1077

Table 38: Most frequent start words of DA segments that follow a previous DA segment of the same speaker without a pause in between. These 33 words make up 78% of the total of 18097 instances of intra-turn follow up DA segments. The first column following the words give the number of occurrences of the word at the start of a new DA segment immediately following a DA segment of the same speaker. The second column after the words shows the number of occurrences of that word within a dialogue act. (total nr of DA segments: 39.561).

vegetables") in which cases it is not the start of a new DA segment. The table shows that most words do occur so often within a DA segment that they can't be considered by itself as a cue for a DA segment boundary. Therefore we need more information about the context of the word. It lies at hand to use a PoS tagger or NP chunker to distinguish for example the "and"-s joining two verb phrases from the "and"-s joining two noun phrases, since the former are more likely to indicate a DA segment boundary than the latter. PoS tags and n-grams of PoS tags are being used often for segmentation and DA classification, but note they often assume that you already have segments Certainly taggers trained and developed for written news paper text are biased to sentential structures, and may not be particularly usefull for tagging spoken text. For characteristics on the large Corpus Spoken Dutch (Corpus Gesproken Nederlands) and the performances of PoS taggers trained on the CGN we refer to [Stegeman et al., 2007].

If we use an off the shelf PoS tagger trained on written text we may not get the information that we were looking for. Some corpus sentences containing the word "and" and PoS tagged by the Stanford tagger with the Penn Treebank tagset are shown in Figure 16. We see that the PoS tags of the words before and after "and" do not tell us the NP joining occurrences from the ones that join two VPs. Thus, to make sense we must use a PoS tagger that is better tuned for DA segmentation of speech. Previous research (see [Stolcke and Shriberg, 1996]) has also indicated that the inclusion of some frequently occurring cue words and phrases as tags (examples are words like "and", "yeah", "okay" and phrases like "i'm") lead to better results, than when using PoS tags only.

Sometimes annotators consider words as to belong to one DA segment even when there is a considerable time between the two words. In a part of the AMI scenario based meeting corpus[17] 110 instances occur of intra DA time gaps of more than 5 seconds between two

[17]including 633964 words in 103044 dialogue act segments.

”I’m/NNP **Laura/NNP and/CC I’m/NNP** the/DT project/NN manager/NN ./.”
”And/CC I’m/NNP **Andrew/NNP and/CC I’m/NNP** uh/UH our/PRP$ marketing/NN”
”Um/NNP I’m/NNP **Craig/NNP and/CC I’m/NNP** User/NNP Interface/NNP ./.”
”so/RB we’re/JJ designing/VBG a/DT new/JJ remote/JJ **control/NN and/CC um/NN** [disfmarker]/NNS”
”So/RB that’s/VBZ David/NNP ,/, **Andrew/NNP and/CC Craig/NNP** ,/, isn’t/VBD it/PRP ?/. ”
”**And/CC you/PRP** all/DT arrived/VBD on/IN time/NN ./. ”

Figure 16: Some DA segments from the hand annotated ami corpus of scenario based meetings PoS tagged with the Stanford tagger using the Penn Treebank tag set. In bold occurrences of “and” in context.

duration	before gap	after gap
5.2	And we’ll see if we can	unscrew this first .
5.4	Okay , well , you got it’s a s	It’s a squirrel ,
9.4	Mm .	Okay .
6.9	Mm .	Right .
6.0	Right , so , seems to me that the thing that I have to do is	is quickly find that uh
9.5	’Kay .	So um
7.0	So .	So
6.0	Mm-hmm .	Okay .
6.6	Oh wrong one .	Uh .
8.4	Okay . Okay . Okay .	Okay ,
50.2	Oh .	Mm .
6.1	Um .	Yeah .
6.6	And then evaluation itself .	Uh .
7.8	Mm-hmm .	If you could uh
5.1	One , t	Seven , eight , oh . Fourth .
8.5	Mm .	Uh
12.0	Mm .	Uh .
10.1	’Kay .	So
5.5	Um See .	Um yeah .
7.4	So we make one for the volume , one for the channel .	Plus scroll .
11.6	How do we call ?	Evaluation criteria .
6.4	The pen ?	No .
6.4	Yeah , that’s the one .	Well , five .
12.9	Aye .	Yeah . Okay .
5.3	Aye . Yeah .	Okay .

Table 39: All 25 occurrences of intra DA gaps of length more than 5 secs between two adjacent *words* in the DA annotations of the AMI scenario based meeting corpus.

adjacent word elements.[18] These word elements contain besides words, also some non-verbal types of elements: punctuations, disfluency markers and vocal sound markers (for lip smacks, coughs and laughters). Of these the vocal sounds and the disfluency markers have positive time durations. Table 39 show the 25 instances of intra DA gaps between two subsequent words, where the gap was more than 5.0 seconds. In the other instances of gaps vocal sounds are involved; either before *or* after, or before *and* after the time gap. We can distinguish the following cases (where VS is vocal sound):

- VS;VS - 22, contain mostly no words; they are all very short feedback utterances, like ”Okay”, ”Oh my God.”, ”MASA?”.
- VS;Word - 43, in all instances (but one) the vocal sound is at start of segment; thus word is the first word of the actual verbal utterance (of various types).
- Word; VS - 10, in all instances (but one) the vocal sound is the last element of the DA segment. The exception is the segment: ”So uh [VS]. So”.

Notice that in almost all of the cases in the table, splitting up the segment into two DA segments would be a sensible choice as well.

The performance of the Simple Segmentation Rule suggests a method for segmentation and classification that combines this rule with a second phase that uses lexical and syntactical knowledge to increase recall by detecting the intra-turn segment boundaries.

4.5.5 Common Segments in DA Annotations

One of the AMI scenario based meetings (IS1003d) was annotated by four annotators. Reliability analysis is based on these four annotators. Two annotators annotated this meeting twice, with a period of time in between.

The table 40 shows the *percentage agreement* on segmentation between 15 pairs of annotators. The left column contains identifiers for annotators, and the number of segments they identified.

	(0)	(1)	(2)	(3)	(4)	(5)
(0)(1504)		0.759	0.752	0.735	0.723	0.690
(1)(1276)			0.840	0.703	0.824	**0.736**
(2)(1270)				0.685	**0.852**	0.722
(3)(1563)					0.672	0.646
(4)(1234)						0.697
(5)(1287)						

Table 40: Percentage Pairwise Agreement on Dialogue Act Segmentation. Intra-annotator agreement are in bold.

[18] begin and end times of words are computed by forced word alignment between manual transcription and audio signal.

	1	2	3	4	5
1	767(66,41%)	214(18,53%)	52(4,50%)	10(0,87%)	0(0,00%)
2	62(5,37%)	33(2,86%)	5(0,43%)	3(0,26%)	0(0,00%)
3	3(0,26%)	4(0,35%)	1(0,09%)	0(0,00%)	1(0,09%)
4	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)
5	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)

Table 41: Comparison of segmentation between two annotators. Number of common segments is 1155. Max segment length :5. The two annotators agreed on 767 segments. There were 214 instances where annotator 1 (row) had 1 segment that annotator 2 (columns) had split in 2 segments. The table shows that annotator 2 identified much more segments than annotator 1.

Usually, dialogue act labeling agreement is measured on the set of agreed segments. This gives us only a part of a picture of the disagreements in the annotations of different annotators.

Common segments One way to get a more detailed picture of the segmentation variations is to compute common segments.
We give an example for two annotators. Suppose annotators A and B segmented the sequence of words w as follows:

```
       a1   a2    a3     a4      a5
A: |-----|-----|----| |----| |-------|

B: |----|------|----| |----   ---|-|-|
      b1    b2    b3     b4      b5 b6
```

Then there are three common segments for this fragment: $< a1a2, b1b2 >$, $< a3, b3 >$ and $< a4a5, b4b5b6 >$.
Only the second common segment consists of length one lists of segments. The two annotators only agree on this segment, and the standard procedure for agreement analysis would consider labeling agreement between annotators on these common segments only.

A common segment of a sequence of tokens for a number of annotators is a set that contains for each annotator a shortest list of consecutive segments that stretch over the same sequence of tokens. Thus, when all elements in a common segment have length 1 then all annotators agreed that the stretch covered by the common segment is a segment.

Table 41 shows that about 83% of the common segments are those where either one or both annotators identified 1 or 2 DA segments.

We see the following types of segmentation dis-agreements in those cases where one annotator sees 1, the other 2 segments.

- Split of Stall.(50%) $X/StallX||Y$ and 30% $X/StallX$, i.e. both annotators agree on the type of the segment. Examples of words that are split of as Stalls: ”so”, ”yeah”, ”okay”, ”uh”.

DA Type	Freq1	Freq2	Agreed	CRF
INF	28.3	19.5	38	19.0
ASS	18.6	19.8	27	18.4
SUG	7.9	6.0	12	4.7
ELI	6.3	8.0	17	4.2
COM	1.9	3.1	2	2.3
BEP	1.9	3.5	2	2.1
FRG	14.0	10.0	23	17.8
STL	6.8	3.0	2	4.0
BCK	11.0	23.0	52	24.2
OTH	2.0	3.7	2	2.2
OFR	-	-	0	1.0

Table 42: Dialogue act types and frequencies of their occurrence in the corpus (Freq1). The third column (Freq2) shows the percentage of DA types in the subset of segments that all 4 annotators of meeting IS1003d agreed on. The column Agreed contains the number of DA segments that all 4 annotators tagged with the DA type of the row. Thus out of 430 DA segments 38 were labeled as INF by all four annotators. 178 segments were equally labeled by all annotators. (For column CRF: see section 4.5.7.)

- Split of Fragment. (28%), either with a leading Fragment or a Fragment at the end: $X/(X||Y)F$ or $X/F(X||Y)$. Fragments at the end indicate often interrupted speech. But also tag questions: "Three , right?", are sometimes split up. Leading fragment often indicate false starts, or repetitions.

- A segment labeled with DA type X by one annotator is split in two both of the same DA type X by the other annotator (15%).

Most of the $2-2$ split dis-agreements are combinations of the $1-2$ and $2-1$ disagreements in cases that an utterances has potentially both in it. Such as "yeah so maybe if" that is split up as "yeah || so maybe if" and as "yeah so || maybe if". Or where one annotator has split off a part with a leading conjunctive "and" but didn't split of the fragment "so" at the end, and the other annotator did the other way around.

From this analysis of segmentation disagreements between human annotators of transcribed speech it is clear that the two most troublesome cases are (1) fragments of speech that aren't complete dialogue act segments, such as stalls, fragments, backchannels and disfluencies, such as false starts and repetitions, and (2) secondly the segmentation of talk that are conjunctions of more than one utterance.

Common segments are defined for any number of annotations. The common segments that have length 1 for each of the annotations, are those containing the segments that all annotators agreed on. In meeting IS1003d a total of 430 segments were commonly identified by each of the 4 annotators.[19]

Table 42 shows the distribution of DA types over these 430 segments.

[19]Krippendorf α is 0.57 on the DA type annotation on these agreed DA segments.

In these 430 segments the following words occur most frequently.

- Yeah - initial (89), of which 83 just yeah.
- And. 27 contain ”and”, with 19 initial.
- So - 29 of which 25 initial ”so”, almost all as the only word, labeled as FRAG or STL.
- I - 35 contain ”I”, 22 of them start with ”I”.
- Okay - 37, of which 36 only consist of the word ”Okay”.

Although these words are among the most frequent in the whole corpus, they are clearly more frequent in the DA segments that annotators agree on. It is also clear from the distribution of DA types used that this set is not a good representation of the whole corpus. This is an extra reason to be carefull in jumping to conclusions from a statistical DA labeling reliability analysis, if we do this (as standard practice is) on a subset of those units that all annotators identified as a DA segment.

4.5.6 Confusion between DA classes

If we look at pairs of annotations, we see that the most frequent confusions in the dialogue annotations are the following.

- Assess - confused with Inform (systematic, that is: one annotator has preference over one class over the other), with Backchannels (where 80% of the confused items are ”Yeah”), and Be-positive.
- Backchannels - confused with assess and comment-about-understanding.
- Be-positive, is confused with assessments.
- Elicit-inform, confused with elicit-assessment.
- Informs, confused with assess, and with suggest.
- Stall, confused with fragment.
- Suggest, is confused with inform and assess.

Table 43 shows a confusion table of one pair of annotators involved in the annotations.

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
(0)ASS	**75.0**	11.0	5.0	10.0	1.0	1.0		7.0	38.0		1.0	6.0	2.0
(1)BCK	20.0	**112.0**		6.0				3.0	5.0		6.0	9.0	
(2)Be-P	3.0	1.0	**17.0**						4.0				2.0
(3)CAU	1.0		1.0	**5.0**					2.0		1.0		
(4)EL-A	1.0		1.0	4.0	**13.0**	4.0		1.0	2.0				3.0
(5)EL-I	1.0			1.0		**15.0**			6.0				
(6)EL-O				1.0					3.0				1.0
(7)FRG		3.0						**114.0**	2.0		35.0	3.0	1.0
(8)INF	5.0		2.0	2.0	1.0	4.0		4.0	**72.0**		3.0		
(9)OFF									1.0				
(10)OTH		1.0	1.0						2.0		**5.0**	1.0	
(11)STL				1.0				5.0				**21.0**	
(12)SUG			4.0					2.0	8.0	1.0			**22.0**

Table 43: Confusion Matrix of two annotations of DA types. Meeting: IS1003d. Krippendorff α: 0.57.

Dialogue act agreement The tables in Figure 17 and in Figure 18 contain α-values (using Krippendorfs coincidence-matrix) of inter-annotator agreement of Dialogue Act labeling. Different rows contain values for various groups of annotators.[20] The second column (*N*) contains the number of agreed dialogue act segments; this is the number of items that were considered in the classification. The other columns vary in sets of dialogue act labels considered:

set A is the original dialogue act label set

set B is the original set, but *Stall, Fragment*, and *Backchannel* are merged into one class.

set C is the original set, but *Stall, Fragment, Backchannel*, as well as *Other* are merged into one class.

set D is the original dialogue act label set, but all *Elicits* in one class, and *Be-Positive* and *Be-Negative* in one class.

set E as set B but with the merging of *Elicits* and the merging of *Be-Positive* and *Be-Negative* as in D.

set F as set C, but with the merging of *Elicits* and the merging of *Be-Positive* and *Be-Negative* as in D and E.

Figures 17 and 18 indicate that merging classes according to C and F give slightly better kappa statistics (Krippendorfs α statistics is actually used) than in other cases, but differences are small. The kappa values at least warn us to be carefull in both the interpretation of the label that a classifier trained on the annotated data assigns to an event, as well as in the interpretation of the evaluations of the classifier in terms of percent correct on a test set.

[20] legenda: a - vka; b - s95; c - dha; d - mar.

		set A	set B	set C
Group	**N**	α	α	α
a-b-c-d	430	0.572	0.578	0.601
a-b-c	491	0.579	0.581	0.595
a-b	693	0.610	0.613	0.626
a-c	731	0.570	0.574	0.612
b-c	723	0.548	0.551	0.556

Figure 17: Dialogue Act Type Agreement for groups of annotators. Alpha values for the dialogue act label sets A, B and C (see main text).

		set D	set E	set F
Group	**N**	α	α	α
a-b-c-d	430	0.578	0.585	0.608
a-b-c	491	0.582	0.585	0.599
a-b	693	0.613	0.615	0.629
a-c	731	0.575	0.553	0.618
b-c	723	0.560	0.565	0.570

Figure 18: Dialogue Act Type Agreement for groups of annotators. Alpha values for dialogue act labels of sets D, E and F (see main text).

4.5.7 Experiments with two sequence classifiers

In this section we present results with the applications of two different sequence classification methods on the problem of joint DA segmentation and classification. One method uses classical HMMs, the second method uses Conditional Random Fields.

The input of the classifiers is a sequence of words (tokens) of one single speaker. The output is a sequence of tagged words, where tags are combinations of two tag types. The first type is a boundary tag: either B (when the word starts a new DA segment) or I if the word is internal word of a DA segment. The second tag type is the dialogue act type. Since there are 15 DA types there are 30 combined tags in total.

The training data of the classifiers consists of a set of tagged token sequences. These token sequences are build from the hand annotated DA segments in the training corpus. The start word of a DA segment of type INF (for example) is tagged $B-INF$, all other words with $I-INF$. The length of the sequences of words is determined by the parameter T, the pause duration between two subsequent words of the same speaker. A new sequence starts with a word whenever the time between start time of the word and the end time of the last word before the word is greater than T seconds[21].

When we test the classifiers we feed it with sequences of words that are split off in the same way: sequence internal pauses last less than or equal to T seconds; the time lag between two adjacent sequences is more than T seconds. Thus we use the simple segmentation rule for splitting of sequences of words.

[21]Note that we used start and end times of the word elements computed by forced alignment.

Exp	ACC-A	ACC-B	NIST-E	DSER	REC	PREC	F
HMM1	51,51	91,81	53,10	**57,19**	**65,66**	77,77	**71,20**
HMM2	50,46	**92,17**	**50,79**	62,44	57,19	**87,76**	69,25
HMM3	**51,59**	91,95	52,21	58,94	62,28	81,13	70,47

Table 44: Results of Segmentation and Classification with trigram HMM models for the three HMM experiments. ACC-A is overall accuracy, i.e the percentage of correctly tagged words, including the dialogue act type. ACC-B is accuracy on B and I tag only; NIST-E is NIST Error metrics on boundary detection; last three columns gives recall, precision and F measure of classifying words as boundary or DA internal.

The first method using HMM is based on a trigram hidden event language model (the boundary tag indicates the hidden DA segment boundary). For training and testing we used Hammer, a java implementation of a toolkit for learning taggers and chunkers based on Hidden Markov Models.[22] For the second method using Conditional Random Fields we used Tako Kudoh's CRF++ package.[23]) [Lendvai and Geertzen, 2007b] present a similar chunking method for intra-turn segmentation and labeling where tokens are basic units instead of DA segments. They compare CRF with a memory-based tagger.

Experiments using Hidden Markov Models In the experiments with the HMM model we used the standard AMI train-test split for DA classification. If we train and test on the DA segments themselves the HMM trigram tagger shows an accuracy of 61%, with a DER of 0.40. Thus DA classification is about 60% correct both on DA level and on the word level. If we train on a larger set of (presegmented) DAs (103.044 DAs with 633.964 words) and tested 9.658 DAs with 56.800 words, the overall tag accuracy raised to 66.39% correct (on the word level). This improvement is partly caused by the fact that the number of unknown words is lower.

The following experiments with the HMM tagger have been performed:

HMM1 For the training as well as for test data the words are split up if there is more than 0.0 seconds in between two subsequent words (tokens).

HMM2 The HMM trigram model was trained on word sequences that are the correctly segmented DA segments in the training corpus.

HMM3 The HMM is trained on the combined training corpora of HMM1 and HMM2.

Table 44 shows the results of the three experiments with the HMM method.

It depends on the metric chosen which of the three training methods gives the better results. Training on the gap split data gives best results in terms of F-measure as well as in terms of Dialogue Act Segmentation (lowest DSER).

[22] Hammer was developed by Luite Stegeman at HMI-UT and is freely available.
[23] see: www.crf.sourceforge.org

```
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-2,0]/%x[-1,0]/%x[0,0]
U06:%x[-1,0]/%x[0,0]/%x[1,0]
U07:%x[0,0]/%x[1,0]/%x[2,0]
U08:%x[-1,0]/%x[0,0]
U09:%x[0,0]/%x[1,0]
```

Figure 19: The unigram feature templates used for the CRF DA segmentation and classification.

Table 45 reports recall, precision and F values for the two experiments HMM1 and HMM2 on the prediction of segment boundaries, for the most frequent start words of segments following a segment without a silence gap.

The tables shows that for some connectives like "because", "'cause" and "but" a drop in F-measure occurs when the model is not trained on the time segmented data, but only on correct DA segments. For connectives like "and" and "or" the drop is less dramatic, while for most other words the conditions of experiment HMM2 show the same or slightly improved F-values over those obtained in HMM1. It is not clear what causes these differences, but it indicates that the classifiers performances using different training methods varies significantly with the types of lexical items involved.

Experiments using Conditional Random Fields We used the CRF++ package with the feature templates given in table 19, i.e. a context of upto 5 words (the current word, two words before and two words following the current word) is used in the features. The bigram model of class labels is used.

Table 46 shows the results of the experiments with the CRF method. The CRF "sentences" for training and testing were obtained by starting a new sentences when the time gap between a word and the previous word is more than T seconds. The following experiments were performed.

CRF1 $T = 1.0.$, i.e. CRF sentence starts when gap between words is more than 1.0 sec. Number of train sentences: 3083; train size: DAs:6414; words:44285; test size: DAs:6170; words: 39.488. Parameter settings: $Freq = 3$, $\eta = 0,0001$, $C = 4.0$.

CRF2 $T = 1.0$. Train and test sets as in experiment CRF2 but different parameter settings for training: $Freq = 5$, $\eta = 0,001$, $C = 4.0$.

CRF3 $T = 0.0$. Number of train sentences: 10.058. train size: DAs: 16.942; words: 107.018. The test corpus is the AMI standard test corpus. Parameter settings as in CRF2.

word	freq	REC1	PREC1	F1	REC2	PREC2	F2
and	2395	38,17	67,08	48,65	24,54	82,54	37,83
like	689	34,83	57,41	43,36	32,58	69,05	44,27
vocalsound	2401	91,00	92,12	91,56	88,84	96,33	92,43
which	330	24,05	52,78	33,04	17,72	82,35	29,17
yeah	2779	91,06	90,52	90,79	86,39	95,91	90,90
i	2022	54,39	69,40	60,98	40,64	87,15	55,43
that	1867	37,30	60,26	46,08	36,51	73,02	48,68
or	616	37,40	47,12	41,70	22,90	58,82	32,97
okay	1102	89,77	89,77	89,77	82,20	93,22	87,37
just	734	37,97	78,95	51,28	34,18	79,41	47,79
so	1522	58,50	77,08	66,52	34,39	92,31	50,11
what	446	58,02	63,51	60,65	44,44	78,26	56,69
but	866	77,76	72,54	75,06	21,07	88,73	34,05
'cause	148	92,98	78,52	85,14	16,67	82,61	27,74
oh	242	92,15	83,81	87,78	76,96	90,74	83,29
maybe	386	61,80	54,46	57,89	47,19	84,00	60,43
it's	890	45,69	68,39	54,78	38,36	85,58	52,98
that's	637	55,25	73,53	63,09	47,51	83,50	60,56
the	4797	32,13	54,62	40,46	31,22	62,73	41,69
i'm	196	67,74	76,83	72,00	45,16	89,36	60,00
right	330	81,21	92,41	86,45	78,18	97,73	86,87
no	428	80,45	89,54	84,75	75,94	95,73	84,70
we	1712	36,36	50,00	42,11	28,64	66,32	40,00
um	1561	55,46	67,48	60,89	45,71	87,46	60,04
you	2197	32,58	56,21	41,25	26,52	67,96	38,15
because	228	88,59	67,35	76,52	10,74	72,73	18,71
well	581	75,52	78,55	77,01	66,43	94,53	78,03
uh	3766	45,72	66,92	54,32	45,21	80,00	57,77
it	2103	35,93	63,16	45,80	32,93	75,34	45,83
do	441	46,81	70,97	56,41	55,32	78,79	65,00
if	668	34,11	61,97	44,00	27,91	83,72	41,86
is	1110	39,29	70,21	50,38	47,62	75,47	58,39
this	554	30,56	55,00	39,29	25,00	66,67	36,36

Table 45: Recall, precision and F values for the two experiments HMM1 and HMM2 with the HMM trigram tagger on the prediction of segment boundaries, for the most frequent start words of segments following a segment without a silence gap.

Exp	ACC-A	ACC-B	NIST-E	DSER	REC	PREC	F
CRF1	50,94	92,63	47,51	50,01	72,50	78,37	75,32
CRF2	50,32	92,57	47,85	51,28	73,35	77,58	75,40
CRF3	52,53	92.75	47,02	52,86	71,20	79,62	75,17
CRF4	55,80	93,64	41,24	47,85	75,17	82,08	78,47
CRF5	-	93,73	40,64	47,09	76,33	81,81	78,98
CRF6	-	93,47	42,36	47,91	75,91	80,62	78,19

Table 46: Results of segmentation and classification experiments with Conditional Random Fields.

CRF4 $T = 0.0$. Complete ami train corpus used. Number of train crf sentences: 50.402. The test corpus is the AMI standard test corpus. Parameter settings as in CRF2.

CRF5 $T = 0.0$. Segmentation only; i.e. only B and I tags are used. Standard ami train test split and parameters as in CRF4.

CRF6 $T = 1.0$. Segmentation only. Number of DA segments per CRF sentence: 1.75. Corpora and parameters as in CRF5.

Training times for CRF are large (hours, due to large features sets; CRF3 uses 538.000 features; CRF4 more than 2 million features.) For $T = 1.0$ the mean number of DAs per "CRF sentence" is 1.75, for $T = 0$ this equals 1.60. Note that CRF sentence boundaries are not necessarily DA segment boundaries; CRF sentences boundaries are based on a pause between words.

Conclusion Table 47 shows the segmentation performance statistics for the CRF4 experiment on the sets of occurrences of most frequent start words of follow up DAs within a turn. If we compare the numbers in this table with Table 45 for the HMM experiments, we see that the CRF performs significantly better on most of the words. Overall, the CRF performs better than the generative method using HMMs. Even with a fraction of the training data, the CRF bigram model performs better than the HMM trigram model. Table 48 shows the main results on DA segmentation and classification. The first column gives Dialogue Error Rates which represent the percentages of correctly segmented and classified DAs. This is computed by counting a correct DA segment as correctly recognized if the first word of the segment has been labeled correctly. In the CRF results, the DA type tag of internal words of a segment always equal the DA type tag of the initial word. This doesn't hold for the HMM. The results improve results reported in [Dielmann and Renals, 2007b] that were obtained using a Dynamic Bayesian Network with the most restricted Factored Language Model trained on the AMI training data only.

Both sequential classifiers operate on words sequences, rather than on whole DA units. The results of CRF3 on the word level are shown in table 49. The last column of Table 42 (section 4.5.5) reports the distribution of the correct DA types of the correctly identified 7919 DA segments of CRF3, showing that shorter act types (backchannels and fragments) are relatively more frequent than longer act types.

word	freq	REC	PREC	F
and	2395	69,92	73,82	71,82
like	689	43,82	65,00	52,35
vocalsound	2401	91,34	94,54	92,91
which	330	51,90	43,16	47,13
yeah	2779	91,84	92,25	92,05
i	2022	66,67	74,75	70,48
that	1867	46,03	63,04	53,21
or	616	52,67	58,97	55,65
okay	1102	92,70	90,77	91,72
just	734	54,43	71,67	61,87
so	1522	87,35	84,27	85,78
what	446	55,56	67,16	60,81
but	866	82,61	77,92	80,19
'cause	148	91,23	81,25	85,95
oh	242	93,19	86,83	89,90
maybe	386	65,17	69,88	67,44
it's	890	56,90	69,84	62,71
that's	637	60,22	69,43	64,50
the	4797	39,82	62,41	48,62
i'm	196	76,34	78,02	77,17
right	330	80,61	91,72	85,81
no	428	86,47	91,63	88,97
we	1712	40,91	57,69	47,87
um	1561	77,98	78,38	78,18
you	2197	38,26	61,59	47,20
because	228	81,21	75,62	78,32
well	581	79,37	86,64	82,85
uh	3766	59,59	75,65	66,67
it	2103	48,50	68,07	56,64
do	441	61,70	69,05	65,17
if	668	39,53	59,30	47,44
is	1110	58,33	74,24	65,33
this	554	50,00	59,02	54,14

Table 47: Recall, precision and F values results of experiment CRF4 on the prediction of segment boundaries, for the most frequent start words of segments following a segment without a silence gap.

	DER	NIST-E	DSER	F
HMM	72	53	57	71
CRF	65	41	48	78
CRFs	-	40	47	79

Table 48: CRF and HMM results on DA segmentation and classification. Last row show results for segmentation only.

	Precision	**Recall**	$\mathbf{F}_{\beta=1}$
B-ASS	44.58%	37.03%	40.46
B-BAC	67.42%	83.07%	74.43
B-BEN	0.00%	0.00%	0.00
B-BEP	56.36%	38.78%	45.94
B-CAU	30.48%	10.19%	15.27
B-EAS	22.63%	16.17%	18.86
B-ECU	50.00%	25.00%	33.33
B-EIN	41.60%	24.67%	30.97
B-EOS	16.28%	13.46%	14.74
B-FRG	57.83%	67.03%	62.09
B-INF	44.00%	39.45%	41.60
B-OFR	41.40%	25.49%	31.55
B-OTH	35.29%	17.76%	23.63
B-STL	32.47%	28.13%	30.15
B-SUG	32.24%	28.23%	30.10
I-ASS	41.20%	39.61%	40.39
I-BAC	37.05%	32.19%	34.45
I-BEN	0.00%	0.00%	0.00
I-BEP	42.47%	24.23%	30.85
I-CAU	18.07%	8.85%	11.88
I-EAS	28.28%	23.44%	25.63
I-ECU	54.76%	32.86%	41.07
I-EIN	46.07%	31.39%	37.34
I-EOS	13.90%	22.25%	17.11
I-FRG	43.66%	46.24%	44.91
I-INF	68.96%	73.33%	71.08
I-OFR	55.08%	39.14%	45.76
I-OTH	22.92%	10.39%	14.30
I-STL	32.28%	27.87%	29.92
I-SUG	40.79%	45.52%	43.02
Overall	55.80%	55.80%	55.80

Table 49: CRF4 results per class label on word level tagging.

Table 50 shows the confusion table of the CRF4 DA classifier on the set of 8761 correct DA segments. The κ equals 0.61, comparable with the kappa of some pairs of human annotators on the same task. We see that the same types of confusions occur as we identified when comparing human annotators: Backchannels and Assessments, as well as Assessments and Informs.

Table 51 shows that about 88.3% of the common segments are those where either the machine classifier or the human annotator identified 1 or 2 DA segments. This is comparable with the figures in table 41, for two human annotators. Of the 1019 instances where the machine split up a segment into two segments, there are 88 where the second segments starts with "and". Manual inspection reveals that in 55 cases the split is correct (according the segmentation procedure in the manual). In the incorrect "and"-splits we see NP-NP connectives that are splits, and splits that are not allowed because they are embedded in conditional clauses, or in the second part a pronominal anaphora is used or an elliptic construction. A grammatical analyses is required to improve segmentation.

A post-hoc judgment about the classifiers output, as we do in telling real errors from "errors", that are due to disagreement between the classifier and the manual annotation in the test data, is also a judgment about the human annotation. But, this judgment is based on analyses and comparison of the different annotators. We see that in "similar" situations, an other annotator or even the same annotator annotated in a way that is in agreement with the machine classifier. If we base evaluation of the machine classier only on a comparison of the output of the classifier with the manual annotation in a test corpus, we may have a too negative picture of the classifier, since we add up false negatives and false positive due to disagreements with the annotation.

It came a bit as a surprise that when we train and test on the B and I tags only, thus doing only segmentation, we get better results on segmentation than with the joint CRF classifier. See table 46 for the results. Contrary to what we thought, information about DA types of words does not add to the recognition of segment boundaries, at least not with this method. The question whether we should do segmentation first and then classification on correct segments or do it jointly has not been settled. Our experiments seem to point at the first option.

How good is the CRF in segmentation compared with the simple rule. We computed the performance for $T = 0.0$ and for $T = 1.0$ sec. Table 52 shows the results separately for the set of words that start a CRF sentence and for the set of words that are "sentence" internal. The top line shows the overall result. We see that on the Start words the precision of the CRF classifier is only a bit better than the precision simple rule classifier. But the recall of CRF on this set is much better. The performance on start words is much better than on the internal words. Further, we see that there is hardly any difference in performance between the method where we chunk with $T = 0.0$ and the model with $T = 1.0$.

Not surprisingly, the correctly identified segments have a mean length, that is significantly less than the mean length of DA segments in train corpus, 4.4 vs. 6.5.

The performance of the CRF segmentation and classification is comparable with those of human annotators. This holds for DSER values on segmentation, as well as for agreement in DA classification on the subset of agreed segments. But improvements can probably be made using more features than just the lexical items that we used in these experiments.

	ASS	BAC	BEN	BEP	CAU	EAS	ECU	EIN	EOS	FRG	INF	OFR	OTH	STL	SUG	SUM
ASS	791	482	0	7	11	2	0	5	0	16	220	1	9	29	35	1608
BAC	112	1725	0	0	9	0	0	0	0	4	25	0	17	46	1	1939
BEN	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
BEP	24	0	0	121	7	1	0	1	1	4	31	0	2	0	8	200
CAU	59	76	0	0	24	2	0	4	0	3	18	0	5	13	0	204
EAS	14	8	0	0	0	28	0	20	1	3	17	1	1	3	11	107
ECU	2	1	0	0	0	0	3	1	0	0	1	0	1	0	0	9
EIN	18	4	0	2	1	28	0	101	2	12	74	2	2	0	14	260
EOS	0	0	0	0	0	2	0	4	6	2	5	0	0	0	4	23
FRG	22	16	0	1	2	2	0	8	0	1327	53	1	1	52	6	1491
INF	242	71	0	10	2	6	0	15	1	44	1252	14	10	3	116	1786
OFR	11	7	0	1	2	0	0	2	0	3	25	45	1	0	5	102
OTH	24	31	0	6	7	1	0	4	0	2	61	2	48	3	5	194
STL	35	48	0	0	2	0	0	0	0	65	8	0	2	232	1	393
SUG	26	0	0	5	1	4	0	11	1	10	158	2	6	1	219	444
SUM	1380	2469	0	153	68	76	3	176	12	1495	1949	68	105	382	425	8761

Table 50: Confusion matrix of CRF4 classifier on the set of correct DA segments. $\kappa = 0.61$.

	1	2	3	4	5	6
1	8761/(69,38%)	1019/(8,07%)	142/(1,12%)	17/(0,13%)	1/(0,01%)	0/(0,00%)
2	1205/(9,54%)	449/(3,56%)	106/(0,84%)	22/(0,17%)	7/(0,06%)	1/(0,01%)
3	335/(2,65%)	169/(1,34%)	43/(0,34%)	9/(0,07%)	5/(0,04%)	1/(0,01%)
4	98/(0,78%)	50/(0,40%)	34/(0,27%)	10/(0,08%)	1/(0,01%)	0/(0,00%)
5	35/(0,28%)	27/(0,21%)	14/(0,11%)	5/(0,04%)	2/(0,02%)	2/(0,02%)
6	16/(0,13%)	10/(0,08%)	5/(0,04%)	3/(0,02%)	2/(0,02%)	0/(0,00%)

Table 51: Comparison of segmentation between the human annotator and the CRF machine classifier. The total number of common segments is 12.628. Max segment length :13 (upto 6 are shown). The two annotators agreed on 8761 segments. There were 1019 instances where the human annotator (row) had 1 segment that the machine annotator (columns) had split in 2 segments.

	$T = 1.0$			$T = 0.0$		
	R	P	F	R	P	F
All	76	80	78	76	82	79
Start	99	94	97	99	92	96
Intern	48	59	53	46	62	53
Simple	54	94	68	57	91	70

Table 52: Recall, precision and F-values for the start and the internal words of "CRF sentences".

Prosodic and interactional features, as well as syntactic features (NP chunks, for example) are good candidates. A more in depth error analysis of the common segments may lead to the formulation of post segmentation steps that correct the segmentation errors before DA type classification is being performed.

4.5.8 Conclusion

We studied dialog act segmentation and classification and reported about the analyses of a human multi-layer annotated audio- and video recorded corpus of meeting conversations. We considered three question central to the methodology of developing machine classifiers based on annotated corpora.

- How good is the classifier? - questioning the evaluation methods for classifiers.
- How good can the classifier be? - questioning the intrinsic complexity of the task.
- How good should the classifier be? - questioning the kind of applications the classifier can be used for given its performance and error analysis.

We can conclude that:

- A strategy for real-time segmentation and classification of talk in multi-party interaction, that is worthwhile to be investigated further, consists of a combination of a simple decision rule based on timing alone and a sequential classification method for further segmentation and classification of chunks of the speech recognition output.
- Our experiments with Conditional Random Fields show that segmentation alone gives better segmentation results than when we use them to do segmentation and classification jointly.
- In evaluating machine classifiers we need to take into account errors due to noisy or erroneous training data, as well as ambiguities in the test data. Apart from any application, the output of a classifier can be really wrong or just different from the test data, but nevertheless correct.
- We have performed a rather detailed reliability analysis of the DA segmentation and classification task. This reveals that looking at kappa statistics only is certainly insufficient, but also that a DA classification agreement analysis as is usually done on the agreed segments gives a misleading picture of the complete data, since some classes are over represented, other are less frequent in this selection of segments.
- In general we plea for a critical evaluation of the current main stream methodology in developing machine classifiers for the classification of fuzzy higher level semantic phenomena (as addressing types and dialogue act types, emotion, etc.) that is based on statistical methods, feature selection, in the light of the reliability of the outcomes of these classifiers.
- To get some insight in "how good our classifier can be" we proposed to measure human annotations by the same metrics as we measure machine classifiers.

5 Summarization

5.1 Introduction

Much previous work on summarizing spontaneous spoken dialogues has involved the summarization of meetings that have been recorded and archived. On the AMIDA project, in contrast, we are analyzing meetings in as close to real-time as possible, so as to facilitate the actual conduct of the meeting for participants who may be attending in person or remotely. This poses some interesting challenges for the extractive summarization task. For example, we must decide whether or not to extract a candidate dialogue act before we have seen the global context, and we cannot use term-weighting that relies on overall term-frequency in the meeting.

In this chapter, we examine two aspects of extractive summarization within the AMIDA project. First, we describe a set of experiments regarding online summarization of the AMI corpus test set. Second, we describe the effect of using pause-based spurts as our extraction units rather than dialogue acts.

After that, we will describe how rich annotations can be used to generate indicative abstractive summaries. Finally, we present ongoing work that aims at the integration of both our extractive and the abstractive summarization research in form of hybrid multimedia summaries.

5.2 Towards Online Speech Summarization

5.2.1 Introduction

The majority of speech summarization research has focused on extracting the most informative dialogue acts from recorded, archived data. However, a potential use case for speech summarization in the meetings domain is to facilitate a meeting in progress by providing the participants - whether they are attending in-person or remotely - with an indication of the most important parts of the discussion so far. This requires being able to determine whether a dialogue act is extract-worthy before the global meeting context is available. This paper introduces a novel method for weighting dialogue acts using only very limited local context, and shows that high summary precision is possible even when information about the meeting as a whole is lacking. A new evaluation framework consisting of weighted precision, recall and f-score is detailed, and the novel online summarization method is shown to significantly increase recall and f-score compared with a method using no contextual information.

When applying speech summarization to the meetings domain, the goal of most research has been to extract and concatenate the most informative dialogue acts from an archived meeting in order to create a concise and informative summary of what transpired. Such summaries are analogous to the traditional manual minutes of a meeting, and are relevant to use cases such as a person wanting an overview of a meeting they missed, or a person wanting to review a meeting they attended, as a mental refresher. However, there are many use cases that go beyond the scenario of a user accessing an archived meeting. For example, someone might join a meeting halfway through and require a method of

catching up on the discussion without disturbing the other participants. A second example is a person who is remotely monitoring a meeting with the intention of joining the group discussion when a certain topic is broached. These use cases require the development of online summarization methods that classify dialogue acts based on a much more limited amount of data than previously relied upon.

This section introduces effective methods for scoring and extracting dialogue acts based on examining each candidate's immediate context. A method of *score-trading* is introduced and described wherein redundancy is reduced while informativeness is maximized, thereby significantly increasing weighted f-scores in our evaluation.

5.2.2 Weighting Dialogue Acts

This section describes three methods of scoring and extracting dialogue acts, the first of which relies on a simple term-score threshold, and the second two of which rely on a more complex score-trading system within the dialogue act's immediate context.

Residual IDF Previous work [Murray and Renals, 2007] has shown *ridf* to be a competitive term-weighting metric for summarization of spontaneous speech data. Our first method of extraction then is to simply sum *ridf* term-scores over each dialogue act and extract a given dialogue act if it exceeds a pre-determined threshold. Based on using various thresholds on a separate development set of meetings, a threshold of 3.0 is used for the experiments below. *ridf* scores were calculated using a collection of documents from the AMI, ICSI, MICASE and Broadcast News corpora, totalling 200 speech documents (AMI test set meetings were excluded).

Score-Trading The previously described method uses no knowledge of dialogue act context, and therefore does not address redundancy or importance relative to neighboring dialogue acts. A dialogue act is simply extracted if it scores above a given threshold. In contrast, the following two methods use a limited amount of context in order to maximize informativeness in a given region and to reduce redundancy, via a simple score-trading scheme.

For each dialogue act, we examine the ten preceding and ten subsequent dialogue acts. For each unique word in that 21-dialogue-act window, we total its overall score (its *ridf* score times its number of occurrences in that window) and reapportion that overall score according to the relative informativeness of the dialogue acts containing the term. For example, if the word 'scroll' has an *ridf* score of 1.2 and it occurs twice in that window, in two different dialogue acts, it has a total score of 2.4. If one of the dialogue acts containing the term has a dialogue act score of 5.0 and the other has a dialogue act score of 3.0, the overall term score is apportioned in favor of the former dialogue act, so that is receives a revised term score of 1.5 and the latter receives a revised term score of 0.9. As a result, the dialogue act score for the former has increased while it has decreased for the latter. This method of score-trading places the burden of carrying that term's information content onto the more generally informative dialogue acts, which also has the effect of reducing redundancy. Figure 20 illustrates the basic premise behind this scheme.

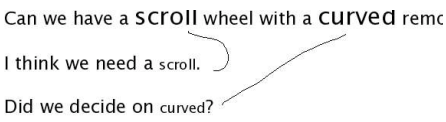


Figure 20: *Score-Trading Between Dialogue Acts*

More formally, the revised term-score for term t in dialogue act d is given by

$$Sc(t,d) = ridf(t) \cdot N(t) \cdot (\frac{Ascore(d)}{\sum_{i=1}^{M} Ascore(i)})$$

where $ridf(t)$ is the original *ridf* score for the term, $N(t)$ is the number of times that the term t appears in the context window, M is the number of dialogue acts in the window that are indexed by term t, and $Ascore(i)$ is the original score for a dialogue act i indexed by t, i.e. its total summed *ridf* scores.

A dialogue act's Bscore is then the sum of its revised term-scores. After deriving the Bscore score, the dialogue act in question is extracted if it satisfies the case

$$Bscore >= 3.0$$

The second score-trading method is similar to the first, but a dialogue act is extracted if it satisfies the formula

$$Bscore - (Ascore - Bscore) >= 3.0$$

where Ascore is the original score and Bscore is the adjusted score. The reasons motivating this latter method are twofold. First, a dialogue act's adjusted score (i.e. Bscore) may still be below the 3.0 threshold, but if it has increased significantly compared to the Ascore, that indicates its importance in the local context and we want to increase its chances of being extracted. Second, a dialogue act's adjusted score may be above 3.0 but it is well below its original Ascore, indicating that it has lost informativeness and may well be redundant in the local context. As a result, we want to reduce its chance of being extracted.

5.2.3 Experimental Setup

For this set of experiments we use the AMI meeting corpus test set, comprised of 20 meetings total.

The evaluation method is an extension of the *weighted precision* metric introduced by Murray et al [Murray et al., 2006], and relies on the many-to-many mapping between dialogue acts and abstract sentences described in the previous section. The work described in [Murray et al., 2006] involved the creation of very short summaries of 700-words, and the evaluation was therefore limited to weighted precision due to the very low recall scores of all approaches. In the present experiments, we extend the evaluation metric to

sys	man-prec	man-rec	man-fsc	asr-prec	asr-rec	asr-fsc
ridf	0.608	0.286	0.382	0.612	0.276	0.374
trade	0.611	0.295	0.391	0.610	0.285	0.383
tdiff	0.603	**0.305**	0.399	0.605	**0.295**	**0.392**

Table 53: Weighted Precision, Recall and F-Scores
ridf=DA extracted if Ascore >= 3.0, **trade**=DA extracted if Bscore >= 3.0, **tdiff**=DA extracted if Bscore - (Ascore-Bscore) >= 3.0

weighted precision, recall and f-score, as our new summaries tend to be much longer and are of varying lengths.

To calculate weighted precision, we count the number of times that each extractive summary dialogue act was linked by each annotator, averaging these scores to get a single dialogue act score, then averaging all of the dialogue acts scores in the summary to get the weighted precision score for the entire summary. To calculate weighted recall, the total number of links in our extractive summary is divided by the total number of links to the abstract as a whole. A difference between weighted precision and weighted recall is that weighted recall has a maximum score of 1, in the case that all linked dialogue acts are included in the extractive summary, whereas there is no theoretical maximum for weighted precision since annotators were able to link a given dialogue act as many times as they saw fit.

More formally, both weighted precision and recall share the same numerator

$$num = \sum_{d} L_s/N$$

where L_s is the number of links for a dialogue act d in the extractive summary, and N is the number of annotators.

Weighted precision is equal to

$$precision = num/D_s$$

where D_s is the number of dialogue acts in the extractive summary. Weighted recall is given by

$$recall = num/(L_t/N)$$

where L_t is the total number of links made between dialogue acts and abstract sentences by all annotators, and N is the number of annotators.

The f-score is calculated as

$$(2 * precison * recall)/(precision + recall)$$

The generated summaries range between 600 and 3000 words in length, as the meetings themselves greatly vary in length. Unlike summarization of archived meetings, here we do not specify a set summary length in advance since the length of the meeting is not known

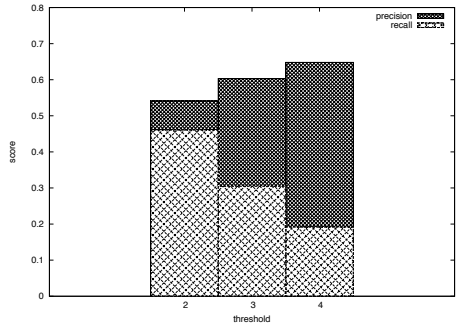


Figure 21: *Score-Trading at Multiple Thresholds*

beforehand. The resultant summaries could, of course, be revised to fit a particular length requirement once the meeting has finished, but here we simply decide whether or not to extract each dialogue act candidate without consideration of the summary length at that point.

5.2.4 Results

Table 53 presents the weighted precision, recall and f-scores for the three approaches described above. One of the most surprising results is that the weighted precision in general is not drastically lower than the scores found when creating very brief summaries of archived meetings. For example, in [Murray and Renals, 2007], creating 700-word summaries of the same test set using *ridf* yielded an average weighted precision of 0.66. All three online approaches presented here have average weighted precision around 0.61. This is particularly surprising and encouraging given that these summaries are on average much longer than 700 words.

The third approach, labeled **tdiff** in Table 53, is superior in terms of f-score on both manual and asr transcripts. *ridf* performs the worst on both sets of transcripts, and the second approach labeled **trade** is in-between. Significant results in the table are presented in boldface. The method **tdiff** achieves significantly higher recall than the other two methods on manual transcripts, and both recall and f-score are significantly higher on ASR (paired t-test, $p<0.05$). The most encouraging result of this third approach is that it is able to significantly increase recall without significantly reducing precision.

Having determined the effectiveness of the third approach, we subsequently run this score-trading method at multiple thresholds of 2.0, 3.0 and 4.0 to gauge the effect on weighted precision, recall and f-score. The results are displayed in Figure 21. A threshold between 2 and 3 results in a good balance between recall and precision, while a threshold of 4 results in drastically lower recall and only slightly higher precision.

The score-trading results reported so far stem from an implementation of the method that has an algorithmic delay of 10 dialogue acts. We are interested in what benefit, if any, could be gained by increasing the algorithmic delay and thereby increasing the amount of context used. The two score-trading approaches are therefore run fully offline, so that the context for each dialogue act is the entire meeting (the first approach, based simply on *ridf* results, is the same online versus offline since it does not use context). Because there

sys	man-prec	man-rec	man-fsc	asr-prec	asr-rec	asr-fsc
trade	0.599	0.291	0.386	0.608	0.291	0.388
tdiff	0.589	**0.306**	**0.398**	0.593	**0.304**	**0.398**

Table 54: Weighted Precision, Recall and F-Scores (Offline)
trade=DA extracted if Bscore >= 3.0, **tdiff**=DA extracted if Bscore - (Ascore-Bscore) >= 3.0

is a larger amount of score-trading when using all meeting dialogue acts for comparison, a given dialogue act would have to be very informative in order to have its overall Ascore increase. The expectation is that running this method offline would therefore result in higher precision and perhaps lower recall. Table 54 presents the weighted precision, recall and f-scores for the offline systems. The third approach, labeled **tdiff** in Table 54, is again superior to the second approach, labeled **trade**, with significant differences between the two in terms of recall and f-score on both manual and ASR transcripts. However, neither approach is significantly different when run offline versus online. The trend is for precision to be slightly lower when run offline and recall to be slightly higher, the opposite of what was expected.

5.2.5 Discussion

The results above show that the score-trading scheme is able to significantly increase recall and f-score with no significant decrease in precision. More specifically, it allows us to reject dialogue acts that may have scored high but were redundant compared with similar and more informative neighboring dialogue acts, and allows us to retrieve dialogue acts that may have scored below the threshold originally but subsequently had their scores adjusted based on local context.

In general, it is interesting that high precision is attained via methods that use either no context or only local context. As mentioned earlier, previous experiments on creating very concise summaries using global information about the meeting achieved weighted precision of only a few points higher. It turns out that restrictions such as the inability to create an overall ranking of dialogue acts in a meeting or to rely on term-frequency information are not severely detrimental to the ultimate results.

A related finding is that there is no benefit to running the score-trading methods completely offline, using the entirety of the meeting's dialogue acts as context. In fact, precision results were slightly better when examining only the limited context. It may be that dialogue acts sharing some of the same terms and existing within proximity to each other tend to be more similar than dialogue acts sharing some of the same terms but existing at various locations spread throughout the meeting. In that case, score-trading between ostensibly similar dialogue acts would not always be beneficial if the examined context is too large.

While the score-trading methods outperform the simple *ridf* threshold method, with the third summarization system performing the best, it would seem that the methods are complementary. Because the *ridf* method requires no contextual information, a dialogue act can be immediately extracted or rejected on a preliminary basis. Once the subsequent

context for a dialogue act becomes available, that decision can be revised based on score-trading. User feedback could provide a further source of input for such dynamic summary creation.

5.3 Summarization Without Dialogue Acts

5.3.1 Introduction

In the previous section, our summarization system relied on dialogue acts as input, using those segments as the units of extraction. In this section, we briefly consider the use of spurts rather than dialogue acts as our summary units. A spurt can simply be defined as a region where a meeting participant is speaking continuously, with boundaries determined by pause information. A primary benefit of using spurts rather than dialogue acts is that we can quickly segment the speech stream into meaningful units without time-consuming dialogue act segmentation. This is of particular importance for online summarization as described in the previous section. Spurt segmentation may also result in units of finer granularity than dialogue acts and allow us to more accurately pinpoint informative regions of the meeting.

5.3.2 Spurt Segmentation

In defining spurts, we rely entirely on pauses and filled pauses for determining the unit boundaries. This is in contrast to most work on dialogue act segmentation, where prosodic features along with n-gram language models are used for segmentation [Ang et al., 2005a, Dielmann and Renals, 2007b]. Taking speaker-segmented ASR output as our input, we place a spurt boundary at any location where the inter-word pause for a speaker is 400 ms or longer, or where there is a pause of at least 200 ms plus a filled pause such as "um," "uh," or "erm." Once we have segmented the speech stream of each speaker in the meeting, the final input to the summarization system is the list of spurts ordered so that they are monotonically increasing according to start-time.

5.3.3 Experimental Overview

These spurt-based experiments are performed on the AMI corpus test set, comprised of 20 meetings total.

Once we have the input format described above, summarization proceeds simply by scoring each spurt using the *su.idf* metric [Murray and Renals, 2007]. Each spurt's score is calculated as the sum of its constituent word scores. We then rank the spurts according to their scores and extract until we reach the length limit of 700 words.

Previously, we have relied on weighted precision/recall/f-score for our evaluation metrics, using multiple human extractive annotations of dialogue acts. Now that we're no longer using dialogue acts as our summary units, we have to rely on other evaluation metrics. For this purpose, we use the ROUGE-2 and ROUGE-SU4 n-gram metrics, which are normally the ROUGE metrics that best correlate with human evaluations [Lin, 2004].

Meet	ASR-Spurts	Human
ES2004a	0.02657	0.05105
ES2004b	0.01770	0.01735
ES2004c	0.03994	0.02675
ES2004d	0.01102	0.01362
ES2014a	0.06946	0.08037
ES2014b	0.03252	0.03518
ES2014c	0.06032	0.07938
ES2014d	0.05168	0.06133
IS1009a	0.10370	0.14720
IS1009b	0.02184	0.06278
IS1009c	0.03873	0.10256
IS1009d	0.06166	0.08995
TS3003a	0.04813	0.04558
TS3003b	0.07564	0.04234
TS3003c	0.06742	0.06541
TS3003d	0.04843	0.05155
TS3007a	0.08180	0.08254
TS3007b	0.01933	0.01591
TS3007c	0.04792	0.06069
TS3007d	0.02420	0.02417
AVERAGE	**0.047**	**0.058**

Table 55: ROUGE-2 Scores for Spurt Summarization and Human Summarization

For comparison, we include human summaries of the same length, 700 words, choosing one annotator at random for each meeting and extracting their most-linked dialogue acts until reaching the length limit. These human summaries are then also compared with human gold-standard abstracts using ROUGE.

5.3.4 Results

Table 55 lists the ROUGE-2 scores for the AMI test set meeting summaries, for both the automatic spurt-based approach described above and human-level performance. We find that according to ROUGE-2, not only does performance not decrease when using simple spurt segmentation instead of dialogue act segmentation, the scores are actually higher than the ROUGE-2 scores when using dialogue acts, averaging 0.047 compared with 0.041.

Table 56 lists the ROUGE-SU4 scores for the spurt-based summaries and the human summaries. The average for the spurt-based approach is 0.079, which again is better than the ROUGE-SU4 scores when using dialogue acts, which average 0.070. We also find that the average for the spurt-based method approaches human-level performance on this metric. On many meetings it is in fact superior to human performance.

5.3.5 Discussion

The reason that the spurt-based approach performs better than the dialogue-act based approach according to ROUGE seems to be that there is a finer level of granularity. For the AMI test set, there are on average nine fewer dialogue acts extracted for each meeting than spurts extracted. The spurts simply tend to be shorter, and so we can extract more of them. Furthermore, since our units are a finer granularity we can more easily separate the informative and non-informative portions of the transcript. For example, with dialogue

Meet	ASR-Spurts	Human
ES2004a	0.04255	0.06016
ES2004b	0.04845	0.05664
ES2004c	0.06145	0.07204
ES2004d	0.04263	0.04812
ES2014a	0.09303	0.10463
ES2014b	0.07475	0.07813
ES2014c	0.09769	0.10030
ES2014d	0.08080	0.07982
IS1009a	0.15810	0.15256
IS1009b	0.06884	0.08556
IS1009c	0.06422	0.11776
IS1009d	0.10325	0.12989
TS3003a	0.06924	0.05704
TS3003b	0.12962	0.07534
TS3003c	0.10156	0.10064
TS3003d	0.06387	0.07626
TS3007a	0.09938	0.09572
TS3007b	0.05804	0.05687
TS3007c	0.08521	0.07990
TS3007d	0.05244	0.05595
AVERAGE	**0.079**	**0.084**

Table 56: ROUGE-SU4 Scores for Spurt Summarization and Human Summarization

acts we might extract a very long dialogue act because it has several high-scoring words, but in fact there is only one part of the dialogue act that is particularly relevant and the remainder is simply included because it is one extraction unit.

Of course, one solution to this problem is to compress dialogue acts after extraction, and the first section of this chapter described one set of compression experiments. However, a certain amount of compression would be unnecessary if we began with a finer granularity for our extraction units. It is somewhat of a roundabout process to segment dialogue acts, extract the most informative ones, which tend to be longer units, and then compress them, compared with simply using finer extraction units to begin with. Compression is still very useful, especially when the informative portions of the extraction unit are spread throughout the unit with intervening uninformative words or phrases, but using spurts may decrease our need to carry out any further compression.

5.4 Indicative Abstractive Summaries

Extractive summarization of documents has been studied extensively over the last decades (s. [Mani and Maybury, 1999] for an overview), but faces additional challenges when applied to natural language dialogs. As opposed to carefully authored articles, spontaneous utterances are often ungrammatical and contain speech disfluencies ([Shriberg, 2001]). Moreover, free discussions are naturally less well structured, e. g., when people digress. For an automated system, additional difficulties arise from the limitations of current ASR systems, introducing recognition errors into all subsequent processing steps. [Zechner, 2001] and [Murray et al., 2005] show ways to cope with such issues.

Generative approaches, on the other hand, are based on an internal represenation of summary contents verbalized through NLG techniques (e. g. [Kan et al., 2001]). Such approaches have also been applied to natural discourse domains. For instance, [Reithinger et al., 2000] generate summaries of machine-translated phone conversations. However,

we are not aware of prior work attempting to generate abstracts of multi-party interaction.

5.4.1 Propositional Content

We have annotated a small subset of the AMI corpus ([McCowan et al., 2005a]) with categories from a domain ontology to represent the propositional content of speaker utterances. In addition, various annotations of these meetings are already available with the corpus, for instance, speech transcription, syntactic chunks, named entities, dialog acts, addressing, argumentative structure, hot spots, decision points and topics.

The AMIMATTER ontology which we created for the purpose of representing propositional content models the remote control design scenario in a formal ontology based on Dolce-Lite-Plus [Masolo et al., 2003]. Embedded in a comprehensive theory of representing situations and descriptions, it provides a taxonomy of relevant terms, ordered by an `IS-A` relation that expresses subsumption, or specialization. For instance, it contains information such as (remote_control IS-A technical_device) which expresses that the category remote_control is a sub-category of the category technical_device. Hence, a reasoner can infer that all remote controls (which technically would be considered *instances* of the category remote_control) are technical devices.

The AMIMATTER ontology covers over 20 different subdomains, with a total of 53,319 categories. 52,072 of those are extracted from WordNet [Fellbaum, 1998], the remaining 1,247 cover scenario-specific concepts and the Dolce-Lite-Plus upper model.

Three subdomains—physical objects, meeting-related categories and project-related categories—were used to annotate the discourse transcription.

The current system relies only on the annotation of relevant categories, ignoring relations within or beyond the dialog act segment boundaries[24].

Fig. 22 shows an example of such an annotation: three instances from the physical object subdomain were created (shown as boxes) and linked to the respective words in the source utterance above.

5.4.2 Summary Content Representation

We currently concentrate on three of the above annotation layers, topic labels, dialog acts and propositional content. For the pre-existing topic annotation, the recordings were split into larger segments and labeled with one of 24 topics matching typical activities in the remote control design scenario, e. g., "discussion" or "presentation of prototype(s)". These segments are used by our system as the basic structuring unit for the summaries. In most cases, the label can be used to verbalize the general subject of the topic segment, with the exception of the "other" label which is used for unknown topics.

In a similar practice, all participants' utterances in the manual transcript of the meeting discourse were segmented and labeled with dialog acts such as "inform", "suggest", etc. according to a scheme consisting of 15 distinguished dialog acts. However, our system

[24]More precisely, annotators were asked to identify those terms in a speaker utterance that belong to one of the three subdomains, identify the appropriate AMIMATTER category and create an instance of it, and connect the instance with the original word.

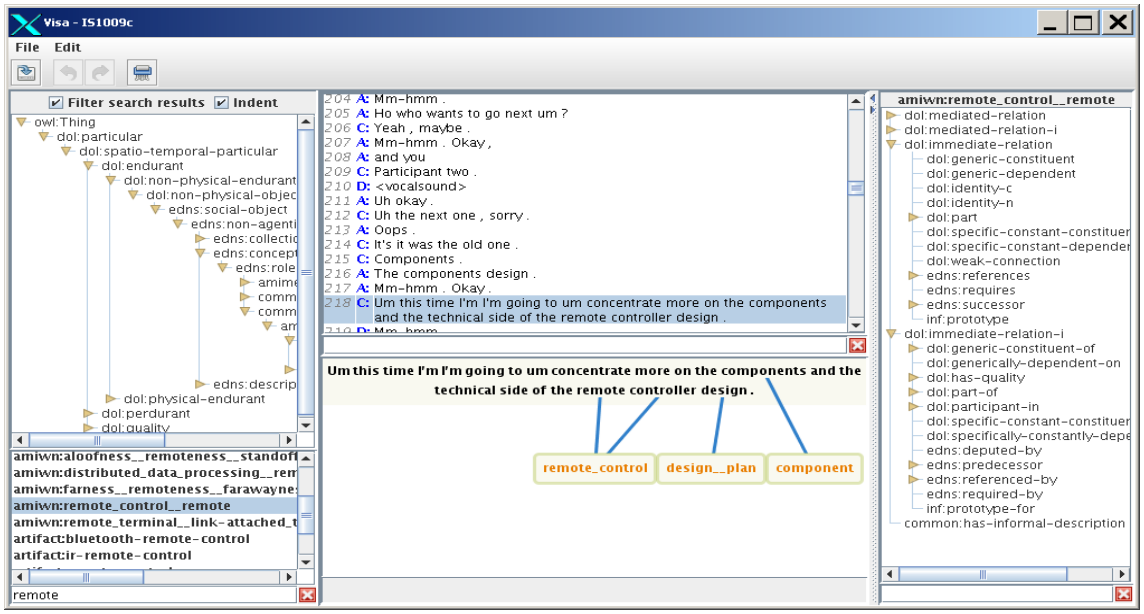


Figure 22: Example annotation of an utterance in meeting IS1009c in the AMI corpus. The outer sides display categories and relations of the AMIMATTER ontology in tree views, the center part contains the meeting transcript (top) and the annotation area (bottom).

currently discards the labels themselves, but uses the segments as a common unit for the propositional content annotation outlined in subsubsection 5.4.1. We perform a frequency analysis of all annotated ontology instances and select the three items that occur most often. We found this a useful heuristic, although it sometimes produces unexpected results (s. fig. 26: the term "beep" stems from an ontology category of the same name that was used to annotate a discussion about audio signals in the corpus).

5.4.3 Text Generation

The actual generation of the abstracts is done in a three-step pipeline:

1. Analysis of meeting annotation layers
2. Sentence planning
3. Surface realization

In the first step, information drawn from the annotation layers is transformed into expressions in a propositional logic-like formalism (figure 23). These assertions are used as a knowledge base by the sentence planner PREPLAN, a hierarchical, goal-driven planner [André, 1995]. In addition to the assertions, PREPLAN is provided with a library of plan operators, each of which encodes strategies how to reach a given goal. Figure 24 shows an example of such an operator which describes how to reach the goal "ShowSummary" as the result of solving three subgoals, one of which is an iteration over all topics. Here, the "with"-condition is matched against the knowledge base that was generated before.

PREPLAN successively finds matching plan-operators until all goals and subgoals are resolved. The outcome of this process is an XML-encoded description of instructions in a

```
(topic "t0")
(about "t0" "opening")
(content "t0" "introduction")
(content "t0" "project manager")
(after "t0" "t1")  ...
```

Figure 23: The input for the sentence planner: topic t0 which is the opening of the meeting occurs before topic t1 and contains the content items "introduction" and "project manager".

```
strategy: (ShowSummary)
subgoals: (WriteXMLHeader)
          (for-each ?t with (topic ?t)
             (ShowTopic ?t))
          (WriteXMLFooter)
```

Figure 24: A complex plan operator in PrePlan

logical form which is passed to the surface realizer, NipsGen [Engel, 2006], a template-based generator. NipsGen converts the semantic input into typed feature structures which are then transformed into a natural language utterance.

A derivation tree for the XTAG-grammar [XTAG Research Group, 2001] is created using transformation rules which are applied to the input structure (see figure 25 for a sample rule). The actual syntax tree is constructed using the derivation tree. The generation of the

```
$VP=VP(o:Introduction(has-topic:$T,
           has-agent:$A), not(lex:))
-> $VP(lex:introduce, sub:NP(o:$A),
       obj:NP(o:$T))
```

Figure 25: A NipsGen rule: the semantic concept 'Introduction()' is lexicalized with the verb 'introduce'. The values of the features 'has-topic' and 'has-agent' are realized as NP's in object and subject position, respectively.

correct morphological inflections is achieved by percolating the morphological features through the XTAG tree and looking up the correct inflections for all lexical leaves in the XTAG lexicon for English. Traversing the lexical leaves from left to right produces the natural language utterance.

5.5 Hybrid multimedia summaries

One of the most central applications of summaries is to save the reader time by sparing him to have to consult the original source document(s), yet giving him access to the information he is looking for. Coming from the document summarization tradition, summaries are classically presented in form of a text. But when the summary source is not a text itself, as in the case of the highly interactive environment of meetings, it is unclear whether written text is an optimal way to convey the summary. Given the richness of the data in

"The meeting was opened and the meeting group talked about the user interface, the remote control and the design. They debated the costs, the company and the project while discussing the project budget. The signal, the remote control and the beep were mentioned afterwards. They talked about meeting before closing the meeting."

Figure 26: Example of a meeting summary.

the AMI corpus, the possibility arises to condense information further by using mutliple media to encode different aspects of a summary.

Depending on the scenario, such an approach offers various advantages. For instance, in the case of meeting archives, one important task of summaries is helping the user accessing the archive to find the relevant meeting given a specific information need. A participant of a previous meeting might wish to consult the recordings to remind himself of a certain decision taken during the meeting, or a project member who missed one of the meeting might just get a quick overview of the topics discussed.

Also, especially in an *online* situation, where summaries need to be accessed while the meeting is taking place, the time factor might be crucial: it can not be expected that a participant who is in the middle of a meeting can spare a lot of his time reading a lengthy text.

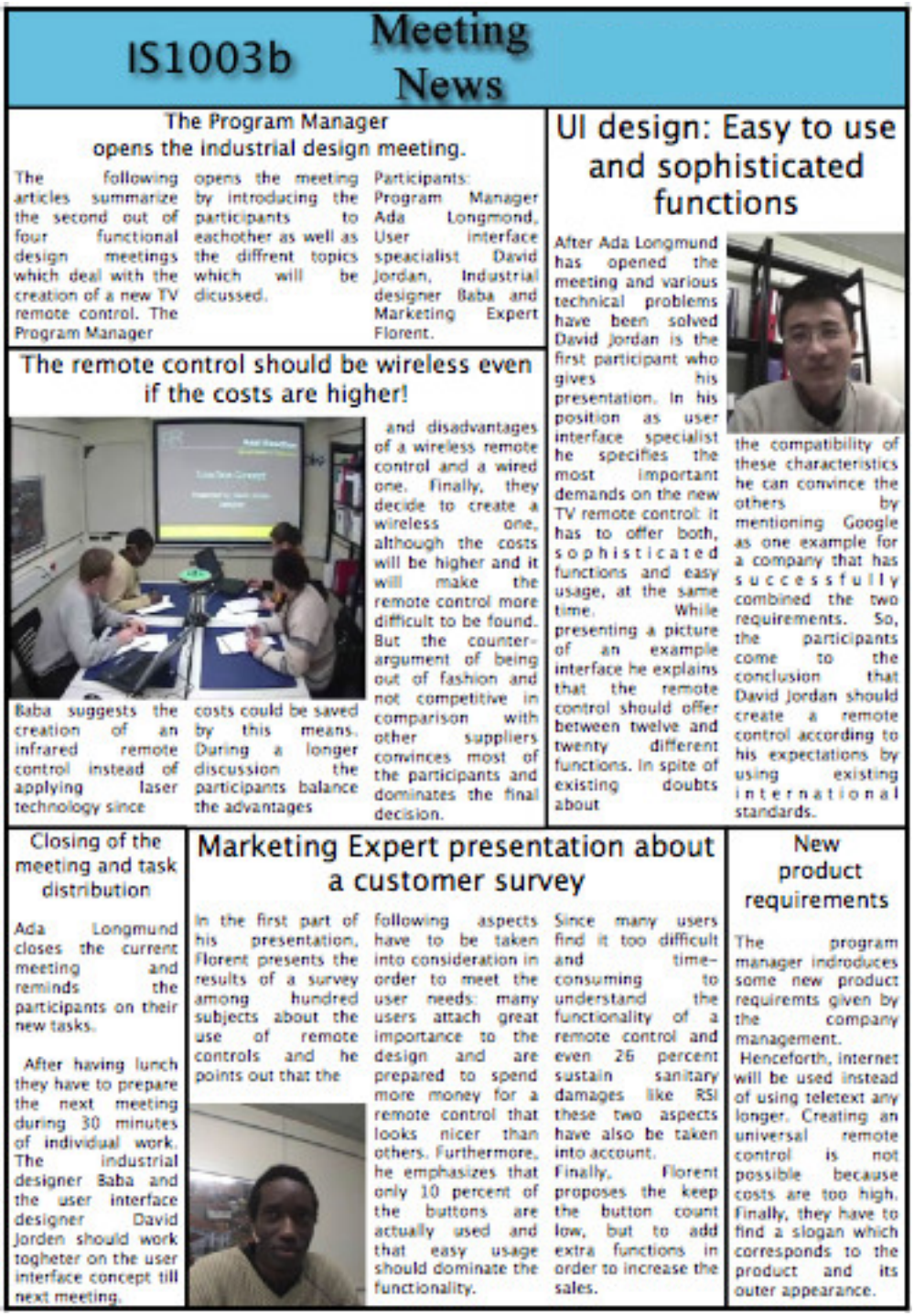
We have started work on multimedia summaries that combine text with pictures from the video signals. So far, we have concentrated on two different kinds of summaries: *result-oriented summaries* and *progress-oriented summaries*. By using automatic layout techniques we aim at generating fully automatic mixed-media documents that allow readers to understand the gist of a meeting at a single glance. To do so, we use widely spread document styles that are well suited for high information condensation.

In particular, we represent result-based summaries in a newspaper style and progress-based summaries in a comic-strip style. Both types of documents can be found in our everyday environment and hence users are experienced in comsuming them efficiently. Another novelty of this work, besides introducing a multimedia aspect, is to combine the previously separated research areas of extractive and abstractive summarization.

5.5.1 Layout Generation

Even for the same group of participants, any two meetings can differ in a vast number of aspects:

- meeting duration
- number of participants
- topics discussed
- dominance levels of different speakers
- discussion styles (presentations, open discussion, etc.)



IS1003b Meeting News

The Program Manager opens the industrial design meeting.

The following articles summarize the second out of four functional design meetings which deal with the creation of a new TV remote control. The Program Manager opens the meeting by introducing the participants to eachother as well as the diffrent topics which will be dicussed. Participants: Program Manager Ada Longmond, User interface speacialist David Jordan, Industrial designer Baba and Marketing Expert Florent.

UI design: Easy to use and sophisticated functions

After Ada Longmund has opened the meeting and various technical problems have been solved David Jordan is the first participant who gives his presentation. In his position as user interface specialist he specifies the most important demands on the new TV remote control: it has to offer both, sophisticated functions and easy usage, at the same time. While presenting a picture of an example interface he explains that the remote control should offer between twelve and twenty different functions. In spite of existing doubts about the compatibility of these characteristics he can convince the others by mentioning Google as one example for a company that has successfully combined the two requirements. So, the participants come to the conclusion that David Jordan should create a remote control according to his expectations by using international standards.

The remote control should be wireless even if the costs are higher!

Baba suggests the creation of an infrared remote control instead of applying laser technology since costs could be saved by this means. During a longer discussion the participants balance the advantages and disadvantages of a wireless remote control and a wired one. Finally, they decide to create a wireless one, although the costs will be higher and it will make the remote control more difficult to be found. But the counter-argument of being out of fashion and not competitive in comparison with other suppliers convinces most of the participants and dominates the final decision.

Closing of the meeting and task distribution

Ada Longmund closes the current meeting and reminds the participants on their new tasks.

After having lunch they have to prepare the next meeting during 30 minutes of individual work. The industrial designer Baba and the user interface designer David Jorden should work togheter on the user interface concept till next meeting.

Marketing Expert presentation about a customer survey

In the first part of his presentation, Florent presents the results of a survey among hundred subjects about the use of remote controls and he points out that the following aspects have to be taken into consideration in order to meet the user needs: many users attach great importance to the design and are prepared to spend more money for a remote control that looks nicer than others. Furthermore, he emphasizes that only 10 percent of the buttons are actually used and that easy usage should dominate the functionality. Since many users find it too difficult and time-consuming to understand the functionality of a remote control and even 26 percent sustain sanitary damages like RSI these two aspects have also be taken into account. Finally, Florent proposes the keep the button count low, but to add extra functions in order to increase the sales.

New product requirements

The program manager indroduces some new product requiremts given by the company management. Henceforth, internet will be used instead of using teletext any longer. Creating an universal remote control is not possible because costs are too high. Finally, they have to find a slogan which corresponds to the product and its outer appearance.

Figure 27: A generated newspaper-style summary of AMI meeting IS1003b

- ...

As a result, summaries of different meetings will convey different characteristics and hence multimedia presentation displaying these summaries may vary drastically for different meetings. In effect, this means that in order to generate such presentations automatically, we cannot expect to use just a single, pre-defined layout to match the diversity inherent in the different meeting summaries.

Constraint-based multimedia presentation systems have been proved to be an appropriate approach for situations that require high flexibility (s., e. g., [André et al., 1997]). In the following sections, we demonstrate that constraint-based layout generation is a suitable technique for the generation of multimedia summaries.

5.5.2 The Newspaper metaphor

Efficient access to the main results of a meeting can be one of the central goals a summary has to meet. We have found that rendering the most important in the style of a newspaper page, where one "article" summarized one specific topic of the underlying meeting, useful for the following reasons:[25]

1. Structuring summary results *per topic* facilitates concentrating on the most relevant information for a specific information need.

2. By arranging all topics on a single page, all information is accessible at the same time. Moreover, comparison between different outcomes of the meetings is made very simple.

3. The newspaper-typical style of displaying articles with a headline allows for even more efficient information compression. One can think of a newspaper page as a way to display a hierarchy where the first level contains only headlines, an optional second level article abstracts and the last level contains the actual article.

4. If useful, images and diagrams can easily be integrating without breaking the metaphor, as newspaper articles frequently contains pictures.

When looking at actual newspapers, we notice that the two basic techniques for a publisher to convey the expected relevance of an article to user are *placement on the page* and *size of the article*. The article which is considered most important for the current issue of the newspaper is usually presented in the middle of the upper half of the title page. Likewise, less important articles are moved to side positions and are typically assigned less space than more important ones.

In order to stay as close as possible to the familiarity offered by the newspaper metaphor, we aim at taking over this concept for the automatic generation of meeting summaries. To represent the relevance of a summary topic through position and size of an article, we represent the layout of a newspaper page as a weighted grid (s. 28). Each article is

[25] Here, we are only interested in the most important outcomes of a meeting. Hence we restrict ourselves to the generation of a single page, resembling the title page of a newspaper.

Title				
7	7	4	10	10
3	3	4	10	10
4	4	4	4	4
3	9	9	5	3
2	9	9	5	2

Figure 28: Weighted grid for a newspaper page

placed on top of this grid , i. e., each article will have boundaries that coincide with the outline of a rectangular set of fields from the grid, making sure that article boundaries are aligned with each other[26]. Since we cannot know in advance how many articles our page will contain and what will be each article's relevance value–both being dependent on the meeting and its summary–we define for each field of the grid a *prominence* value. The prominence value is a positive number specifying how suitable a grid field is to portrait relevant information. Intuitively, the more relevant a topic is the more prominently it should be laid out on the page.[27]

The task of the layout manager is to assign a topic ID to each grid field. A topic's article will then be rendered on all grid fields bearing the same topic id. This assignment process is limited by the following constraints:

Size constraint Each article must be mapped to at least one and at most six grid fields.

MaxBounds constraint Articles should not be wider and not be higher than three grid fields at most. As with the grid resolution, this constraint will help keeping the search space compact.

Compactness constraint This constraints states that articles must always have a rectangular outline.

Rectangle constraint This constraints states that articles must always have a rectangular outline.

Maximum image distance constraint If more than one article contains a picture, the pictures should be placed with maximum distance to each other to avoid a cluttered look.

Overlap constraint Images should not overlap with headlines.

Given these constraints, the task of the layout manager is to find a layout that will feature the most relevant topics on the most prominent positions of the newspaper page. In

[26]For reasons of computational efficiency, we can not choose an arbitrary size for the grid, as higher resolutions showed to embiggen the search space of the constraint solver to an unmanageable degree. In our implementation, we found a 5×5 grid to be a perfectly cromulent choice as a trade-off between tractability and flexibility.

[27]For our implementation, we have used common sense assignments for the prominence values, where a broader empiric study might have been more appropriate. Note, however, that the actual values used for prominence representation are not inherent to our layout algorithm and could easily be exchanged.

technical terms, this is realized by having the underlying constraint solver optimizing the following objective function: $\sum_{i=1}^{n} prominence(n) \times relevance(n)$, where $prominence(n)$ is the prominence value of the $n-th$ grid field and $relevance(n)$ is the relevance of the topic assigned to the $n-th$ grid field.

The example in fig. 27 is the result of the following input data:

#	Topic	Relevance
1	Opening	5
2	Interface specialist presentation	40
3	Industrial designer presentation	15
4	Marketing expert presentation	30
5	New requirements	10
6	Closing	1

5.5.3 The Comic Strip Metaphor

As outlined above, the topic-based division of summary data into newspaper articles displayed on one single page allows for highly parallel information access, a quality highly desirable for result-based summaries. However, for other types of summaries, and in particular progress-based summaries, this approach suffers from the fact that it is not suitable for the representation of temporal progression. Progress-bases summaries can be very useful if the user is interested in the development of a meeting rather than just the final results.

To be able to make use of the advantages of multimedia summary presentation also for the case of progress-based summaries, we have explored a second type of layout: sequential art, or–as this type of medium is more commonly referred to–*comic strips*.

As the name suggests, sequential art is predestined to display temporal progression. Although contemporary art knows many variations of comic strip design and layout, the basic principle remains that of a sequence of graphical units, called *panels*, that display a snapshot of a scene. The sequence of snapshots is sorted temporally and generally ordered in the same way as the reading order for text (in western culture left to right, top to bottom).

In direct comparison with a newspaper layout, we observe that comic strips put an even higher focus on the medium *graphics* while newspapers are still very much text-based. Textual information in comic strips is typically conveyed in two different ways: through contextualizing narratives and through personal dialogs. The former is typically used to introduce a new scene to the reader or to provide non-obvious background information; it is usually displayed in form of a rectangular box in the upper left corner of the first panel of a new scene. Personal dialog represent spoken contributions and are always connected to a specific person (the speaker). Visually, these types of texts are realized with so-called "bubbles", text-boxes of elliptic shape and some kind of pointer to the speaker of the utterance[28].

[28]Comic strips use different types of bubbles to differentiate between *speech*, *thoughts* and others. In the case of meeting summaries, we restrict ourselves to speech bubbles for apparent reasons.

The following constraints determine the comic layout in our current implementation:

Anchor constraint The first panel has to be placed in the top left corner of the page.

Border constraint Panels may not exceed the borders of the page.

Sequential positioning constraint A panel is either placed directly to the right of its predecessor or it starts a new line. Subsequent lines must be placed directly underneath each other.

Gap filling constraint The last panel in a line must be placed less than 45mm away from the right hand border.

Panel width constraint The width of a panel is determined by its type: *panorama panels* have a width of 90mm, *simple portrait panels* 40mm, *wide portrait panels* 60mm, and *double portrait panels* 80mm.

Bubble placement constraint The first speech bubble must be placed in the first panel. The last speech bubble must be placed in the last panel.

Maximum bubble constraint A panel may not contain more than five bubbles to avoid cluttering. A simple portrait panel may not contain more than three bubbles.

Bubble order constraint Two subsequent speaker utterances must be realized in either the same panel or in immediately following panels.

Topic constraint Each topic is realized by one or more panels, based on the number of speaker utterances per topic and the maximum number of panels per comic page.

Topic panel constraint All speaker utterances within a panel must belong to the topic of the panel.

Bubble placement constraint Bubbles may overlap the panel border by no more than 2mm.

Reading direction constraint The placement of bubble must adhere to the left-to-right reading direction.

Narrative placement constraint Narrative text boxes are always placed in the top left corner of a panel.

Figure 29 shows an example output of a generated comic layout. We use the Comic Life tool[29] to render the final result of the generation process.

[29] Available at `http://www.plasq.com`

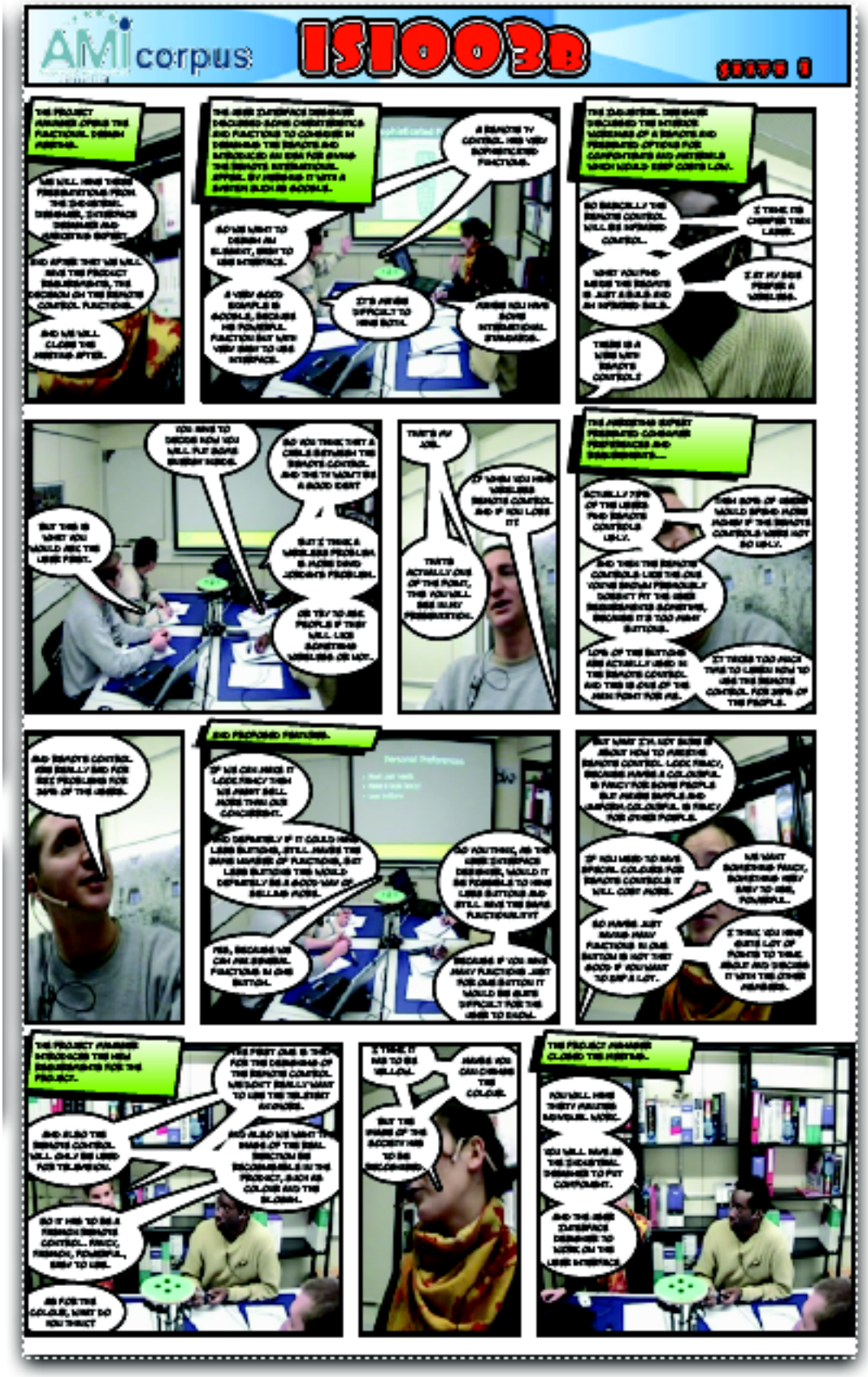


Figure 29: A generated comic-style summary of AMI meeting IS1003b

5.5.4 SuVi

Based on requirements drawn from the descriptions above, we have developed SuVi[30], an automatic layout system for the generation of summaries as multimedia documents. A core component of this system is a constraint solver that assigns the positions and sizes to all media object in a summary based on the constraints introduced in sections 5.5.2 and 5.5.3. Through the generality of the constraint-based approach, SuVi can be used for generating both, the newspaper and the comic book layouts.

From an abstract point of view, this layout process can be divided into two general steps. Both newspapers and comic books use rectangular boxes as a basic visual unit, newspaper articles and comic book panels respectively. Thus a first step consists in computing the size and position of each of these boxes on the page. Here, comic book generation differs in one important aspect from newspaper generation: the position of the next box doesn't have to be computed, it is either right next to the previous box or at the beginning of the next line. Newspaper articles, on the other hand, could in principle be placed at any position on the page, which is why we use the grid-based approach outlined above (s. 5.5.2).

Once the *macroscopic* computation of the layout of the unit boxes has been finished, SuVi lays out the content of each of the boxes. In the case of the newspaper system, this means the selection and placement of the article text, headlines and pictures according to the layout constraints. For comic books, the system has to select the relevant pictures and texts, assign the texts to speech bubbles or narrative text boxes and position these objects within the panel.

To do so, SuVi relies on external components that provide relevant information for the layout process, but which at this point of the implementation are not fully realized. Rather, they are replaced by "black boxes" to simulate the behavior of actual fully featured components. These black boxes are: headline and article text generator for newspapers, speaker contribution generator for comic books and image extractor for both.

Through its constraint-based nature, SuVi introduces the possibility to generate user-tailored layouts based on the personal preferences of a user. For instance, the relevance values of the different topics for newspaper articles could be drawn from user preferences, as could parameters, such as which topics to display, favorite colors, the number of pages, maximum number of panels per page etc., for comic books.

5.6 Conclusion and Future Work

This chapter has examined two challenges facing extractive summarization of meetings in progress: summarizing without access to the global meeting context, and summarizing without dialogue act segmentation. To address the former challenge, we implemented and described a score-trading mechanism by which we can reduce redundancy and increase informativeness based on a small amount of context for each candidate dialogue act. To address the latter challenge, we implemented a pause-based spurt segmentation and found that ROUGE results actually improved slightly compared with using dialogue acts, far

[30] SuVi is an acronym for Summary Visualizer

from decreasing summarization performance.

For abstractive summarization, we have laid the foundation for a system that can generate indicative abstracts automatically. We are currently developing our system further by adding more annotation layers to the processing pipeline. Additional steps are the extension of the underlying knowledge bases to increase the generation quality and work towards an online version of the system.

For multimedia summaries, our main goal is replacing the black box components currently used to simulate certain input data by actual implementations of the same functionalities. In particular, we would like to integrate our work on extractive and abstractive text summarization for the generation of newspaper article texts and for the generation of speaker utterances in comic books.

6 Decision Audit Evaluation

6.1 Introduction

In previous chapters, the automatic summaries were evaluated *intrinsically* by scoring them according to multiple human annotations of informativeness. That is, they were evaluated according to how well their information content matched the information content of gold-standard summaries. The most comprehensive and reliable evaluation of the quality of a given summary, however, is the degree to which it aids a real-world *extrinsic* task: an indication not just of how informative the summary is, but how useful it is in a realistic task. As mentioned in the introduction to this thesis, the purpose of these summaries is not to serve as stand-alone indicators of meeting information content, but to aid user *navigation* of the entire meeting content. The meeting summaries are meant to index the greater overall meeting record. We therefore design an extrinsic task that models a real-world information need, creat multiple experimental conditions comprised of various representations of meeting information content, and enlist subjects to participate in the task.

The chosen task is a *decision audit*, wherein a user must review previously held meetings in order to determine how a given decision was reached. This involves the user determining what the final decision was, which alternatives had previously been proposed, and what the arguments for and against the various proposals were. The reason this task was chosen is that it represents one of the key use cases for AMI technologies - that of aiding *corporate memory*, the storage and management of a organization's knowledge, transactions, decisions, and plans. A organization may find itself in the position of needing to review or explain how it came to a particular position or why it took a particular course of action. When business meetings are archived and summarized, this task is made much more efficient.

6.2 Task Motivation

Summarization evaluation can be divided into two types: *intrinsic* evaluation and *extrinsic* evaluation. Intrinsic evaluation involves measuring the actual information content of the summaries, usually as compared with a gold standard human summary or multiple human summaries. Weighted precision, as described in the preceding sections of this thesis, is an intrinsic measure that relies on multiple human extracts for comparison. Other metrics such as ROUGE [Lin and Hovy, May 2003] compare automatic summaries and model summaries at the n-gram level. Extrinsic methods, on the other hand, measure the usefulness of summaries in aiding the completion of a real-world task. For example, one might measure how well summaries aid users in answering a series of questions about a meeting.

This thesis argues that truly robust summarization evaluation will incorporate extrinsic measures in addition to intrinsic measures. While intrinsic evaluation metrics are indispensable for development purposes and can be easily replicated, they ideally need to be chosen based on whether or not they are good predictors for extrinsic usefulness, e.g. whether they correlate to a measure of real-world usefulness. Evaluating according to hu-

man gold-standard annotations is sensible, but ultimately all summarization work is done for the purpose of facilitating some task and should be evaluated in that context.

Specifically, our incorporation of extrinsic measures here is related to our domain of speech summarization and to the predicted use cases of the meeting summaries generated. The summaries are meant to be used in the context of a meeting browser, aiding a time-restricted user who needs to quickly review meeting content for such use cases as preparing for a subsequent meeting or plumbing corporate memory. In these cases, it is not sufficient merely to know that our automatically generated summaries are to some degree similar to manually drafted summaries, as the documents are not intended to be stand-alone documents. Rather, they are included in a meeting browser as a navigational tool. For example, a user of the meeting browser can first read the extractive summary in its entirety and then navigate the entire transcript and audio/video record by clicking on summary dialogue acts as an index into the record. It is crucial, therefore, to know just how good extractive summaries are as navigational tools for such purposes. Figure 30 illustrates the relationship between an extractive summary and the overall meeting record. Ultimately summarization may be merely one component of a multimedia meeting browser, but here we want to isolate the impact of summarization compared with other possible components or configurations. We are interested in how well we meet the needs of a particular use-case (a decision audit) when each individual information component is featured.

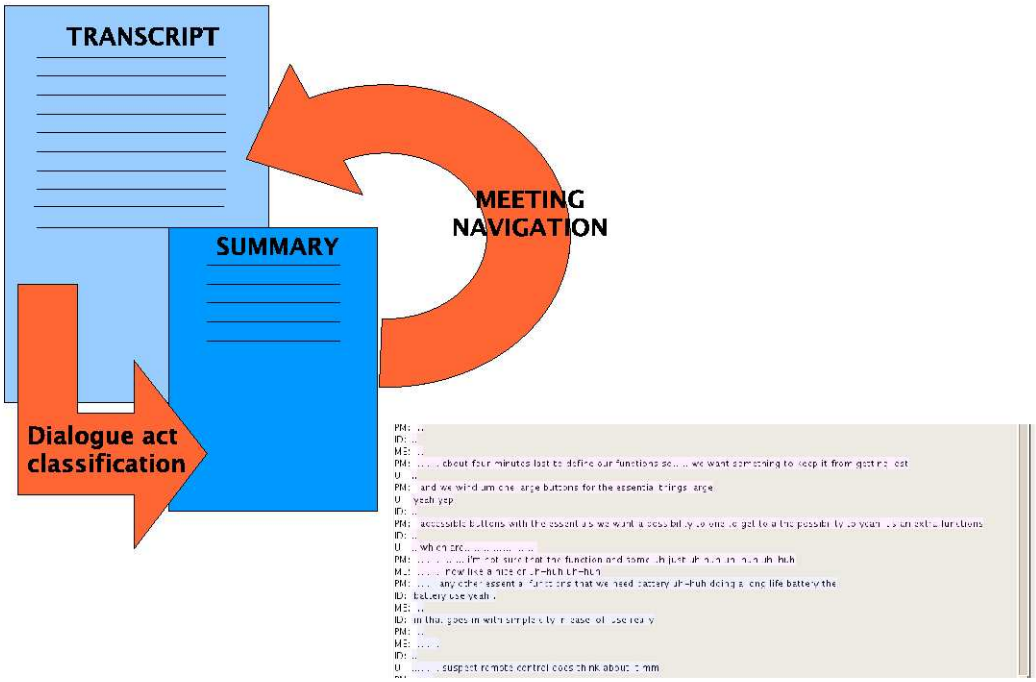


Figure 30: Summaries as Navigation Aids

6.3 Related Work

This section describes previous extrinsic evaluations relating either to summarization specifically, or else to the browsing of multiparty interactions more generally. We then describe how our decision audit browsers fit into a typology of multimedia interfaces.

6.3.1 Previous Work

In the field of text summarization, a commonly used extrinsic evaluation has been the *relevance assessment* task [Mani, 2001]. In such a task, a user is presented with a description of a topic or event and then must decide whether a given document (e.g. a summary or a full-text) is relevant to that topic or event. Such schemes have been used for a number of years and on a variety of projects [Jing et al., 1998, Mani et al., 1999, Harman and Over, 2004]. Due to problems of low inter-annotator agreement on such ratings, Dorr et. al [Dorr et al., 2005] proposed a new evaluation scheme that compares the relevance judgment of an annotator given a full text with that same annotator given a condensed text.

Another type of extrinsic evaluation for summarization is the *reading comprehension* task [Mani, 2001, Hirschman et al., 1999]. In such an evaluation, a user is given either a full source or a summary text and is then given a multiple-choice test relating to the full source information. A system can then calculate how well they perform on the test given the condition.

In the speech domain, there have been several large extrinsic IR evaluations in the past few years, though not necessarily designed with summarization in mind. Wellner et. al [Wellner et al., 2005] introduced the Browser Evaluation Test (BET), in which *observations of interest* are collected for each meeting, e.g. the observation "Susan says the footstool is expensive." Each observation is presented as both a positive and negative statement and the user must decide which statement is correct by browsing the meetings and finding the correct answer. It is clear that such a set-up could be used to evaluate summaries and to compare summaries with other information sources. We chose not to use this evaluation paradigm, however, because the observations of interest tend to be skewed towards a keyword search approach, where it would always be simplest to just search for a word such as "footstool" rather than read a summary.

Also on the AMI project, the Task-Based Evaluation (TBE) [Kraaij and Post, 2006] evaluates multiple browser conditions containing various information sources relating to a series of AMI meetings. Participants are brought in four at a time and are told that they are replacing a previous group and must finish that group's work. In essence, the evaluation involves re-running the final meetings of the series with new participants. The participants are given information related to the previous group's initial meetings and must finalize the previous group's decisions as best as possible given what they know. The reason we did not choose the TBE for this summarization evaluation is that the TBE evaluation relies on lengthy post-questionnaire results rather than more objective criteria. For example, users are asked to rate the statement "There is no better information source than this browser," when they may not in fact be in the position to know whether or not there are better options.

The SCANMail browser [Hirschberg et al., 2001, Whittaker, 2002] is an interface for managing and browsing voicemail messages, with multimedia components such as audio, ASR transcripts, audio-based paragraphs, and extracted names and phone numbers. To evaluate the browser and its components, the authors compared the SCANMail browser to a state-of-the-art voicemail system on four key tasks: scanning and searching messages, extracting information from messages, tracking the status of messages (e.g. whether or not a message has been dealt with), and archiving messages. Both in a think-aloud laboratory study and a larger field study, users found the SCANMail system outperformed the comparison system for these extrinsic tasks. The field study in particular yielded several interesting findings. In 24% of the times that users viewed a voicemail transcript with the SCANMail system, they did not resort to playing the audio. This testifies to the fact that the transcript and extracted information can, to some degree, act as substitutes for the signal, which user comments also back up. On occasions when users did play the audio, 57% of the time they did not play the entire audio. Most interestingly, 57% of the audio play operations resulted from clicking within the transcript. The study also found that users were able to understand the transcripts even with recognition errors, partly by having prior context for many of the messages.

The SpeechSkimmer browser [Arons, 1997] is an audio-based browser incorporating skimming, compression and pause-removal techniques for the efficient navigation of large amounts of audio data. The authors conducted a formative usability study in order to refine the interface and functionality of SpeechSkimmer, recruiting participants to find several pieces of relevant information within a large portion of lecture speech using the browser. Results were gleaned both from a think-aloud experiment structure as well as follow-up questions on ease of use. The researchers found that experiment participants often began the task by listening to the audio at normal speed to first get a feel for the discussion, and subsequently made good use of the skimming and compression features to increase search efficiency.

Whittaker et. al [Whittaker et al., to appear] describe a task-oriented evaluation of a browser for navigating meeting interactions. The browser contained a manual transcript, a visualization of speaker activity, audio and video streams with play, pause and stop commands, and artefacts such as slides and whiteboard events (the slides, but not the whiteboard events, are indices into the meeting record). Users were given two sets of questions to answer, the first set consisting of general "gist" question about the meeting, and the second set comprised of questions about specific facts within the meeting. There were 10 questions or tasks in total. User responses were subsequently scored on correctness compared with model answers. There are several interesting findings from this task-based evaluation. While general performance was not high, users found it much easier to answer specific questions than "gist" questions using this browser setup. This has special relevance for our work, as certain types of information needs might be easily satisfied without recourse to derived data such as summaries or topic segments, but getting the general gist of the meeting seems to be much more difficult. Very interestingly, users often felt that they had performed much better than they actually had. In other words, users seemed to be unaware that they had missed relevant or vital information and felt that they had provided comprehensive answers. Across the board, participants focused on reading the transcript rather than beginning with the audio and video records directly.

6.3.2 Multimodal Browser Types

Tucker and Whittaker [Tucker and Whittaker, 2005] provide an overview of the mechanisms available for browsing multimodal meetings. They establish a four-way browser classification: audio-based browsers, video-based browsers, artefact-based browsers, and derived data browsers. With audio-based browsers, the audio recordings of the meeting are the main focus, and are sometimes coupled with a visual index for navigating through the audio record by clicking on, for example, speaker segments [Kimber et al., 1995]. Other audio browsers feature the facility to alter playback speed or to compress the audio in some fashion [Arons, 1997, Tucker and Whittaker, 2006].

With video browsers, both audio and video are provided to the user, but the focus is on the video. These browsers are highly dependent on the actual environment of the meetings, as in some cases each participant will have a camera trained solely on them with additional room-view cameras [Carletta et al., 2006], and in other cases there may be a single panoramic camera for recording the meetings [Lee et al., 2002]. As with audio browsers, there may be a visual index or a facility for speed-up or compression. Another possibility for video browsers is to extract *keyframes* or video grabs, which are relevant static images from the video stream, and then present the keyframes in a story-board or comics format [Girgensohn et al., 2001, Kleinbauer et al., 2007].

The third class as established by Tucker and Whittaker is comprised of artefact-based browsers, with artefacts being information recorded in the meeting other than the audio/video streams. For the AMI meetings, artefacts include slides, notes, whiteboard drawings, and emails. Each of these can be very informative, and by synchronizing all of these sources of information to the audio/video record, a person using the browser can more fully get a sense of the meeting interactions. Furthermore, artefacts such as slides can be useful for indexing into the audio/video record.

The fourth class is comprised of browsers incorporating derived data forms. These browsers feature components that result from in-depth analysis of the meetings rather than simply recording various phenomena in the meetings. These components include ASR transcripts, topic segmentation, automatically generated summaries, dialogue act segmentation and labeling, and emotion or sentiment detection. These components provide structure and semantics to the meeting record, and again can act as efficient indices into the meeting record.

In light of this classification scheme, our decision audit browsers are video browsers incorporating derived data forms. Although other incarnations of our browsers contain meeting artefacts such as slides, we simplify the browsers as much as possible for this task by putting the focus on derived data forms and their usefulness for browsing the meeting records. Each version of the experiment browser is built using the Ferret [Wellner et al., 2004], an easily modifiable multimedia browser framework.

6.4 Task Setup

The data for the extrinsic evaluation is one meeting series ES2008 from the AMI corpus, comprised of 4 related meetings. The particular meeting series is chosen because it has been used in previous AMI extrinsic evaluations and the participant group in that series

worked well together on the task. The group took the task seriously and exhibited deliberate and careful decision-making processes in each meeting and across the meetings as a whole.

6.4.1 Task Overview

The extrinsic task is an individual task, unlike the AMI TBE, described above, which was a group-based scenario task. We recruited only participants who were native English speakers and who had not participated in previous AMI experiments or data collection. 10 subjects were run per condition, for a total of 50 subjects. For each condition, 6 participants were run in Edinburgh and 4 were run at DFKI, an AMI partner.

Each participant is first given a pre-questionnaire relating to background, computer experience and experience in attending meetings (see appendix X). In the case that the participant regularly participates in meetings, we ask how they normally prepare for a meeting, e.g. using their own notes, consulting with other participants, etc.

Each participant is then given general task instructions (appendix X). These instructions explain the meeting browser in terms of the information provided in the browser and the navigation functions of the browser, the specific information need they are meant to satisfy in the task, and a notice of the allotted time for the task. The total time alloted is 45 minutes, which includes both searching for the information and writing up the answer. This amount of time is based on the result of a pilot task for Condition EM, extractive summarization on manual transcripts.

The portion of the instructions detailing the specific task reads as follows:

> We are interested in the group's decision-making ability, and therefore ask you to evaluate and summarize a particular aspect of their discussion.
>
> The group discussed the issue of separating the commonly-used functions of the remote control from the rarely-used functions of the remote control. What was their final decision on this design issue? Please write a short summary (1-2 paragraphs) describing the final decision, any alternatives the participants considered, the reasoning for and against any alternatives (including why each was ultimately rejected), and in which meetings the relevant discussions took place.

This particular information need is chosen because the relevant discussion manifested itself throughout the 4 meetings, and the group went through several possibilities before designing an eventual solution to this portion of the design problem. In the first meeting, the group discussed the possibility of creating two separate remotes. In the second meeting, it was proposed to have simple functions on the remote and more complex functions on a sliding compartment of the remote. In the third meeting, they decided to have an on-screen menu for complex functions, and in the final meeting they finalized all of the details and specified the remote buttons. A participant in the decision audit task therefore would have to consult each meeting in order to get the full answer to the task's information need.

Condition	Description
KW	Top 20 keywords
EM	Extractive summary of manual transcripts
EA	Extractive summary of ASR transcripts
AH	Human abstracts
AA	Automatic abstracts

Table 57: Experimental Conditions

6.4.2 Experimental Conditions

There are 5 conditions run in total: one baseline condition, two extractive conditions and two abstractive conditions.

The baseline condition, Condition KW, consists of a browser with a manual transcript, audio/video record, and a list of the top 20 keywords in the meeting. The keywords are determined automatically using *su.idf*, a weighting scheme described earlier. Figure 31 shows a screenshot for the browser in Condition KW.

Conditions EM and EA present the user with a transcript, audio/video record and an extractive summary of each meeting, with the difference between the conditions being that the latter is based on ASR and the former on manual transcripts. The length of the respective extractive summaries is based on the length of the manual extracts for each meeting: approximately 1000 words for the first meeting, 1900 words for the second and third meetings, and 2300 words for the final meeting. These lengths correlate to the lengths of the meetings themselves. Figure 32 shows a screenshot for the browser in Conditions EM and EA.

Condition AH is the gold-standard condition, a human-authored abstractive summary. Each summary is divided into subsections: decisions, actions, goals and problems. These abstractive summaries vary in length. Each abstractive sentence is normally also linked to one or more transcript dialogue acts, making the experimental condition a hybrid of abstractive and extractive. Figure 33 shows a screenshot for the browser in Conditon HA.

Condition AA presents the user with an automatically generated abstractive summary, described by Kleinbauer et. al [Kleinbauer et al., 2007]. This summarization method utilizes automatic topic segmentation and topic labels, and finds the most commonly mentioned content items in each topic. A sentence is generated for each meeting topic indicating what was discussed, and these sentences are linked to the actual dialogue acts in the discussion. Figure 34 shows a screenshot for the browser in Condition AA.

Figure 57 lists and briefly describes the experimental conditions.

6.4.3 Browser Setup

The meeting browsers are built so as to exhibit as similar browser behaviour as possible across the experimental conditions. In other words, the interface is kept as similar as possible in all conditions to eliminate any potential confounding factors relating to the user interface.

In each browser, there are 5 tabs for the 4 meetings and a writing pad. The writing pad

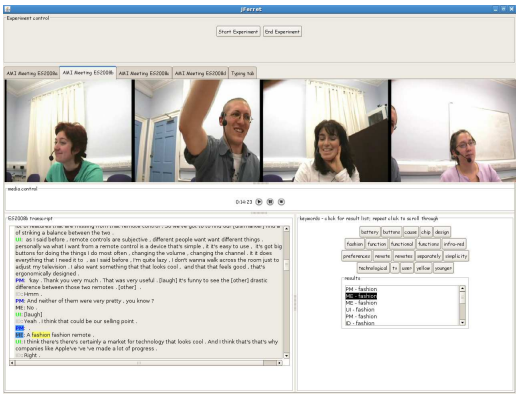


Figure 31: Condition KW Browser

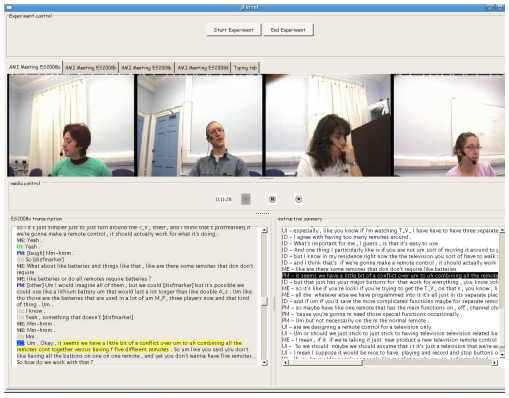


Figure 32: Conditions EM and EA Browser

was for the participant to author their decision audit summary. In each meeting tab, the videos displaying the 4 meeting participants are layed out horizontally with the media controls beneath. The transcript is shown in the lower left of the browser tab in a scroll window.

In Condition KW, each meeting tab contains buttons corresponding to the top 20 keywords for that meeting. Pressing the button for a given keyword highlights the first instance of the keyword in the transcript, as well as opening a listbox illustrating all of the occurrences of the word in the transcript, giving the user a context in terms of the word's frequency. Subsequent clicks highlight the subsequent occurrences of the word in the transcript, or the user may choose to navigate to keyword instances via the listbox.

In Conditions EM and EA, a scroll window containing the extractive summary appears next to the full meeting transcript. Clicking on any dialogue act in the extractive summary takes the user to that point of the meeting transcript and audio/video record.

In Conditions AH and AA, the abstractive summary is presented next to the meeting transcript. In Condition AA, the abstractive summary has different tabs for *decision*, *problems*, *goals* and *actions*. Clicking on any abstract sentence highlights the first linked dialogue act in the transcript and also presents a listbox representing all of the transcript dialogue acts linked to that abstract sentence. The user can thus navigate either by repeatedly clicking the sentence, which in turn will take them to each of the linked dialogue

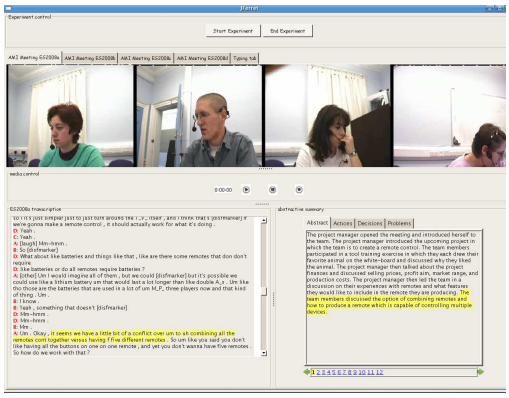


Figure 33: Condition AH Browser

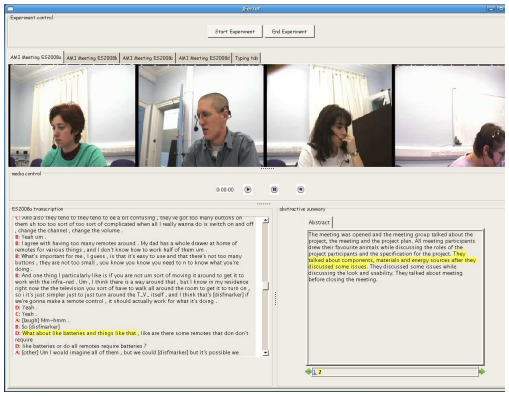


Figure 34: Condition AA Browser

acts in the transcript, or else they can choose a dialogue act from the listbox. The navigation options are underlyingly the same as Condition KW. The primary difference between Conditions KW, AH and AA on the one hand and Conditions EM and EA on the other is that the extractive dialogue acts link to only one point in the meeting transcript, whereas keywords and abstract sentences have multiple indices.

The browsers are designed in such a way that the writing tab where the participant types his answer is a fifth tab in addition to the four individual meeting tabs. As a consequence, the participant cannot view the meeting tabs while typing the answer; they are restricted to tabbing back and forth as needed. This was designed deliberately so as to be able to discern when the participant was working on formulating or writing the answer on the one hand and when they were browsing the meeting records on the other.

After reading the task instructions, each participant is briefly shown how to use the browser's various functions for navigating and writing in the given experimental condition. They are then given several minutes to familiarize themself with the browser, until they state that they were comfortable and ready to proceed. The meeting used for this familiarization session is not one of the ES2008 meetings used in the actual task. In fact, it was one of the AMI non-scenario meetings; this is done so that the participant will not become familar with the ES2008 meetings specifically or the scenario meetings in general before beginning the task. This familiariation time is carried out before the task began so

that we could control for the possibility that one condition would have a more difficult learning curve than the others.

6.4.4 Logfiles

In each condition of the experiment, we log a variety of information relating to the participant's browser use and typing. In all conditions, we log transcript clicks, media control clicks (i.e. play, pause, stop), movement between tabs, and characters entered into the typing tab, all of which are time-stamped. In Condition KW, we log each keyword click and note its index in the listbox, e.g. the first occurrence of the word in the listbox. In Conditions EM and EA, each click of an extractive summary sentence is logged, and in the abstract conditions each abstract sentence click is logged along with its index in the listbox, analogous to the keyword condition. Because there are not multiple links in the extractive condition – in other words, each extract sentence links only to one transcript sentence – there is no need for listboxes and listbox indices.

To give an example, the following portion of a logfile from a Condition AH task shows that the participant click on the transcript, played the audio, paused the audio, clicked link number 1 of sentence 5 in the Decisions tab for the given meeting, then switched to the typing tab and began typing the word "six."

```
2007-05-24T14:46:45.713Z transcript_jump 687.85 ES2008d.sync.1375
2007-05-24T14:46:45.715Z button_press play state media_d
2007-05-24T14:46:45.715Z button_press play state media_d
2007-05-24T14:47:30.726Z button_press pause state media_d
2007-05-24T14:47:30.726Z button_press pause state media_d
2007-05-24T14:47:52.379Z MASCOT  (observation ES2008d): selected link #1
2007-05-24T14:47:53.613Z tab_selection Typing tab
2007-05-24T14:47:54.786Z typed_insert s 316
2007-05-24T14:47:54.914Z typed_insert i 317
2007-05-24T14:47:55.034Z typed_insert x 318
```

6.4.5 Evaluation Features

For evaluation of the decision audit task, there are 3 types of features to be analyzed: the answers to the users' post-questionnaires, human ratings of the users' written answers, and features extracted from the logfiles that relate to browing and typing behaviour in the different conditions.

Upon completion of the decision audit task, we present each participant with a post-task questionnaire consisting of 10 statements with which the participant can state their level of agreement or disagreement via a 5-point Likert scale, such as *I was able to efficiently find the relevant information*, and 2 open-ended questions about the specific type of information available in the given condition and what further information they would have liked. Of the 10 statements evaluated, some are re-wordings of others with the polarity reveresed in order to gauge the users' consistency in answering.

In order to gauge the goodness of a participant's answer, we enlist two human judges to do both *subjective* and *objective* evaluations. For the subjective portion, the judges first read through all 50 answers to get a view of the variety of answers. They then rate each answer using a 1-8 Likert-scale on criteria relating to the precision, recall and f-score of the answer. For the objective evaluation, 3 judges construct a gold-standard list of items that should be contained in an ideal summary of the decision audit. For each participant answer, they check off how many of the gold-standard items are contained. Due to the fact that some participant answers included written text in paragraph form in addition to rough notes, summaries with both notes and text are evaluated twice, first considering all the text that was submitted and a second time considering only the written paragraphs were submitted. This is done because it was not clear whether the notes were meant to be submitted as part of the answer or were simply not deleted before time had expired.

The remainder of the features for evaluation are automatically derived from the logfiles. These features have to do with browsing and writing behaviour as well as the duration of the task. These include the total experiment length, the amount of time before the participant began typing their answer, the total amount of tabbing the user did normalized by experiment length, the number of clicks on content buttons (e.g. keyword buttons or extractive summary sentences) per minute, the number of content button clicks normalized by the number of unique content buttons, number of times the user played the audio/video stream, the number of content clicks prior to the user clicking on the writing tab to begin writing, the document length including deleted characters, the document length excluding deleted characters, how many of the 4 meetings the participant looked at, and the average typing timestamp normalized by the experiment length.

The total experiment length is included because it is assumed that participants would finish earlier if they had better and more efficient access to the relevant information. The amount of time before typing begins is included because it is hypothesized that efficient access to the relevant information would mean that the user would begin typing the answer sooner. The total amount of tabbing is considered because a participant who is tabbing very often during the experiment is likely jumping back and forth between meetings trying to find the information, indicating that the information is not conveniently indexed. The content clicks are considered because a high number of clicks per minute would indicate that the participant is finding that method of browsing to be helpful, and the number of content clicks normalized by the total unique content buttons indicates whether they made full use of that information source. The number of audio/video clicks is interesting because it is hypothesized that a user without efficient access to the relevant information will rely more heavily on scanning through the audio/video stream in search of the answers. The number of content clicks prior to the user moving to the writing tab indicates whether a content click is helpful in finding a piece of information that led to writing part of the answer. The document length is considered because a user with better and more efficient access to the meeting record will be able to spend more time writing and less time searching. Because the logfiles show deleted characters, we calculate both the total amount of typing and the length of the final edited answer in characters. The number of meetings examined is considered because a user who has trouble finding the relevant information may not have time to look at all 4 meetings. The final feature, which is the average timestamp normalized by the experiment length, is included because a user with efficient access to the information will be able to write the answer throughout the course

Question	CondKW	CondEM	CondEA	CondAH	CondAA
Q1: *I found the meeting browser intuitive and easy to use*	3.8	4.0	3.02_{AH}	$4.3^{EA,AA}$	3.7_{AH}
Q2: *I was able to find all of the information I needed*	2.9_{AH}	3.8	2.9_{AH}	$4.1^{KW,EA,AA}$	3.0_{AH}
Q3: *I was able to efficiently find the relevant information*	2.8_{AH}	3.4^{AA}	2.5_{AH}	$4.0^{KW,EA,AA}$	$2.65_{EM,AH}$
Q4: *I feel that I completed the task in its entirety*	2.3_{AH}	3.1	2.3	3.2^{KW}	2.9
Q5: *I understood the overall content of the meeting discussion*	3.8	**4.5**	3.9	4.1	3.9
Q6: *The task required a great deal of effort*	3.0	2.6^{EA}	3.9_{EM}	3.1	3.2
Q7: *I had to work under pressure*	3.3	**2.6**	3.3	2.7	3.1
Q8: *I had the tools necessary to complete the task efficiently*	3.1_{EM}	$4.3^{KW,EA,AA}$	3.0_{EM}	4.1	3.5_{EM}
Q9: *I would have liked additional information about the meetings*	3.0_{EM}	2.0^{KW}	2.4	2.6	2.7
Q10: *It was difficult to understand the content of the content of the meetings using this browser*	2.1	$1.5^{EA,AA}$	2.7_{EM}	2.0	2.3_{EM}

Table 58: Post-Questionnaire Results

of the experiment, whereas somebody who has difficulty finding the relevant information may try to write everything at the last moment.

6.5 Results

6.5.1 Post-Questionnaire Results

Table 58 gives the post-questionnaire results for each condition. For each score in the table, that score is significantly better than the score for any conditions in superscript, and significantly worse than the score for any condition in subscript. The only significant results listed are those that are significant at the level ($p<0.05$). Results that are not significant but are nonetheless unexpected or interesting are listed in boldface.

For the first post-questionnaire question, *I found the meeting browser intuitive and easy to use*, the best condition overall is Condition AH, incorporating human abstracts, followed by Condition EM. There is no significant difference between the two conditions. The lowest score is for Condition EA. Since the only difference between Conditions EM and EA is manual versus ASR transcripts, it's clear that ASR alone makes the browser less straight-forward and easy to use for participants.

For the second post-questionnaire question, *I was able to find all of the information I needed*, the conditions roughly form two groups. Conditions AH and EM are again at the top, scoring 4.1 and 3.8 respectively, while the remaining three conditions all score around 3.0. There is no significant difference between Conditions AH and EM.

The third question was *I was able to efficiently find the relevant information*, and for this criterion the human abstracts are clearly superior, performing significantly better than Conditions KW, EA and AA. Condition EM is second best and not significantly worse than Condition AH, but is substantially lower on average. Surprisingly, the automatic abstracts perform worse than the baseline Condition KW on this criterion.

For question four, *I feel that I completed the task in its entirety*, the scores overall are somewhat low, indicating the difficulty of the task. The best conditions are Condition EM and Condition AH with scores of 3.1 and 3.2 respectively. Condition AH is significantly better than the baseline Condition KW. The lack of large differences across conditions

regarding this criterion confirms that it is a challenging task to complete in the allotted time.

For question five, *I understood the overall content of the meeting discussion*, the best condition is Condition EM, extractive on manual transcripts, with a score of 4.5. While this is several points higher than even the human abstract condition, there are no significant differences between the conditions for this criterion. Nonetheless, it is very encouraging that the extractive conditions provide a good overview of the meeting content compared with the other conditions. Even with ASR, Condition EA fares very well on this criterion.

For question six, *The task required a great deal of effort*, Condition EM is again the best with a score of 2.6 (the lower the score, the better). The worst score, i.e. the highest, is Condition EA, showing that an ASR transcript does increase the effort required to complete the task compared with having a manual transcript.

Similarly for question seven, *I had to work under pressure*, Condition EM is the best with a score of 2.6 and Condition AH is comparable with a score of 2.7. There are no significant differences between the conditions. Conditions KW and EA score the worst on this criterion.

For question eight, *I had the tools necessary to complete the task efficiently*, Condition EM is again the highest with a score of 4.3 followed by Condition AH with a score of 4.1 Condition EM is significantly better on this criterion than Conditions KW, EA and AA. This is quite an encouraging result for extractive summarization, as the question directly addresses the tools available to the user and the extractive condition comes out on top. Not only does it perform the best overall, but the score of 4.3 is quite high on the 1-5 Likert scale, indicating user satisfaction with the browser content.

For the final two questions, Condition EM again performs the best. For the question *I would have liked additional information about the meetings*, Condition EM is rated with a 2.0 on average, followed by Condition EA with a score of 2.4 Thus, the two extractive conditions come out on top, superior to even the human abstract condition. For the question *It was difficult to understand the content of the meetings using this browser*, Condition EM is rated with a 1.5 on average followed by Condition AH with an average score of 2.0 (again, the lower the better for the last two questions). For this criterion, Condition EM is considerably better than the rest, with significant results compared with Conditions EA and AA. The low score for Condition EA shows that the incorporation of ASR transcripts does make it more difficult to understand the meetings for participants in this task, but even that score of 2.7 for Condition EA is fairly low on the Likert scale. These final two questions indicate that users are quite satisfied with the information provided by the extractive summaries and that the summaries allow them to understand the meetings without much difficulty.

Discussion It can first be noted that participants in general find the task to be challenging, as evidenced by the average answers on questions 4, 6 and 7. The task was designed to be challenging and time-contrained, because a simple task with a plentiful amount of time would allow the participants to simply read through the entire transcript or listen and watch the entire audio/video record in order to retrieve the correct information, disregarding other information sources. The task as designed requires efficient navigation of the

information in the meetings in order to finish the task completely and on time.

The results of the post-questionnaire data are quite encouraging in that the users seem very satisfied with the extractive summaries relative to the other conditions. It is not surprising that the gold-standard human-authored summaries are ranked best overall on several criteria, but even on those criteria the extractive condition on manual transcripts is a close second. For question 5, which relates to overall comprehension of the information in the meetings, extractive summaries are rated the highest of all. Extractive summaries of manual transcripts are also rated the best in terms of the effort required to conduct the task. But perhaps the most compelling result is on question 8, relating to having the tools necessary to complete the task. Not only is Condition EM rated the best, but it is significantly better than all conditions except the gold-standard abstracts. These results taken together indicate that extractive summaries are natural to use as navigation tools, facilitate understanding of the meeting content, and allow users to be more efficient with their time. From the viewpoint of user satisfaction, this result is the best that could be hoped for.

However, it is quite clear that the errors within an ASR transcript present a considerable problem for users trying to quickly retrieve information from the meetings. While it has repeatedly been shown that ASR errors do not cause problems for our algorithms according to intrinsic measures, these errors make user comprehension more difficult. For the questions relating to the effort required, the tools available, and the difficulty in understanding the meetings, Condition EA is easily the worst, scoring even lower than the baseline condition. It should be noted however, that a baseline such as Condition KW is not a true baseline in that it is working off of *manual* transcripts and would be expected to be worse when applied to ASR.

This finding about the difficulty of human processing of ASR transcripts will change and improve as the state-of-the-art in speech recognition improves. The finding also indicates that the use of confidence scores in summarization is desirable. While summarization systems naturally tend to extract units with lower WER, the summaries can likely be further improved for human consumption by compression via the filtering of low-confidence words.

6.5.2 Human Evaluation Results - Subjective and Objective

Subjective Evaluation Table 59 gives the results for the human subjective and objective evaluations. For each score in the table, that score is significantly better than the score for any conditions in superscript, and significantly worse than the score for any condition in subscript. The only significant results listed are those that are significant at the level ($p<0.05$). Results that are not significant but are nonetheless unexpected or interesting are listed in boldface.

Before beginning the subjective evaluation of decision audit answers, the two human judges read through all 50 answers in order to gauge the variety of answers in terms of completeness and correctness. They then rate each answer on several criteria roughly related to ideas of precision, recall and f-score, as well as effort, comprehension and writing style. They used a 1-8 Likert scale for each criterion. We then average their scores to derive a combined score for each criterion.

Criterion	CondKW	CondEM	CondEA	CondAH	CondAA
Q1: *overall quality*	3.0_{AH}	4.15	3.05_{AH}	$4.65^{KW,EA}$	4.3
Q2: *conciseness*	$2.85_{EM,AH,AA}$	4.25^{KW}	3.05_{AH}	$4.85^{KW,EA}$	4.45^{KW}
Q3: *completeness*	2.55_{AH}	3.6	2.6_{AH}	$4.45^{KW,EA}$	3.9
Q4: *task comprehension*	$3.25_{EM,AH}$	$\mathbf{5.2}^{KW,EA}$	$3.65_{EM,AH}$	$5.25^{KW,EA}$	4.7
Q5: *participant effort*	4.4	$\mathbf{5.2}^{EA}$	$3.7_{EM,AH,AA}$	5.3^{EA}	4.9^{EA}
Q6: *writing style*	4.75	5.65^{EA}	$4.1_{EM,AH,AA}$	5.7^{EA}	5.8^{EA}
Q7: *objective rating*	4.25_{AH}	7.2	5.05_{AH}	$9.45^{KW,EA}$	7.4

Table 59: Human Evaluation Results - Subjective and Objective

For the "overall quality" criterion, Condition AH, incorporating human abstracts, is superior, with an average of 4.85. The worst conditions overall are Condition KW and Condition EA, each scoring around 3.0. Extracts of manual transcripts and automatic abstracts are slightly worse than the gold-standard condition.

For the evaluation of "conciseness," the trends are largely the same as for the "overall quality" question. Condition AH is the best with an average of 4.85, followed by Conditions 4 and 1 with scores of 4.45 and 4.25, respectively. Condition KW is easily the worst, performing significantly worse than every other condition with the exception of Condition EA.

The pattern is similar for the evaluation of "completeness," with Condition AH faring best of all followed by Conditions AA and EM in order. On this criterion there is a clearer gap between the gold-standard condition and the remaining conditions, illustrating the utility of a manual abstract for providing complete coverage of the meeting. Worst for "completeness" is Condition KW.

For the criteria of "task comprehension" and "participant effort", we find Condition EM scoring nearly as well as Condition AH. For Condition EA, incorporating ASR, these scores significantly decrease, illustrating the challenge that an errorful transcript poses in terms of users understanding the task and demonstrating a concerted effort to satisfy the information need. Of course, it is difficult to discern incomprehension or low effort from what could simply be a difficult task.

For the evaluation of "writing style", we find that Conditions EM, AH and AA are rated similarly, while Condition EA scores the worst. There may be numerous factors for how ASR affects writing style in this task, but it may be that users are unable to decipher exactly what is discussed and subsequently their write-ups reflect this partial understanding, or it could simply be that they have less time to spend on writing because their browsing is less efficient. We will examine this latter point in further detail in the logfile results section below.

What these findings together help illustrate is that extractive summaries can be very effective for conducting a decision audit by helping the user to generate a concise, complete high-quality answer, but that the introduction of ASR has a measurable and significant impact on the subjective evaluation of quality. Interestingly, the scores on each criterion and for each condition tend to be somewhat low on the Likert scale, due to the difficulty of the task.

Objective Evaluation After the annotators carried out their objective evaluations, they met again and went over all experiments where their ratings diverged by more than two points, in order to form a truly *objective* and agreed-upon evaluation of how many gold-standard items each participant found. There were 12 out of 50 ratings pairs that needed revision in this manner.

According to the objective evaluation, Condition AH is superior, with an average more than two points higher than the next best condition. The worst overall is the baseline Condition KW, averaging only 4.25 hits. However, while the worst two conditions are significantly worse than the best overall condition, there are no significant differences between the other pairs of conditions, e.g. Condition EA incorporating ASR is not significantly worse than Conditions EM and AA. So even with an errorful transcript, participants in Condition EA are able to retrieve the relevant pieces of information at a rate not significantly worse than participants with a manual transcript. The quality may be worse from a subjective standpoint, as evidenced in the previous section, but the decision audit answers are still informative and relevant.

For the objective evaluation, in any given condition there is a large amount of variance that is simply down to differences between users. For example, even in the gold-standard Condition AH there are some people who can only find one relevant item whilst others find 16 or 17. Given a challenging task and a limited amount of time, some people may have simply felt overwhelmed in trying to locate the informative portions efficiently.

Table 59 summarizes the human evaluation results for both the subjective and objective criteria.

Discussion For the objective human evaluation, the gold-standard condition scores dramatically higher than the other conditions in hitting the important points of the decision process being audited. This goes to show that there is much room for improvement in terms of automatic summarization techniques. However, Conditions EM, EA and AA average much higher than the baseline Condition KW. There is considerable utility in such automatically-generated documents. It can also be noted that Condition EM is the best of the conditions with fully-automatic content selection.

Perhaps the most interesting result of the objective evaluation is that Condition EA, which uses ASR transcripts, does not deteriorate relative to Condition EM as much as might have been expected considering the post-questionnaire results. What this seems to demonstrate is that ASR errors are annoying for the user but that the users are able to look past the errors and still find the relevant information efficiently. Condition EA scores much higher than the baseline condition that utilizes manual transcripts, and this is a powerful indicator that summaries of errorful documents are still very valuable documents.

6.5.3 Logfile Results

Table 60 gives the results for the logfiles evaluation. For each score in the table, that score is significantly better than the score for any conditions in superscript, and significantly worse than the score for any condition in subscript. The only significant results listed are those that are significant at the level ($p<0.05$). Results that are not significant but are

Feature	CondKW	CondEM	CondEA	CondAH	CondAA
Q1: *duration*	45.4	43.1	45.4	45.42	43.2
Q2: *first typing*	16.25	13.9	17.14	8.61	10.22
Q3: *tabbing*	0.98	0.81^{AH}	0.72^{AH}	$1.4_{EM,EA}$	1.13
Q4: *perc. buttons clicked*	0.39	0.11	0.08	0.08	0.18
Q5: *clicks per minute*	1.33	**2.24**	1.47	1.99	0.83
Q6: *media clicks*	15.4^{EA}	14.4^{EA}	$40.4_{KW,EM,AH}$	16.6^{EA}	20.6
Q7: *click/writing corr.*	0.03	0.01	0.01	0.01	0.01
Q8: *unedited length*	1400	1602	1397	**2043**	1650
Q9: *edited length*	1251	1384	1161	**1760**	1430
Q10: *num. meetings*	3.9	4.0	3.9	4.0	4.0
Q11: *ave. writing timestamp*	0.68	0.73	$0.76_{AH,AA}$	0.65^{EA}	0.65^{EA}

Table 60: Logfile Feature Results

nonetheless unexpected or interesting are listed in boldface.

One result that was not anticipated is that almost all participants take the full 45 minutes to complete the experiment. There are no significant differences between the conditions on this criterion, though Condition EM has the lowest average task duration at 43 minutes. One hypothesis is that paid volunteers want to do as thorough of a job as possible and so remain for the entirety of the allotted time even if they have finished the bulk of the experiment earlier. This is backed anecdotally by participants reporting afterwards that "you can always use more time," suggesting that answers can always be refined even when near completion. More generally, it turned out to be a challenging task to complete in 45 minutes, regardless of condition.

The second feature is the amount of time before the participant began typing the answer. Condition AH is best overall with an average time of 8.6 minutes. Condition AA is next best with 10.225 minutes, Condition EM with 13.9 minutes, Condition KW with 16.25 minutes and Condition EA with 17.137 minutes. However, there are no significant differences between conditions. It is nonetheless clear that human abstracts allow the users to quickly index into the relevant portions of the meeting and begin writing the decision audit answer quite quickly.

The results of the third feature are surprising. The metric is the total amount of moving between browser tabs, normalized by the length of the experiment. The intuition behind the inclusion of this feature is that users who have efficient access to the relevant, important information will not need to continually tab back and forth between the browser tabs, searching for the information. The best (i.e. lowest) score overall is Condition EA, extractive summaries on ASR transcripts, followed by Condition EM, extractive summaries on manual transcripts. The worst overall is Condition AH, human abstracts. Conditions EM and EA are significantly better than Condition AH.

The fourth and fifth features relate to the number of clicks on content items, e.g. keyword clicks or extractive summary clicks. The fourth feature normalizes the number of clicks by the total number of content buttons. For example, if five unique keyword buttons were clicked out of a possible 20, the score would be 0.25. The fifth feature normalizes the number of content clicks by the length of the experiment, i.e. it represents the number of clicks per minute. For the fourth feature, Condition KW is the best overall with an average score of 0.386, significantly better than Conditions EA and AH. For the fifth feature, Condition EM is best overall with an average of 2.24 content clicks per minute,

followed by Condition AH with an average of 1.993. Condition AA is the worst with an average of 0.831. There are no significant differences between conditions. The fifth logfile feature is more likely to be reliable than the fourth, as the number of keywords for each meeting is only 20 and it's not surprising that the percentage of buttons clicked is higher than for the other conditions. The clicks-per-minute result is interesting for two reasons: extracts are used for navigation with considerably more frequency than the other conditions, and there are very few navigation clicks in Condition AA, incorporating automatic abstracts. We find that with extracts on ASR, users click the extracted dialogue acts less often than on manual transcripts, but still more often than in Conditions KW and AA.

The sixth feature is the number of media clicks, i.e. the number of times the user played the audio/video. The best condition is Condition EM, followed by Condition KW. The most interesting and dramatic result, however, is that Condition EA, extractive summarization on ASR, is much worse than all the other conditions. Whereas the average number of media clicks for Condition EM is 14.4, for Condition EA it is 40.4. This illustrates that the errorful ASR transcripts cause the users to rely much more heavily on the audio/video stream. Participants in Condition AA also rely more on the audio/video streams than participants in the top three conditions.

The seventh feature is the proximity of content clicks to writing tab clicks. Condition KW is best overall, but there are no significant differences between conditions. It seems to simply be a rare occurrence for a user to click a content item and began writing soon afterward. More likely, they click a content item and navigate to that part of the meeting, study the transcript in more detail, and finally synthesize the information in the writing tab.

The eight and ninth features relate to the length of the user's answer. For feature eight, the unedited answer length, Condition AH is best overall with an average character length of 2043.2. The worst is the baseline Condition KW with an average of 1399.6. Interestingly, for the ninth feature - edited answer length - the scores are much closer. Condition AH is still the best overall with an average length of 1760.6, but Condition KW is 1251.1. This illustrates that users in Condition AH have much more time for editing and refining their answers. They might begin by writing everything they find that seems relevant, then they condense or combine information for the final answer.

The tenth feature is the number of meetings the user looked at. The intuition is that if a given condition was not very efficient in the way that it presented information, users might not have time to look at all the data. In reality, however, almost all participants looked at all of the meetings, and so there are no differences on this criterion.

The final feature is the average location within the 45 minute period of the user typing. That is, it is the average of the timestamps normalized by the initial timestamp. The intuition is that users in a condition with more efficient access to information will do more typing early on in the experiment, whereas a person in an inefficient condition would be forced to do much of the writing at the end of the experiment. Condition AH was best overall with a score of 0.650, whereas Condition EA was the worst with a score of 0.725. Participants with access to a human summary are able to do the bulk of their writing earlier on in the experiment, whereas participants using an ASR transcript do much of their writing towards the end of the experiment. In the latter case, this leaves them less

time for revision, which is presumably related to the low writing quality scores presented in the previous section on subjective evaluations, above.

Discussion It is difficult to derive a single over-arching conclusion from the logfile results, but there are several interesting results on specific logfile features. Perhaps the most interesting is the dramatic difference that exists in terms of relying on the audio/video record when using ASR. The average number of media clicks when using extractive summaries on manual transcripts is only just above 14, but when applied to ASR this number is over 40 clicks. This ties together several interesting results from the post-questionnaire data, the human evaluation data, and the logfile data. While the ASR errors seem to annoy the participants and therefore affect their user satisfaction ratings, they are nonetheless able to employ the ASR-based summaries to locate the relevant information efficiently and thereby score highly according to the human objective evaluation. Once they have indexed into the meeting record, they then rely heavily on the audio/video record presumably to disambiguate the dialogue act context. It is *not* the case that particpants in this condition used only the audio/video record and disregarded the summaries, as they clicked the content items more often than in Conditions KW and AA (Q5). Overall, the finding is thus that ASR errors are annoying but do not obscure the value of the extractive summary.

It is also interesting that both extractive conditions lead to participants needing to move between meeting tabs less than in other conditions. As mentioned above, the intuition behind the inclusion of this feature was that a lower number would be better because it meant the user was finding information efficiently. However, it's surprising that Condition EA scored the "best" and Condition AH the "worst." It may be the case that participants in Condition AH felt more free to jump around because navigation was generally easier.

Many of the logfile features confirm that the human abstract gold-standard is difficult to challenge in terms of browsing efficiency. Users in this condition begin typing earlier, write most of their answer earlier in the task, write longer answers, and have more time for editing.

6.6 General Discussion

Overall these results are very good news for the extractive summarization paradigm. Users find extractive summaries to be intuitive, easy-to-use and efficient, are able to employ such documents to locate the relevant information in a timely manner according to human evaluations, and users are able to adapt their browsing strategies to cope with ASR errors. While extractive summaries might be far from what people conceptualize as a meeting summary in terms of traditional meeting minutes, they are intuitive and useful documents in their own right.

Specifically, we have found that users in Condition EM are very satisfied with the tools at their disposal, with the efficiency and intuitiveness of the browser setup, and their ability to rapidly find the relevant information. Condition EM is the superior condition for several post-questionnaire criteria, such as Q8, which asks whether the user has the tools necessary to find the relevant information efficiently. In Condition EA, incorporating ASR, users reported that they understood the overall content of the meeting discussions

and did not desire any additional information, giving positive ratings compared with other conditions. The ASR did, however, affect their effiency and ease-of-use ratings.

For the subjective human evaluation, the gold-standard Condition AH was rated the best on nearly all criteria, but was challenged by Condition EM on several of them, including the criteria of task comprehension and participant effort. Condition EM also had high scores for overall quality, conciseness and completeness compared with Condition AH. While the answers in Condition EA were scored more severely in the subjective evaluation, the human objective evaluation showed that participants working with ASR were still able to locate the relevant pieces of information at a rate not significantly worse than participants using manual transcript extracts.

Finally, there are a couple of especially interesing results from the logfiles analysis. First of all, participants in Condition AH are able to answer the question earlier in the experiment than participants in Condition EA. Second, participants in Condition EA rely much more on the audio/video streams than particpants in other conditions.

Perhaps the most interesting result from the decision audit overall is regarding the effect of ASR on carrying out such a complex task. While participants using ASR find the browser to be less intuitive and efficient, they nonetheless feel that they understand the meeting discussions and do not desire additional information sources. In a subjective human evaluation, the quality of the answers in Condition EA suffers according to most of the criteria, including writing style, but the participants are still able to find many of the relevant pieces of information according to the objective human evaluation. We find that users are able to adapt to errorful transcripts by using the summary dialogue acts as navigation and then relying much more on audio/video for disambiguating the conversation in the dialogue act context. Extractive summaries, even with errorful ASR, are useful tools for such a complex task, particularly when coupled with multimodal sources of information.

6.7 Conclusion

We have presented an extrinsic evaluation paradigm for the automatic summarization of spontaneous speech in the meetings domain: a decision audit task. This represents the largest extrinsic evaluation of speech summarization to date. In each condition of the experiment, users were required to use the presented information in order to find and extract information relevant to a specific information need. The largely positive results for the extractive conditions justify continued research on this summarization paradigm. However, the considerable superiority of gold-standard abstracts in many respects also support the view that research should begin to try to bridge the gap between extractive and abstractive summarization.

It is widely accepted in the summarization community that there should be increased reliance on extrinsic measures of summary quality. It is hoped that the decision audit task will be a useful paradigm for future evaluation work. For development purposes, it is certainly the case that intrinsic measures are indispensable: as mentioned before, in this work we use intrinsic measures to evaluate several summarization systems against each other and use extrinsic measures to judge the usefulness of the extractive methods in general. Intrinsic and extrinsic methods should be used hand-in-hand, with the former

as a valuable development tool and predictor of usefulness and the latter as a real-world evaluation of the state-of-the-art.

7 Topic Segmentation

7.1 Multimodal Integration in Meeting Discourse Segmentation

7.1.1 Introduction

Recent advances in multimedia technologies have led to huge archives of audio and video recordings of multiparty conversational speech in a wide range of areas including clinical use, online video sharing services, and meeting capture and analysis. While it is straightforward to replay such recordings, finding information from the often lengthy archives is a more challenging task. Annotating implicit semantics to enhance browsing and searching of recorded conversational speech has therefore posed new challenges to the field of multimedia information retrieval.

One critical problem is how to divide unstructured conversational speech into a number of locally coherent segments. The problem is important for two reasons: First, empirical analysis has shown that annotating transcripts with semantic information (e.g., topics) enables users to browse and find information from multimedia archives more efficiently [Banerjee et al., 2005]. Second, because the automatically generated segments make up for the lack of explicit orthographic cues (e.g., story and paragraph breaks) in conversational speech, dialogue segmentation is useful in many spoken language understanding tasks, including anaphora resolution [Grosz and Sidner, 1986], information retrieval (e.g., as input for the TREC Spoken Document Retrieval (SDR) task), and summarization [Zechner and Waibel, 2000].

This study therefore aims to explore whether a Maximum Entropy (MaxEnt) classifier can be used to integrate multiple knowledge sources for segmenting recorded speech. In this subsection, we first evaluate the effectiveness of features that have been proposed in previous work, with a focus on features that can be extracted automatically. Second, we examine other knowledge sources that have not been studied systematically in previous work, but which we expect to be good predictors of dialogue segments. In addition, as our ultimate goal is to develop an information retrieval module that can be operated in a fully automatic fashion, we also investigate the impact of automatic speech recognition (ASR) errors on the task of automatic dialogue segmentation.

7.1.2 Related Work

In previous work, the problem of automatic dialogue segmentation is often considered as similar to the problem of topic segmentation. Therefore, research has adopted techniques previously developed to segment topics in text [Kozima, 1993, Hearst, 1997, Reynar, 1998] and in read speech (e.g., broadcast news) [Ponte and Croft, 1997, Allan et al., 1998, Trieschnigg and Kraaij, 2005]. For example, lexical cohesion-based algorithms, such as LCSEG [Galley et al., 2003], or its word frequency-based predecessor TextTiling [Hearst, 1997] capture topic shifts by modelling the similarity of word repetition in adjacent windows. For a more detailed overview, please refer to the AMI/AMIDA State-of-the-art overview report [Hsueh, 2007].

However, recent work has shown that LCSEG is less successful in identifying "agenda-

based conversation segments" (e.g., *presentation*, *group discussion*) that are typically signalled by differences in group activity [Hsueh and Moore, 2006]. This is not surprising since LCSEG considers only lexical cohesion. Previous work has shown that training a segmentation model with features that are extracted from knowledge sources other than words, such as speaker interaction (e.g., overlap rate, pause, and speaker change) [Galley et al., 2003], or participant behaviours, e.g., note taking cues [Banerjee and Rudnicky, 2006], can outperform LCSEG on similar tasks.

In many other fields of research, a variety of features have been identified as indicative of segment boundaries in different types of recorded speech. For example, Brown et al. [Brown et al., 1980] have shown that a discourse segment often starts with relatively high pitched sounds and ends with sounds of pitch within a more compressed range. Passonneau and Litman [Passonneau and Litman, 1993] identified that topic shifts often occur after a pause of relatively long duration. Other prosodic cues (e.g., pitch contour, energy) have been studied for their correlation with story segments in read speech [Tur et al., 2001, Levow, 2004, Christensen et al., 2005] and with theory-based discourse segments in spontaneous speech (e.g., direction-given monologue) [Hirschberg and Nakatani, 1996]. In addition, head and hand/forearm movements are used to detect group-action based segments [McCowan et al., 2005b, Al-Hames et al., 2005].

However, many other features that we expect to signal segment boundaries have not been studied systematically. For instance, speaker intention (i.e., dialogue act types) and conversational context (e.g., speaker role). In addition, although these features are expected to be complementary to one another, few of the previous studies have looked at the question how to use conditional approaches to model the correlation among features.

7.1.3 Meeting Corpus and Structural Discourse Segmentation Annotations

This study aims to explore approaches that can integrate multimodal information to discover implicit semantics from conversation archives. Recently, many natural meetings have been recorded in the context of the ICSI Meetings [Janin et al., 2003], the CALO project[31] [CALO, 2006] and the AMI project [Carletta et al., 2005b]. As our goal is to identify multimodal cues of segmentation in face-to-face conversation, we use the AMI meeting corpus [Carletta et al., 2005b], which includes audio-video recordings, to test our approach. In particular, we are using 50 scenario-based meetings from the AMI corpus, in which participants are assigned to different roles and given specific tasks related to designing a remote control. On average, AMI meetings last 26 minutes, with over 4,700 words transpired. This corpus includes annotation for dialogue segmentation and topic labels. In the annotation process, annotators were given the freedom to subdivide a segment into sub-segments to indicate when the group was discussing a subtopic. Annotators were also given a set of segment descriptions to be used as labels. Annotators were instructed to add a new label only if they could not find a match in the standard set. The set of segment descriptions can be divided to three categories: activity-based (e.g., presentation, discussion), issue-based (e.g., budget, usability), and functional segments (e.g., chitchat, opening, closing). Since the meetings are conducted with an agenda, annotators are expected to find most of the meeting discussion reoccur. Therefore,

[31] *http://www.ai.sri.com/project/CALO*

To evaluate the performance of various features on meeting segmentation, we need to first break a recorded meeting into minimal units, which can vary from sentence chunks to blocks of sentences. In this study, we use *spurts*, that is, consecutive speech with no pause longer than 0.5 seconds, as minimal units.

Then, to examine the difference between the set of features that are characteristic of segmentation at both coarse and fine levels of granularity, this study characterises a dialogue as a sequence of segments that may be further divided into sub-segments. We take the theory-free dialogue segmentation annotations in the corpus and flatten the sub-segment structure and consider only two levels of segmentation: top-level segments and all sub-level segments.[32] We observed that annotators tended to annotate activity-based segments only at the top level, whereas they often included sub-topics when segmenting issue-based segments. For example, a top-level *interface specialist presentation* segment can be divided into *agenda/equipment issues*, *user requirements*, *existing products*, and *look and usability* sub-level segments.

7.1.4 Features Extraction

As reported in Section 7.1.2, there is a wide range of features that are potentially characteristic of segment boundaries. For example, previous research has shown that interlocutors do speak and behave differently when trying to end a discussion and initiate a new one, pause longer than usual when making sure that everyone is ready to move on to a new discussion, use some conventional expressions (e.g., *well*, *okay*, *let's*) when attempting to get everyone's attention about an upcoming new discussion. We expect to find some of them useful for automatic recognition of segment boundaries. The features we explore can be divided into the following five classes:

Conversational Features: We follow Galley et al. [Galley et al., 2003] and extracted a set of conversational features, including the amount of overlapping speech, the amount of silence between speaker segments, speaker activity change, the number of cue words, and the predictions of LCSEG (i.e., the lexical cohesion statistics, the estimated posterior probability, the predicted class).

Lexical Features: Following the "bag of words" representations of documents used for document classification: we back off from high-level descriptions of documents to low-level order-free representations. We compile the list of words that occur more than once in the spurts that have been marked as a top-level or sub-segment boundary in the training set. Each spurt is then represented as a vector space of uni-grams from this list.

Prosodic Features: Prosodic features are suprasegmental features that can be derived from the intonation, rhythm, and lexical stress in speech. Functionally, prosodic features, i.e., intonation, energy, and fundamental frequency (F0), is used to indicate segmentation and saliency [Shriberg et al., 2000, Grosz and Hirschberg, 1992, Liu et al., 2004]. In this study, we follow [Shriberg et al., 2001]'s direct modelling approach to manifest prosodic features, among other things, as duration, pause, speech rate, pitch contour, and energy level. This study utilizes the individual sound files recorded by close-talking far field

[32] We take the spurts which the annotators choose as the beginning of a segment as the topic boundaries. On average, the annotators marked 8.7 top-level segments and 14.6 sub-segments per meeting.

head-mounted microphones and the cross-talking sound files using the desktop microphones provided in the AMI corpus. As the first step towards extracting prosodic features from these sound files, we use Snack Sound Toolkit to compute a list of pitch and energy values delimited by frames of 10 ms, using the normalised cross correlation function. Then we apply a piecewise linearisation procedure to remove the outliers and average the linearised values of the units within the time frame of a word as its pitch value. Pitch contour of a discourse unit is approximated by measuring the pitch slope at multiple points within a discourse unit, e.g., the overall slope, the first and last 100 and 200 ms, first and second half, and each quarter of a discourse unit. The rate of speech is calculated as the number of words spoken per second and number of syllables per second. We use Festival's speech synthesis front-end to return phonemes and syllabification information.

Prosodic features in context are also considered. As literature has shown the benefits of including immediate prosodic contexts, this study includes features that provide information about the preceding and following discourse units. Table 61 contains a list of prosodic context features used in this study.

Type	Feature
Duration	Number of words spoken in current, previous and next discourse unit Duration (in seconds) of current, previous and next discourse unit
Pause	Amount of silence (in seconds) preceding a discourse unit Amount of silence (in seconds) following a discourse unit
Speech rate	Number of words spoken per second in current, previous and next discourse unit Number of syllables per second in current, previous and next discourse unit
Energy	Average energy level Average energy level in the first, second, third, and fourth quarter of a discourse unit
Pitch	Maximum and minimum F0 overall slope and variance Slope and variance at the first 100 and 200 ms and last 100 and 200 ms Slope and variance at the first and second half Slope and variance at each quarter of the discourse unit

Table 61: *Prosodic features and its contexts.*

Motion Features: We measure the magnitude of relevant movements in the meeting room using a system developed in TNO which detects movements directly from video recordings in frames of 40 ms. Of special interest are the frontal shots as recorded by the close up cameras, the hand movements as recorded by the overview cameras, and shots of the areas of the room where presentations are made. We then average the magnitude of movements over the frames within a spurt as its feature value.

Contextual Features: These include dialogue act type[33] and speaker role (e.g., project manager, marketing expert). As each spurt may consist of multiple dialogue acts, we represent each spurt as a vector of dialogue act types, wherein a component is 1 or 0

[33]In the annotations, each dialogue act is classified as one of 15 types, including acts about information exchange (e.g., Inform), acts about possible actions (e.g., Suggest), acts whose primary purpose is to smooth the social functioning (e.g., Be-positive), acts that are commenting on previous discussion (e.g., Elicit-Assessment, Elicit-inform, Elicit-suggest), and acts that allow complete segmentation (e.g., Stall).

depending on whether the type occurs in the spurt.

7.1.5 Multimodal Integration Experiment and Feature Effects

Previous work has used MaxEnt models for sentence and topic segmentation and shown that conditional approaches can yield competitive results on these tasks [Christensen et al., 2005, Hsueh and Moore, 2006]. In this study, we also use a MaxEnt classifier[34] for dialogue segmentation under the typical supervised learning scheme, that is, to train the classifier to maximise the conditional likelihood over the training data and then to use the trained model to predict whether an unseen spurt in the test set is a segment boundary or not. Because continuous features have to be discretized for MaxEnt, we applied a histogram binning approach, which divides the value range into *N* intervals that contain an equal number of counts as specified in the histogram, to discretize the data.

The first question we want to address is whether the different types of characteristic multimodal features can be integrated, using the conditional MaxEnt model, to automatically detect segment boundaries. In this study, we use a set of 50 meetings, which consists of 17,977 spurts. Among these spurts, only 1.7% and 3.3% are top-level and sub-segment boundaries. For our experiments we use 10-fold cross validation. The baseline is the result obtained by using LCSEG, an unsupervised approach exploiting only lexical cohesion statistics.

	TOP		SUB	
Error Rate	PK	WD	PK	WD
BASELINE(LCSEG)	0.40	0.49	0.40	0.47
MAXENT(CONV)	0.34	0.34	0.37	0.37
MAXENT(ALL)	0.30	0.33	0.34	0.36

Table 62: Compare the result of MaxEnt models trained with only conversational features (CONV) and with all available features (ALL).

Table 62 shows the results obtained by using the same set of conversational (CONV) features used in previous work [Galley et al., 2003, Hsueh and Moore, 2006], and results obtained by using all the available features (ALL). The evaluation metrics PK and WD are conventional measures of error rates in segmentation. Pk [Beeferman et al., 1999] is the probability that two utterances drawn randomly from a document (in our case, a meeting transcript) are incorrectly identified as belonging to the same topic segment. WindowDiff (Wd) [Pevzner and Hearst, 2002] calculates the error rate by moving a sliding window across the meeting transcript counting the number of times the hypothesized and reference segment boundaries are different.

In Row 2 of Table 62, we see that using a MaxEnt classifier trained on the conversational features (CONV) alone improves over the LCSEG baseline by 15.3% for top-level segments and 6.8% for sub-level segements. Row 3 shows that combining additional knowledge sources, including lexical features (LX1) and the non-verbal features, prosody (PROS), motion (MOT), and context (CTXT), yields a further improvement (of 8.8% for

[34]The parameters of the MaxEnt classifier are optimised using Limited-Memory Variable Metrics.

top-level segmentation and 5.4% for sub-level segmentation) over the model trained on conversational features.

The second question we want to address is which knowledge sources (and combinations) are good predictors for segment boundaries. In this round of experiments, we evaluate the performance of different feature combinations. Table 63 further illustrates the impact of each feature class on the error rate metrics (PK/WD). In addition, as the PK and WD score do not reflect the magnitude of over- or under-prediction, we also report on the average number of hypothesized segment boundaries (Hyp). The number of reference segments in the annotations is 8.7 at the top-level and 14.6 at the sub-level.

	TOP			SUB		
	Hyp	PK	WD	Hyp	PK	WD
BASELINE (LCSEG)	17.6	0.40	0.49	17.6	0.40	0.47
LX1	61.2	0.53	0.72	65.1	0.49	0.66
CONV	3.1	0.34	0.34	2.9	0.37	0.37
PROS	2.3	0.35	0.35	2.5	0.37	0.37
MOT	96.2	0.36	0.40	96.2	0.38	0.41
CTXT	2.6	0.34	0.34	2.2	0.37	0.37
ALL	7.7	0.29	0.33	7.6	0.35	0.38

Table 63: Effects of individual feature classes and their combination on detecting segment boundaries.

Rows 2-6 in Table 63 show the results of models trained with each individual feature class. We performed a one-way ANOVA to examine the effect of different feature classes. The ANOVA suggests a reliable effect of feature class ($F(5, 54) = 36.1$; $p < .001$). We performed post-hoc tests (Tukey HSD) to test for significant differences.

Analysis shows that the model that is trained with lexical features alone (LX1) performs significantly worse than the LCSEG baseline ($p < .001$). This is due to the fact that cue words, such as *okay* and *now*, learned from the training data to signal segment boundaries, are often used for non-discourse purposes, such as making a semantic contribution to an utterance.[35] Thus, we hypothesise that these ambiguous cue words have led the LX1 model to over-predict. Row 7 further shows that when all available features (including LX1) are used, the combined model (ALL) yields performance that is significantly better than that obtained with individual feature classes ($F(5, 54) = 32.2$; $p < .001$).

Table 64 illustrates the error rate change (i.e., increased or decreased PK and WD score)[36] that is incurred by leaving out one feature class from the ALL model. Results show that CONV, PROS, MOTION and CTXT can be taken out from the ALL model individually without increasing the error rate significantly.[37] Moreover, the combined models always perform better than the LX1 model ($p < .01$), cf. Table 63.

[35] Hirschberg and Litman [Hirschberg and Litman, 1987] have proposed to discriminate the different uses intonationally.

[36] Note that the increase in error rate indicates performance degradation, and vice versa.

[37] Sign tests were used to test for significant differences between means in each fold of cross validation.

	TOP			SUB		
	Hyp	PK	WD	Hyp	PK	WD
ALL	7.7	0.29	0.33	7.6	0.35	0.38
ALL-LX1	3.9	0.35	0.35	3.5	0.37	0.38
ALL-CONV	6.6	0.30	0.34	6.8	0.35	0.37
ALL-PROS	5.6	0.29	0.31	7.4	0.33	0.35
ALL-MOTION	7.5	0.30	0.35	7.3	0.35	0.37
ALL-CTXT	7.2	0.29	0.33	6.7	0.36	0.38

Table 64: Performance change of taking out each individual feature class from the ALL model.

This suggests that the non-lexical feature classes are complementary to LX1, and thus it is essential to incorporate some, but not necessarily all, of the non-lexical classes into the model.

	TOP			SUB		
	Hyp	PK	WD	Hyp	PK	WD
LX1	61.2	0.53	0.72	65.1	0.49	0.66
MOT	96.2	0.36	0.40	96.2	0.38	0.41
LX1+CONV	5.3	0.27	0.30	6.9	0.32	0.35
LX1+PROS	6.2	0.30	0.33	7.3	0.36	0.38
LX1+MOT	20.2	0.39	0.49	24.8	0.39	0.47
LX1+CTXT	6.3	0.28	0.31	7.2	0.33	0.35
MOT+PROS	62.0	0.34	0.34	62.1	0.37	0.37
MOT+CTXT	2.7	0.33	0.33	2.3	0.37	0.37

Table 65: Effects of combining complementary features on detecting segment boundaries.

Table 65 further illustrates the performance of different feature combinations on detecting segment boundaries. By subtracting the PK or WD score in Row 1, the LX1 model, from that in Rows 3-6, we can tell how essential each of the non-lexical classes is to be combined with LX1 into one model. Results show that CONV is the most essential, followed by CTXT, PROS and MOT. The advantage of incorporating the non-lexical feature classes is also shown in the noticeably reduced number of over-predictions as compared to that of the LX1 model.

The column Hyp reported in this table can be used to determine which algorithm results in a better approximation in terms of the number of hypothesized segments. Combining any of the non-lexical feature classes reduces the number of over-predictions by LX1 noticeably. Further comparison of performance improvement across the top-level and the sub-level segmentation models suggests that little difference exists between these results. However, none of the feature combinations has yielded a good gauge at the number of sub-level segment boundaries.

7.1.6 Degradation Using ASR

The third question we want to address here is whether using the output of ASR will cause significant degradation to the performance of the segmentation approaches. The ASR transcripts used in this experiment are obtained using standard technology including HMM based acoustic modelling and N-gram based language models [Hain et al., 2005]. The average word error rates (WER) are 39.1%.

The ASR system used a vocabulary of 50,000 words, together with a trigram language model trained on a combination of in-domain meeting data, related texts found by web search, conversational telephone speech (CTS) transcripts and broadcast news transcripts (about 10^9 words in total), resulting in a test-set perplexity of about 80. The acoustic models comprised a set of context-dependent hidden Markov models, using gaussian mixture model output distributions. These were initially trained on CTS acoustic training data, and were adapted to the ICSI meetings domain using maximum a posteriori (MAP) adaptation. Further adaptation to individual speakers was achieved using vocal tract length normalisation and maximum likelihood linear regression. A four-fold cross-validation technique was employed: four recognisers were trained, with each employing 75% of the meetings as acoustic and language model training data, and then used to recognise the remaining 25% of the meetings.

We also applied a word alignment algorithm to ensure that the number of words in the ASR transcripts is the same as that in the human-produced transcripts. In this way we can compare the PK and WD metrics obtained on the ASR outputs directly with that on the human transcripts.

In this study, we again use a set of 50 meetings and 10-fold cross validation. We compare the performance of the reference models, which are trained on human transcripts and tested on human transcripts, with that of the ASR models, which are trained on ASR transcripts and tested on ASR transcripts. Table 66 shows that despite the word recognition errors, none of the LCSEG, the MaxEnt models trained with conversational features, and the MaxEnt models trained with all available features perform significantly worse on ASR transcripts than on reference transcripts. One possible explanation for this, which we have observed in our corpus, is that the ASR system is likely to mis-recognise different occurrences of words in the same way, and thus the lexical cohesion statistic, which captures the similarity of word repetition between two adjacency windows, is also likely to remain unchanged. In addition, when the models are trained with other features that are not affected by the recognition errors, such as pause and overlap, the negative impacts of recognition errors are further reduced to an insignificant level.

7.1.7 Discussion

The results in Section 7.1.5 show the benefits of including additional knowledge sources for recognising segment boundaries. The next question to be addressed is what features in these sources are most useful for recognition. To provide a qualitative account of the segmentation cues, we performed an analysis to determine whether each proposed feature discriminates the class of segment boundaries. Previous work has identified statistical measures (e.g., Log Likelihood ratio) that are useful for determining the statistical asso-

	TOP		SUB	
Error Rate	PK	WD	PK	WD
LCSEG(REF)	0.45	0.57	0.42	0.47
LCSEG(ASR)	0.45	0.58	0.40	0.47
MAXENT-CONV(REF)	0.34	0.34	0.37	0.37
MAXENT-CONV(ASR)	0.34	0.33	0.38	0.38
MAXENT-ALL(REF)	0.30	0.33	0.34	0.36
MAXENT-ALL(ASR)	0.31	0.34	0.34	0.37

Table 66: Effects of word recognition errors on detecting segments boundaries.

ciation strength (relevance) of the occurrence of an n-gram feature to target class [Hsueh and Moore, 2006]. Here we extend that study to calculate the Log Likelihood relevance of all of the features used in the experiments, and use the statistics to rank the features.

Our analysis shows that people do speak and behave differently near segment boundaries. Some of the identified segmentation cues match previous findings. For example, a segment is likely to start with higher pitched sounds [Brown et al., 1980, Ayers, 1994] and a lower rate of speech [Lehiste, 1980]. Also, interlocutors pause longer than usual to make sure that everyone is ready to move on to a new discussion [Brown et al., 1980, Passonneau and Litman, 1993] and use some conventional expressions (e.g., *now*, *okay*, *let's*, *um*, *so*).

Our analysis also identified segmentation cues that have not been mentioned in previous research. For example, interlocutors do not move around a lot when a new discussion is brought up; interlocutors mention agenda items (e.g., *presentation*, *meeting*) or content words more often when initiating a new discussion. Also, from the analysis of current dialogue act types and their immediate contexts, we also observe that at segment boundaries interlocutors do the following more often than usual: start speaking before they are ready (*Stall*), give information (*Inform*), elicit an assessment of what has been said so far (*Elicit-assessment*), or act to smooth social functioning and make the group happier (*Be-positive*).

7.1.8 Conclusion

This study explores the use of features from multiple knowledge sources (i.e., words, prosody, motion, interaction cues, speaker intention and role) for developing an automatic segmentation component in spontaneous, multiparty conversational speech. In particular, we addressed the following questions: (1) Can a MaxEnt classifier integrate the potentially characteristic multimodal features for automatic dialogue segmentation? (2) What are the most discriminative knowledge sources for detecting segment boundaries? (3) Does the use of ASR transcription significantly degrade the performance of a segmentation model?

First of all, our results show that a well performing MaxEnt model can be trained with available knowledge sources. Our results improve on previous work, which uses only conversational features, by 8.8% for top-level segmentation and 5.4% for sub-level segmentation. Analysis of the effectiveness of the various features shows that lexical features

(i.e., cue words) are the most essential feature class to be combined into the segmentation model. However, lexical features must be combined with other features, in particular, conversational features (i.e., lexical cohesion, overlap, pause, speaker change), to train well performing models.

In addition, many of the non-lexical feature classes, including those that have been identified as indicative of segment boundaries in previous work (e.g., prosody) and those that we hypothesized as good predictors of segment boundaries (e.g., motion, context), are not beneficial for recognising boundaries when used in isolation. However, these non-lexical features are useful when combined with lexical features, as the presence of the non-lexical features can balance the tendency of models trained with lexical cues alone to over-predict. We believe there are several reasons for this. First, the presence of the non-verbal features in the model can balance off the over-fitting tendency of models trained with lexical cues. Second, because there is an interaction effect between these non-verbal features, by combining these features we can further improve the performance of the segmentation models.

Experiments also show that it is possible to segment conversational speech directly on the ASR outputs. These results encouragingly show that we can segment conversational speech using features extracted from different knowledge sources, and in turn, facilitate the development of a fully automatic segmentation component for multimedia archives.

With the segmentation models developed and discriminative knowledge sources identified, a remaining question is whether it is possible to automatically select the discriminative features for recognition. This is particularly important for prosodic features, because the direct modelling approach we adopted resulted in a large number of features. We expect that by applying feature selection methods we can further improve the performance of automatic segmentation models. In the field of machine learning and pattern analysis, many methods and selection criteria have been proposed. Our next step will be to examine the effectiveness of these methods for the task of automatic segmentation. Also, we will further explore how to choose the best performing ensemble of knowledge sources so as to facilitate automatic selection of knowledge sources to be included.

7.2 Machine learning and time series analysis approaches to the segmentation of meeting discourse

In this section, building upon previous work, we initially set out to thoroughly compare two machine learning approaches to the discourse segmentation problem. We compare a non-sequential approach with an explicitly sequential approach: maximum entropy vs. conditional random fields. Our research question was whether discourse segmentation benefits from sequential, not strictly local information. Using two widely used evaluation metrics (Pk and WindowDiff), we find that CRFs under certain circumstances outperform maximum entropy models. Yet, actual inspection of the results reveals a fundamental shortcoming of these metrics: recall and precision are not penalized as much as it should be, and both under- and oversegmentation frequently occur. In a subsequent batch of experiments, we therefore optimize both a machine learning method and a time series analysis approach for a more articulate error metric that allows for balancing recall and precision, the Pr_{error} metric of [Georgescul et al., 2006]. Prior to introducing the experi-

ments and results, we discuss the problem of evaluating discourse segmenters.

7.3 Evaluation metrics for discourse segmentation

A problem any machine learning approach to discourse segmentation has to face is class imbalance: a skewed class distribution. Typically, the number of actual segment boundaries is tiny compared with respect to the number of possible boundaries. The main goal therefore in ML approaches is to implicitly downsize the latter quantity, e.g. by using very informative features. The original LCSEG ([Galley et al., 2003]) and TextTiling ([Hearst, 1994]) approaches to discourse segmentation view discourse segmentation as a time series problem amenable to signal processing. Both approaches look for significant patterns in a quasi-temporal representation of the sequential text data, and recast the problem of class imbalance to a detection problem: finding significantly disruptive patterns that indicate a topic shift.

When classes are distributed fairly even, i.e. $P(c_i \mid T) \approx P(c_j \mid T), i \neq j$ for any two classes c and training data T, accuracy is an acceptable measure of quality for a classifier. But when class distributions are highly skewed, recall, precision and harmonic means of these like the F_β-score are better measures. Discourse segmentation, segmenting a text into separate topics, is a typically class-imbalanced task. The number of linguistic units on which segmentation is based (like sentences) typically by far exceeds the number of actual topics. Consequently, optimizing a classifier for accuracy would automatically favor a majority classifier that labels all sentences as not opening a topic. Optimization for the classical notions of recall and precision would not work well here either: for instance, a discourse segmenter that always predicts a topic boundary close but not exactly corresponding to the ground truth prediction would produce zero recall and precision, while its performance can actually be quite good. Specific measures like Pk and WindowDiff ([Pevzner and Hearst, 2002]) compute recall and precision in a fixed-size window to alleviate this problem, but they do not penalize false negatives and false positives in the same way. For discourse segmentation, false negatives probably should be treated on a par with false positives, to avoid undersegmentation. To this end, [Georgescul et al., 2006] proposed a new, cost-based metric called Pr_{error}:

$$Pr_{error} = C_{miss} \cdot Pr_{miss} + C_{fa} \cdot Pr_{fa} \tag{29}$$

Here, C_{miss} and C_{fa} are cost terms for false negatives and false alarms; Pr_{miss} is the probability that a predicted segmentation contains less boundaries than the ground truth segmentation in a certain interval of linguistic units (like words); Pr_{fa} denotes the probability that the predicted segmentation in a given interval contains less boundaries than the ground truth segmentation. We refer the reader to [Georgescul et al., 2006] for further details and the exact computation of these probabilities. Using explicit penalty terms for false negatives and false positives allows for balancing recall and precision, and, under the standard setting, weights false negatives exactly the same as false positives.

7.4 Optimizing for Pk and WindowDiff: non-sequential and sequential machine learning algorithms

7.4.1 Maximum Entropy models

A range of multimodal features have been show indispensable to the recognition of meeting discourse segments (see Section 7.1). In our previous work, we have used an exponential model to combine the various features, each represented as an indicator function of the neighbouring multimodal context of a discourse unit and its segment boundary class.[38]

Formally, the observed context of a discourse unit taking place at t can be written as $o_t = f_1(d, c), f_2(d, c), ...f_n(d, c)$. The exponential model thus can be expressed as a combination of these binary valued functions $f_i(d, c)$ as follows,

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_{i=1}^{n} \lambda_i f_i(d, c)) \tag{30}$$

wherein λ is a real-valued weight associated with f_i, and $Z(d)$ is a constant used as a normalisation term. λ_i can be seen as a measure of the importance of including the ith feature or that of not including it, if λ_i is negative. In the training phase of a log-linear Maximum Entropy (MaxEnt) model, a constrained optimisation procedure is then applied to find $argmax_c P(c|d)$ with the constraint that the likelihood of the data D given the model is maximised.

In the testing phase of the MaxEnt model, the segmentation task is then operated as a series of binary decisions, each determining whether a potential segment boundary site (i.e., the end of each discourse unit) belongs to the positive boundary class or not. [Hsueh and Moore, 2007b] have achieved state-of-the-art success on segmenting meetings by training the MaxEnt models with a combination of multimodal features, ranging from the occurrence counts of discourse connectives to the amount of head movement and gesture of speakers. Despite the success, the MaxEnt modelling approach has some shortcomings. One limitation of the MaxEnt model lies in the fact that these decisions are made independently, without taking into account temporal coherence.

There are many other methods that can make sequential decisions (at least in the local context), for example, the Hidden Markov model (HMM) [van Mulbregt et al., 1999] and its variants (e.g., aspect HMM (AHMM)) [Blei and Moreno, 2001]. Applying these methods involves two major steps. First, k topic models are constructed from large corpus (such as Wall Street Journal articles and CNN transcribed broadcasts). Then, the k topic models are used to compute the emission probability of an observed discourse unit with respect to each of the k topics. As the transition probabilities (including self-loop probability) of these topics can be determined from specific feature functions, it is thus feasible to estimate parameters on variants of HMMs. However, these previous works only focus on modelling topical features, leaving many of the multimodal features that have been

[38] Initial experiments with a support vector machine (SVM) using these features were disappointing, and led to poor results: by optimizing the hyperparameters of these classifiers for accuracy-based metrics like Pk and WindowDiff, majority effects occur, and the tiny amount of positive cases (topic shifts) as compared to negative cases (no topic shifts) leads the classifier astray. We will not report these results here, but we will address SVMs again in subsection 7.5.1.

evidenced to be useful in meeting discourse segmentation untouched. Moreover, none of the previous work has performed a non-local optimisation of the sequence of assigned decisions.

Previous work has used MaxEnt models for sentence and discourse segmentation [Christensen et al., 2005, Hsueh and Moore, 2006] and showed that conditional approaches can yield competitive results on these tasks. In this study, we also use a MaxEnt classifier [Zhang, 2006b] for dialogue segmentation under the typical supervised learning scheme, that is, to train the classifier to maximise the conditional likelihood over the training data and then to use the model trained in predicting whether an unseen spurt in the test set is a segment boundary or not. For those features that are continuous variables, we apply the HIS binning approach to discretize the data.

7.4.2 Conditional Random Fields

Since meeting discourse segmentation, as any other sequential processing task, would benefit from a global optimisation of the sequence of assigned boundary classes ('yes/no'), an interesting approach is to apply such a sequential decision making approach to find meeting segments from multimodal features beyond words. Conditional Random Fields (CRF), a generalised version of the HMM approach which relaxes some of its assumptions on the input and output sequence, is therefore a natural environment to study this question. As opposed to the HMM approach which enforces constant transition probabilities, CRF allows transition probabilities to be determined by arbitrary functions that are derived from the observed feature values in the input sequence and in turn change the transition probabilities accordingly. (???) In this subsection, we will evaluate the sequential CRF with the non-sequential MaxEnt model on the task of integrating multimodal information for meeting discourse segmentation.

CRF has been shown to perform well in many fields, including information extraction modules [Sarawagi and Cohen, 2004, Grenager et al., 2005], NLP (e.g., POS tagging [Lafferty et al., 2001b]), and spoken language understanding (e.g., automatic disfluency detection [Liu et al., 2005b]). Technically, CRFs are undirected graphical models, similar to undirected Markov Chains, that are able to model contextual dependencies that are beyond the capabilities of Hidden Markov Models. For instance, CRFs are able to look forward as well as backward in a sequence of observations. So, CRFs are undirected graphical models globally conditioned on the sequence of observations.

As a hybrid between Maximum Entropy and Hidden Markov sequence optimisation, CRFs effectively target the *label-bias problem*: the problem of cascaded errors when previous predictions of an ML algorithm are used as features for subsequent analysis steps. Normalisation over an entire state sequence leads to corrections of local errors. Given a undirected graph $G = (V, E)$, with V the set of vertexes, and E the set of vertices, we can let the label sequence $\mathbf{y}$ be indexed by the vertices of G: $\mathbf{y} = (\mathbf{y_v})_{v \in V}$. The pair $(\mathbf{x}, \mathbf{y})$, with $\mathbf{x}$ a data or observation sequence of features, is a CRF whenever the random variables $\mathbf{y_v}$ (conditional on $\mathbf{x}$) obey the Markov property with respect to G:

$$P(\mathbf{y}_v \mid \mathbf{x}, \mathbf{y}_w, w \neq v) = P(\mathbf{y}_v \mid \mathbf{x}, \mathbf{y}_w, w \sim v)$$

Here, $w \sim v$ means w and v are neighbours in the graph G. CRF's allow features to become conditioned on state (label) information (Lafferty *et al.* [Lafferty et al., 2001b]); a feature f may produce value 1 if state y_{i-1} (a class label) is A, and 0 otherwise, for instance. This connection between feature values and states is the discriminative aspect of CRFs. Viterbi-style best path algorithms can be used to produce the most likely state sequence explaining the observation sequence $\mathbf{x}$. In this study, we used the CRF++ toolkit package [Kudo, 2006] and applied both FX and MDL binning approaches for discretizing features.

Feature Discretization Continuous features will have to be discretized for MaxEnt and CRF. We applied three binning methods. The first (called 'FX') is a fixed-size ('equal width') binning method. For every feature, it divides the value range into N intervals of equal size so as to form a uniform grid. For A and B the lowest and highest values of a given feature, the width of intervals will be $W = (B - A) + 1/N$. This straightforward method is acceptable for uniformly distributed data, but outliers are not handled well. To avoid the problem, the second (called 'HIS') divides the value range into N intervals that contain equal number of counts as specified in the histogram. The third (called 'MDL') is a binning method based on the Minimum Description Length method of Fayyad and Irani [U.M.Fayyad and Irani, 1995]. For every feature, split points in the value range are recursively computed that have high information gain, until a threshold established by MDL principles is reached.

Class Space Expansion We decided to expand the class space from 2 classes (yes/no) to a larger class space. Recent work by van den Bosch [van den Bosch, 2004] has demonstrated beneficiary effects for n-gram class expansion. The idea is that we extract from our training data n-grams of classes. We replace the unary class symbols by these n-grams. For instance, given the following training data (where 'instance' means: a separate case)

- (instance 1) $f_1, \ldots, f_n, C_1$
- (instance 2) $f_1, \ldots, f_n, C_2$
- (instance 3) $f_1, \ldots, f_n, C_3$

we can replace the original class for instance 2 (C_2) with $C_1 + C_2 + C_3$. This is a composite class symbol consiting of two extra symbols: the classes for instances 2 and 3. After classification, the right element of the predicted trigram for an instance $i-1$ will be a vote for the class of instance i; similarly, the leftmost element of the class trigram of instance $i + 1$ will be a vote for the class Our intuition for CRFs is that sequence optimisation works better for multiclass sequences than for binary sequences. Given the limitation of CRF++ to bi-gram models, a binary class system just does not provide enough information to estimate a useful model $P(c_i \mid c_{i-1})$. We are using a feedback loop in the sense that the data is processed spurt -by-spurt, at each step advancing a 5-left-5-right spurt window with one spurt. On average, a number of 9 (2^3+1) classes was found after expansion. An example is the following class set (1 corresponding to 'yes' and 2 to 'no' boundary): $\{1+2, 1+2+2, 2+2+2, 2+2+1, 2+1+2, 2+1+1, 1+1+2, 1+2+1, 1+1+1\}$. In order to translate the results from classification back to uni-gram classes, we did not

apply voting but instead selected the middle element of every trigram class predicted. The reason for this is that there are some drawbacks associated with voting using a feedback loop as we are using here: the final voting step can be seen as a purely local form of error correction, where errors are resolved on the basis of local, uncorrelated decisions. This conflicts with the global optimisation strategy of CRFs.[39]

Feature Interaction We modeled feature interaction between 5 preceding and consecutive datapoints (spurts) in the training data, allowing for unigram, bigram and trigram combination of feature values. Our initial feature space is the same as the feature space used for the maximum entropy classifier. That is, a window of size 10 is used to generate a yes/no decision (yes: a topic boundary occurs).

7.4.3 Experiments and Results

In our experiments, the hyperparameters of the MaxEnt and CRF classifiers have been optimized for the error metrics Pk and WindowDiff. Specifically, the CRF classifiers were optimized using a grid search through hyperparameter space, applied to heldout data and the MaxEnt classifiers by Limited-Memory Variable Metrics (L-BFGs) [Malouf, 2002]), which converged surprisingly to default parameter settings. [40]

Table 67 shows the results obtained by using the same set of conversational (CONV) features as that used in previous work [Galley et al., 2003, Hsueh and Moore, 2006] and that obtained by using all the available features, including lexical (LX1), conversational (CONV), prosodic (PROS), MOTION, and contextual (CX) features. (See Section 7.1.4 for descriptions of each available feature class.)

		TOP			SUB		
		Hyp	Pk	Wd	Hyp	Pk	Wd
BASELINE		17.6	0.40	0.49	17.6	0.40	0.47
CONV	MaxEnt	3.1	0.34	0.34	2.9	0.37	0.37
	CRF(FX)	15.4	0.31	0.34	30.7	0.36	0.38
	CRF(MDL)	13.2	0.28	0.32	16.3	0.33	0.35
ALL	MaxEnt	5.6	0.31	0.32	6.4	0.35	0.37
	CRF(MDL)	15.9	0.28	0.33	15	0.34	0.36

Table 67: *The result of MaxEnt models and CRFs on detecting segments boundaries. The column Hyp is the average number of hypothesized segment boundaries. The average number of reference segments in the annotations is 8.7 at the top-level and 14.6 at the sub-level. Errors are measured with Pk and WindowDiff (Wd).*

Row 2-4 in Table 67 show the result of a MaxEnt classifier and that of CRFs trained on the FX- and MDL-binning features respectively. These results suggest that when only

[39] An alternative is that, for a certain instance, we not only vote for classes on the basis of neighbouring n-gram classes, but repeat this voting process for every window the instance occurs in. For instance, given a window size of n, every instance occurs in exactly n windows.

[40] Note that the CRFs and MaxEnt classifiers were optimised using different procedures, so the results of models we used for comparison may not be at their optimal operating point.

conversational features (CONV) are used, sequential CRFs outperform non-sequential MaxEnt models that do not exploit global sequence optimisation strategies.

Yet, inspection of the segmentation results indicates that Pk and WindowDiff are not useful metrics to optimize a classifier for; quite often, massive undersegmentation using the MaxEnt classifier was attested. CRFs, on the other hand, seemed to generate many false alarms. Apparently, both low recall and low precision are not penalized severely by Pk and WindowDiff, which is in line with the findings of [Georgescul et al., 2006].

7.4.4 Conclusion

In this subsection we compared conditional random fields with maximum entropy models. We found evidence for the utility of sequential information expressed through feature interactions: the CRF classifier compares favorably from the perspective of Pk and WindowDiff with the maximum entropy classifier using the same features. Both classifiers do not seem to produce a very useful type of segmentation, though, when being optimized for Pk and WindowDiff. Specifically, undersegmentation is attested for the maximum entropy classifier quite frequently (low recall), and the number of false positives for the CRF classifier is relatively high (low precision). Optimizing a classifier for Pk and WindowDiff is not a good idea, as low recall and low precision are not handled well by these metrics.

We found that training a CRF is quite time consuming, and overall, the models induced generate relatively many false alarms. In the next section, we set out to optimize classifiers for an alternative, recently proposed error metric: the Pr_{error} proposed by [Georgescul et al., 2006].

7.5 SVM classification and lowbow segmentation

In this section, we evaluate a machine learning approach as well as a time series approach to AMI meeting transcript segmentation. We first present experimental results obtained with SVM classifiers optimized for the Pr_{error} metric. Next, we turn to a locally weighted bag of words approach to segmentation.

7.5.1 SVM optimized for Pr_{error}

The hyperparameter optimization algorithm ECE described in [Raaijmakers, 2007a], was used to optimize SVM classifiers (RBF kernel machines) on heldout data taken from a dataset of 50 AMI meetings annotated for main topic structure. While our previous attempts at operationalizing SVM's were unfruitful, due to the forementioned majority effects, we now explicitly optimized the RBF kernel parameters and the cost parameter for Pr_{error}.

We compared the Pr_{error} results[41] to four different baselines: A- (generating negative classes only), A+ (generating positive classes only), R (generating random positive and negative classes) and R-n (generating random classes, with n positive classes, where n is the number of known segments in the test data). We set C_{miss} and C_{fa} to 0.5 and k to

[41] A java implementation for computing Pr_{error} is available from [geo]

Fold	Pr_{error}	A-	A+	R	R-n
1	.35	0.5	0.5	0.5	0.48
2	.32	0.5	0.5	0.498	0.499
3	.39	0.5	0.5	0.498	0.49
4	.38	0.5	0.5	0.498	0.483
5	.36	0.5	0.5	0.5	0.47
6	.36	0.5	0.5	0.499	0.51
7	.34	0.5	0.5	0.5	0.489
8	.38	0.5	0.5	0.499	0.486
9	.36	0.5	0.5	0.498	0.499
10	.35	0.5	0.5	0.497	0.485
Average	.36	0.5	0.5	0.499	0.484

Figure 35: Discourse segmentation results with SVM.

half of the average number of words per segment in the training data, in order to penalize undersegmentation as much as oversegmentation.

Results are listed in table 35. The SVM classifiers significantly outperform the 4 baselines, and produce scores comparable to the ones reported by [Georgescul et al., 2006] on ICSI meeting data.

7.5.2 Lowbow optimized for Pr_{error}

In this subsection, we apply the recently developed lowbow technique of [Lebanon et al., 2007] to AMI discourse segmentation. Lowbow, short for 'locally weighted bag of words' can be viewed as a histogram-based LCSEG-style approach. A local histogram of terms is sequenced through a document, and a kernel density estimator is fitted to the changes measured in the histogram statistics as this 'bag of words' moves through the document. Put differently, a local statistical model is fitted at different document positions with a non-parametric kernel smoothing technique. This produces a set of local models that is representable with a smooth curve called a *velocity plot*. Smoothing the kernel function amounts to smoothing the curve. Peaks in the curve indicate significant changes in the lowbow statistics, probably indicating topic shifts.

Experiments and results Using the toolbox made available by [yma], we applied lowbow experiments to 15 AMI meetings annotated for main topic structure, with word-based representations. We used no information other than the Porter-stemmed word forms in the meeting transcripts. We smoothed the kernel with a medium scale factor (0.04), taken from array of possible values [0.01...1],observing that this produced the best Pr_{error} results. The tangent values of the velocity plot were extracted, after which we used a specially implemented peak detector to extract the peaks from these values. Figure 36 displays a sample velocity graph for a meeting and the peaks identified by the peak extraction algorithm. We assigned a topic break to every maximum identified by the peak extractor, and finally applied Georgescul's Java implementation of Pr_{error} with k set to half of the average number of words per segment and C_{miss} and C_{fa} to 0.5 as well as to the

results produced by LCSEG with standard parameters for this data. Results are listed in table 7.5.2. Lowbow was significantly better than LCSEG (1-tailed t-test, $p < .02$). The average number of lowbow-produced segments per meeting was 7.3, with an average of 8.5 in the ground truth data.

	LOWBOW	LCSEG
15 AMI	**34.43**	42.93

Table 68: Average PR_Error for 15 AMI meetings for LCSEG and LOWBOW.

Conclusions Our results indicate that the lowbow strategy constitutes an interesting alternative to machine learning methods. Foremost, the class imbalance problem does not come into play here. Being solidly rooted in the statistical theory of Riemannian manifolds and multinomial document representations, the method amends itself to more finegrained document representations that more accurately approximate the underlying information geometry of this data. For instance, in the work reported in [Raaijmakers and Kraaij, 2008], we found that switching from word forms to character n-grams has benificary effects on the bias of a multinomial classifier. It is quite likely that the noisy and highly elliptic vocabulary of natural meetings is better modelled with character n-grams as well, even under the time series analysis-based lowbow approach. In subsequent work, we will address this issue in more detail, as well as optimizing the kernel scale factors for Pr_{error}. Integrating multimodal features with lowbow is not possible. Yet, the decisions of the lowbow strategy, and, in fact, the entire lowbow feature space itself, can be easily merged with the feature space of an SVM, MaxEnt or CRF machine learning approach.

7.5.3 Conclusions

In this section, we have applied 4 different techniques to the problem of meeting discourse segmentation. We started off with a comparison with two classifiers optimized for the error metrics Pk and WindowDiff, a nonsequential maximum entropy model and a sequential conditional random field. Whereas the latter produced better Pk and WindowDiff error scores compared to the former, evaluation on the basis of these two error metrics was found not to be useful in practice, as both undersegmentation and oversegmentation are not heavily penalized, and were indeed attested in practice. As we are aiming for practical systems, classifier tuning based on these metrics seems not a wise option. Subsequently, we turned to a recently proposed error metric, Pr_{error}, and used it to optimize a kernel machine and a lowbow time series analysis approach. We found that both the SVM classifier and the lowbow approach produced Pr_{error} scores comparable to scores reported on ICSI data. The lowbow approach is particularly interesting in not assuming any training data at all, and significantly outperformed LCSEG, a competitive time series analysis approach to discourse segmentation.

7.6 Online Segmentation of Meeting Discourse

7.6.1 Introduction

In previous sections, we have explored machine learning and time series analysis approaches to the task of meeting discourse segmentation. We have also given evidences on how the multimodal features are useful for improving the performance of the machine learning approach. However, it has not been studied extensively how to accommodate the multimodal characteristics of meeting discourse in the time series analysis approach. Therfore, in this section, we addresses the challenge of segmenting meeting recordings directly from the inputs of its audio source. In particular, we focus on approaches that can be used to segment a meeting when still in progress, since we expect this to be important to the development of downstream online applications that require immediate content-based access.

In fact, many automatic segmentation systems have been developed to structure meeting recordings into a number of coherent segments [Galley et al., 2003, Al-Hames et al., 2005, Purver et al., 2006b, Dielmann and Renals, 2007d, Hsueh and Moore, 2007b]. Typically, the task is decomposed into a series of binary decisions, each of which determines whether an utterance end contains a segment boundary or not. The dominant approach is to train a classifier with rich features that are obtained from both word transcripts and audio inputs. Although this approach has achieved success, it has some shortcomings. For one, training a well-performing discriminative model requires plentiful labelled data; yet, it is uncertain whether the trained model can be applied to segment meetings in a domain different from the labelled data.

One solution is to apply unsupervised approaches. Many have followed TextTiling approaches, first put forth in [Hearst, 1997], to find optimal segmentation by locating lexical changes over meeting speech [Galley et al., 2003]. These works in unsupervised segmentation commonly assume the availability of manual transcripts or automatic speech recognition (ASR) outputs. Although word errors introduced by high-quality off-line ASR systems do not degrade segmentation performance [Garofolo et al., 2000, Hsueh and Moore, 2007b], we cannot assume ASR outputs of this quality to be readily available in the online scenario.

In the field of spoken language understanding, many research groups have attempted to perform segmentation without transcribing speech into word units first. Some have proposed to locate changes over acoustic units. For example, Malioutov et al. [Malioutov et al., 2007] use an unsupervised vocabulary acquisition technique [Park and Glass, 2006] to derive sub-lexical units (i.e. those corresponding to high frequency words and phrases). So inter-utterance similarity can be used in a clustering approach, originally developed for text segmentation [Utiyama and Isahara, 2001, Choi et al., 2001]. However, it is uncertain whether the vocabulary acquisition algorithm that works in monologues (e.g., lectures) is robust to processing meeting dialogues which are recorded in a natural context. Others have proposed to locate changes in speaker activity, which are characterized by features obtained directly from audio inputs [Renals and Ellis, 2003, Galley et al., 2003, Hsueh and Moore, 2007b].

In this subsection, we perform unsupervised segmentation over audio inputs. Our system

leverages phonetic-level information that can be obtained from audio inputs. Compared to the previous approaches that have leveraged word-level information obtained from either the manual transcripts or the ASR outputs, this approach has a better chance to be operated in near real time. In Section 7.6.2, we describe how the speaker activity-enhanced phonetic representations are processed and how the changes in repetitions of phonemes and that of speaker activities are located. In Section 7.6.5, we compare our audio-based system against the system which segments meeting dialogue as text.

7.6.2 Phonetic Transcription

In this work, our system find segmentation in phonetic units, which have been used as proxies of words in many spoken language understanding applications successfully. We modify LCSeg, a lexical chain-based approach proposed in [Galley et al., 2003], to segment multiparty discourse by locating dramatic changes in the phonetic units over utterances.

To characterize what has been transpired in a meeting, we first have to convert speech signals into a sequence of units. Previous work often does this using an ASR system. As we would like to explore the use of a more language- and speaker-independent way for such conversion, in this work we leverage a phoneme recognition model [Schwarz et al., 2004] that have been successfully applied to cross-language tasks, such as automatic language identification [Matejka et al., 2005], and other spoken language understanding tasks, such as speech recognition and keyword spotting. The phoneme recognizer is trained on ten hours of the SpeechDat-E corpus [42], which consists of recorded spontaneous telephone conversations of 1,000 Hungarian speakers and their pronunciation lexicon [43]. Then the recognizer converts speech signals in the following three steps.

- Feature extraction: First, speech signals are divided into frames of 25 ms long with 10 ms shift. Next, for each frame the system utilizes a Mel-filter bank to obtain its short-term critical band logarithmic spectral density. Finally, temporal pattern (TRAP) feature vectors, i.e., temporal evolution of critical band spectral densities within a single critical band, are generated.

- Phoneme classification: For each critical band a neural network classifier is trained to estimate the posterior probabilities of sub-lexical classes (i.e., phonemes). Then, the outputs of these single band classifiers are merged in another neural network classifier such that a combined estimation of phoneme probabilities can be yielded.

- Representation preparation: A Viterbi decoder is used to produce phoneme strings. We then organize the sequence of phoneme strings into spurts, i.e., speaker turns with pause no longer than 0.5 seconds in-between.

[42] Eastern European Speech Databases for Creation of Voice Driven Teleservices. *http://www.fee.vutbr.cz/SPEECHDAT-E/*.

[43] We use the phonotactic model that is trained on the part of Hungarian speaker data in the corpus, because this model, as shown in [Matejka et al., 2005], outperforms the phonotactic models in other languages in the language identification task.

7.6.3 Modelling Speaker Activity

Previous work has demonstrated the changes in speaker activity as indicative of multiparty discourse segment boundaries [Renals and Ellis, 2003, Galley et al., 2003, Hsueh and Moore, 2007b]. In this work we incorporate the following two types of speaker activity into the recognized phonetic transcripts. The first type ("SPK") includes speaker movements which are characterized by speaker noises (e.g., lip movement, cough), intermittent noises (e.g., door open, note taking), filters (e.g., 'hmm', 'ah') and pauses. The phoneme recognizer we use in this work can provide such information. The second type ("ACT") depicts how talkative each speaker is over the sequence of spurts in the phonetic transcripts. Herein speaker dominance is characterized as the number of phonemes transpired in each spurt; accordingly, we could enhance the phonetic transcription with speaker ID tags, $SPid$, each of which refers to the speaker of a recognized phoneme. Figure 37 (b) is the speaker activity-augmented version of the phoneme representation in Figure 37 (a).

7.6.4 Experiment Setup

This subsection addresses the challenge of whether we can segment a multiparty dialogue recording over its audio sources. In this subsection, we perform experiments to answer the following questions: (1) Whether a lexical chain approach can be extended to find segmentation over utterances represented as phonetic strings; (2) Whether providing speaker activity information in addition to phonetic transcripts can further reduce segmentation errors; (3) Whether segmenting on these different versions of transcripts results in qualitatively different predictions.

In this experiment, we use a set of 48 scenario-driven meeting recordings from the AMI Meeting corpus. These recordings come with manual annotations of hierarchical structure and segment descriptions of these meeting dialogues. We follow previous work to flatten the hierarchical annotations into a two-layer structure of ground truth. We consider all the major discussion segments as the first layer (TOP) and aggregate all the segments in the annotation as the second layer (ALL). The functional segments (FUNC), which serve the purpose of smoothing the procession of a discussion rather than that of contributing to the discussion, are also labelled[44]. On average, each meeting is divided into 14 segments at the second layer (ALL), with around 8 segments at the first layer (TOP); in this two-layer structure, functional segments (FUNC) account for around 42% of the top-level segments and 26% of all segments.

We evaluate the success of segmentation systems using three different metrics: overall segmentation error rate (in Pk and WindowDiff(WD)), time-based accuracy (in precision and recall), and structural similarity between hypothesized and ground-truth segments. First, we use Pk and WD as in previous sections to provide an aggregated account of segmentation errors. Then, we examine which version of transcripts, among the others, yields best predictions of functional segments. We study precision, that is, the proportion of system-predicted segments which correspond correctly to at least one of the functional

[44]Examples of functional segments include opening, closing, chitchat, and discussion about agenda and equipment issues.

segments in ground truth, and recall, that is, the proportion of ground-truth functional segment boundaries which correspond to at least one of the hypothesized segments.

Finally, to understand the performance of segmentation systems in the online scenario, it is also necessary to study systems' capability on gauging the total number of segments in a target dialogue. The structural similarity score is computed as dividing the difference between the number of system-hypothesized segments HYP_K and that of the number of reference segments in ground truth REF_K by REF_K. The closer to zero, the more similar is the system-hypothesized segment structure to ground truth.

7.6.5 Results

	K				unK					
	TOP		ALL		TOP			ALL		
Error Rate/SSim	Pk	WD	Pk	WD	Pk	WD	SSim	Pk	WD	SSim
LC	0.36	0.38	0.36	0.40	0.44	0.55	1.11	0.40	0.49	0.42
PH	0.42	0.43	0.43	0.45	0.40	0.41	0.14	0.41	0.42	-0.23
PH+ACT	0.36	0.39	0.40	0.44	0.35	0.36	-0.38	0.39	0.40	-0.58

Table 69: *Effects of operating unsupervised segmentation on speaker activity-enhanced phonetic transcripts. Pk and WD are error rates of the predictions. SSim is a measure of structural similarity of the predictions in relation to ground-truth segmentation.*

Table 69 demonstrates the effects of different versions of transcripts on segmentation performance. Line 1 shows the performance of the LC model, which locates changes in lexical patterns over word transcripts. Line 2-3 show the performance of the PH model, which locates changes in sublexical patterns over phonetic transcripts[45], and that of the PH+ACT model, which locates changes over speaker activity-augmented phonetic transcripts.

One important parameter to set in this unsupervised segmentation system is the number of segments. In search for segmentation systems that can work in online applications, in this experiment we perform our experiments under two conditions: in the first condition we set the number of segments as the number of reference segments (K)[46], while in the second condition we use a statistically determined threshold to select those most probable segment boundaries (unK)[47]. The first four columns illustrate the K condition. Results show that, when the number of segments is given, the LC model does perform better than the PH model. However, when patterns in speaker dominance (ACT) are jointly considered along with phonetic chains, the new PH+ACT model yields competitive performance to the LC model in the task of recovering top-level segments (TOP) in a dialogue structure.

The right six columns illustrate the unK condition wherein the number of reference segments is unknown. Comparing the results across the two conditions, K and unK, clearly

[45]The phonetic transcripts include both phonemes and information about speaker movements.

[46]We experiment with this condition because we want to compare with many of the previous work that use this setting.

[47]Our system follows previous work to select only potential boundary sites of which the posterior probability predicted by the system are above the mean minus half the standard deviation.

shows a negative effect of the added structural uncertainty on the LC model, increasing the error rate[48] by 22% and 11% on recovering segments at the top level and at all levels respectively. In contrast, the added uncertainty does not significantly affect the performance of the PH model. For the task of recovering the top-level segments, the PH model outperforms the LC model by 10%; Adding the model of speaker dominance (PH+ACT) further reduces the error rate by 14%.

These results suggest that speaker activity-related models have greater potential to be used in online applications to recover granular dialogue segments. Also, as functional segments covers nearly half of the top-level segments (see Section 7.6.4), we expect the accuracy of predicting functional segments to be important to the success of the models for top-level segmentation. Therefore, we perform subsequent experiments to examine the effects of speaker activity-based information on the accuracy of functional segment predictions. Line 1-3 in Table 70 show the results of operating the system on lexical transcripts (LC), phonetic transcripts (PH), and speaker activity-enhanced phonetic transcripts (PH+ACT). Line 4-5 show the results of locating changes in speaker movements and in speaker dominance respectively. Line 6 shows the result of locating changes in both of these two types of speaker activity information. Results suggest that, when the number of segments is given, all the systems that locate changes in speaker dominance patterns (i.e. ACT, PH+ACT, SPK+ACT) yield better precision and recall than LC. In the more realistic condition wherein the number of segments is unknown, these systems still yield higher precision than LC, with the expense of recall.

The columns of SSim in Table 69 and Table 70 demonstrates the level of structural similarity between the predictions of these systems that operate on different versions of transcripts and the ground truth. The close-to-zero figures of the predictions among ACT-related models (such as PH+ACT, ACT, and SPK+ACT) indicates that these systems are better at predicting off-topic functional segments (FUNC).

	K-TOP		K-ALL		unK		
Accuracy/SSim	Prec	Recall	Prec	Recall	Prec	Recall	SSim
LC	0.29	0.75	0.23	0.78	0.16	0.83	6.14
PH	0.27	0.65	0.21	0.70	0.28	0.69	1.91
PH+ACT	0.36	0.86	0.28	0.88	0.40	0.77	0.09
SPK	0.28	0.62	0.20	0.65	0.71	0.61	-1.00
ACT	0.38	0.84	0.25	0.84	0.43	0.77	-.0.05
SPK+ACT	0.37	0.82	0.27	0.88	0.39	0.80	0.39

Table 70: *Effects of speaker-activity models on the accuracy of functional segment prediction. While under the K-TOP and K-ALL condition, the number of manually annotated segments at the TOP and ALL level are given as a constraint for selecting top K predictions from the hypothesis, while the number of segments is unspecified under the unK condition.*

[48] Since the scores of Pk and WD are both aggregated measures of segmentation error rate, we report the change in only one of them, Pk.

7.6.6 Conclusion

Many lexical and non-lexical patterns can be used to recover discourse structure in meeting recordings. Previous work in unsupervised segmentation uses only the lexical patterns obtained on word transcripts. In this work, we explored a novel way to capture lexical patterns, that is, to convert the audio inputs into a sequence of phonetic strings and to derive sub-lexical patterns therein. In addition, we also explored two ways to model non-lexical patterns that pertain to speaker activities: speaker movement (i.e., speaker and intermittent noise, filter, pause) and speaker dominance. We have performed experiments to examine the effectiveness of these different patterns, which can be derived from the audio recordings real time or at least in near real time, on the task of recovering a two-layer structure of meeting dialogues.

Experiments have shown that, when all of these phonetic and speaker activity-related patterns are considered, our audio-based system can yield results comparable to those obtained by operating the system on manual transcripts. Consider a real-life scenario wherein one has missed the first part of a meeting and do not know how many topics have been discussed, our audio-based systems can significantly outperform the word-based system.

Results are encouraging as it shows that speaker activity-augmented phonetic units can serve as proxies of words in unsupervised segmentation of meeting dialogues. Our audio-based system can segment meeting dialogues in absence of manual and high quality ASR transcripts. It is desirable to the development of segmentation components that have to be operated online, or in unfamiliar domains and languages. Also, as the automatically derived dialogue structures can make up for the lack of explicit orthographic cues (e.g., story and paragraph breaks), the audio-based system is expected to be beneficial to developing the online version of many downstream spoken language understanding applications, such as anaphora resolution information retrieval (e.g., as inputs for the TREC Spoken Document Retrieval (SDR) task), summarization, and machine translation.

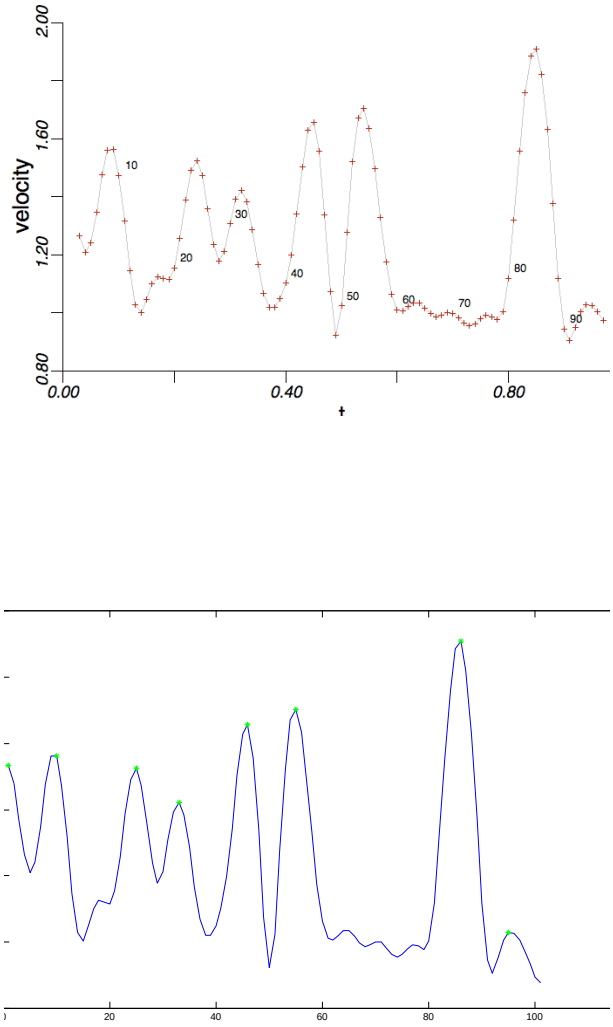


Figure 36: Velocity graph (top) for an AMI meeting, and the corresponding peaks extracted with a peak extraction algorithm (indicated with asterisks).

(a) pau int h m o l k S spk s E m h u E k S m u: l k h E S O k S n E n spk pau int n m spk spk o m O k pau int
(b) pau int h SPb m SPb o SPb l SPb k SPb S SPb spk s SPb E SPb m SPb h SPb u SPb E SPb k SPb S SPb m SPb u: SPb l SPb k SPb h SPb E SPb S SPb O SPb k SPb S SPb n SPb E SPb n SPb spk pau int n SPb m SPb spk spk o SPb m SPb O SPb k SPb pau int

Figure 37: Example of speaker activity-augmented phonetic representation.

8 Addressee classification in meetings using Dynamic Bayesian Networks

We report results on automatic addressee classification in four-participants face-to-face meetings, in particular on the scenario based meeting corpus of AMI. This report builds on that in AMI deliverable D5.2 ([Alexandersson et al., 2006]). There we presented results of our quest for the most relevant features for this task, and about the performance of various types of *static* Bayesian network classifiers.

The task of an addressee classifier is to tell who the addressee of a dialogue act is. Or, "to tell who the speaker is talking to." The classes the classifier can choose from is $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, i.e. one of four individual participants, or **Group**.

The five sorts of features that we considered are:

- utterance features,
- gaze features,
- conversational context features,
- meeting context features and
- participant roles features.

We experimented with a number of types of graphical structures, underlying the various BN classifiers.

- NB - Naive Bayesian Networks.
- TAN - Tree Augmented Networks
- BAN - Bayesian Augmented Networks
- GBN - General Bayesian Networks
- DBN - Dynamic Bayesian Networks

All of these but the last are *static* Bayesian Network classifiers, the decision about the addressee of a dialogue act performed at time *t* does not depend on the decision made in previous situations. The contextual features include information about the addressee of the immediately preceding dialogue act. The *static* classifiers used hand labeled values of the addressee of previous dialogue acts (DAs) when predicting the addressee of the current DA. We used Dynamic Bayesian Network (DBN) classifiers to investigate how well the addressee of the current DA can be predicted using the classified instead of hand labeled value for the addressee of the previous one. This is a first step towards a complete sequential method in which sequences of words are segmented into DA units, the DA types are being predicted as well as their addressee types.

We used two different data sets: the M4 corpus and the AMI corpus. For results on the M4 corpus we refer to [Alexandersson et al., 2006] and [Jovanovic, 2007]. The experiments on the AMI data were conducted

- to explore how well the addressee of a dialogue act can be predicted on the AMI data using various sets of features.
- to explore the impact of meeting context modelled in terms of topical structure on the classifiers' performances.
- to examine the effect of using knowledge about the roles participants perform in meetings on the performances of the addressee classifiers
- to compare performances of the TAN and GBN classifiers in addition to the NB and BAN classifiers for the task of addressee prediction on the AMI data over various feature sets.
- to explore the impact of using the classified instead of the hand-annotated value for the addressee of the immediately preceding dialogue act on the classifiers' performances.

Experiments with the static BN classifiers presented earlier were conducted using various BN classifier learning algorithms implemented in WEKA [Witten and Frank, 2000b] whereas experiments with the DBN classifiers were performed using the Bayes Net Toolbox (BNT) for MATLAB [Murphy, 2001]. Additionally, we conducted experiments with the static BN classifiers using BNT in order to compare performances of dynamic and static BN classifiers. For results with the static classifiers we refer to [Alexandersson et al., 2006] and [Jovanovic, 2007]. Here we report results of experiments with the Dynamic Bayesian Networks.

We developed an NXT based application, Feature Extractor, implemented in Java that is employed for feature extraction and the creation of various data sets. Generated data sets are stored in the WEKA file format - Attribute-Relation File Format (ARFF)[Witten and Frank, 2000b].

8.1 Features for Addressee Classification

In this section we discuss the types of candidate features for addressee classification, that we used in our experiments for finding the most relevant features for the AMI scenario based meeting corpus.

Contextual features The **local context** encompasses contextual information obtained from the relevant dialogue acts from the same or a different channel that most recently precede the current dialogue act. In other words, it comprises n-grams of the preceding dialogue acts. In addition to the immediately preceding dialogue act (1-gram), we also experimented with the extended context that includes two (2-gram) and three (3-gram) preceding dialogue acts. Contextual information obtained from the i-th preceding dialogue act encompasses information about the speaker (**SP-i**), the addressee (**ADD-i**) and the type (**DA-i**) of that dialogue act.

As to the **global context**, we distinguished contextual information obtained from a previous turn from the contextual information obtained from the turn in progress. A turn is

defined as a sequence of successive dialogue acts DA_i for $i = 1, \ldots, N$, produced by the same speaker that satisfy one of the following conditions:

- $start(DA_{i+1}) - end(DA_i) = 0$
- $0 < start(DA_{i+1}) - end(DA_i) \leq T$, where T is a defined threshold, and there are no "turn-relevant" dialogue acts produced by other speakers that occur within the gap between DA_i and DA_{i+1}.

  In our experiments, irrelevant dialogue acts for the definition of turns are those dialogue acts marked as Backchannel, Stall or Fragment.

The i-th preceding turn (T_i) of the dialogue act DA_x is defined as a turn that contains the first relevant dialogue act DA_y preceding DA_X that is not part of previous turns $T_1, \ldots, T_{i-1}$. Turns containing only irrelevant dialogue acts are considered as irrelevant turns.

The second condition in the definition of a turn specifies three types of silence: pause, gap and lapse. If the condition is satisfied, the silence is considered as a pause and DA_{i+1} is included in the current turn. If the difference is greater than T, the silence is classified as a lapse if there are no turn-relevant dialogue acts produced by different speakers occurring within the gap, otherwise it is classified as a gap. In both cases, DA_{i+1} is marked as the first dialog act of the next turn. It is to be noted that the definition of the turn also supports simultaneous speech.

Pauses in ordinary conversations are brief. In meetings, a speaker, however, can make longer pauses while, for example, working on a laptop or while drawing something on the whiteboard. In this situation, according to the distinction made in [Edelsky, 1981] between floor and turns, the speaker is having the floor that consists of several turns. In our experiments, this type of having the floor is considered as one turn. Empirical analysis of the data shows that the maximal duration of silences of this type of holding the floor was around 5 sec; most of them are actually less than 3 sec. Therefore, we experimented with T=5 sec.

Contextual information of a preceding turn encompasses information about the speaker, the addressee and the type of the relevant dialogue act of that turn which most recently preceded the current dialogue act. Contextual information of the current turn comprises information about the addressee and the type of a preceding relevant dialogue act of that turn. We conducted a number of experiments with various *window-sizes* regarding the number of preceding turns as well as regarding the number of preceding dialogue acts within the same turn. Additionally, we explored the performances of addressee classifiers when previous turns of the current speaker were both included and excluded from the contextual feature set. The results reported in this section are achieved using two contextual feature sets:

- **C11**- contains contextual information obtained from the immediately preceding turn (**SP-T-1**, **ADD-T-1**, **DA-T-1**) and contextual information obtained from immediately preceding dialogue act within the same turn (**ADD-1**, **DA-1**)
- **C21**-contains contextual information obtained from two preceding turns (**SP-T-1**, **ADD-T-1**, **DA-T-1**, **SP-T-2**, **ADD-T-2**, **DA-T-2**) and contextual information ob-

tained from immediately preceding dialogue act within the same turn (**ADD-1**, **DA-1**)

In both cases, the preceding turns of the current speaker were taken into account. The reasons for choosing these two feature sets are two-fold: (1) addressee classifiers achieved the highest accuracies when those features are combined with gaze and utterance features and (2) the obtained results are comparable to those achieved using the selected n-gram local context features.

Information about the related dialogue act (**SP-R**, **ADD-R**, **DA-R**) and information about the speaker of the current dialogue act (**SP**) have also been included in the contextual feature set both when experimenting with the local context features and when experimenting with the global context features. For any contextual features included, the NULL value has been introduced to account for instances in which a previous dialogue act segment, as specified in the local or global context, does not exist. The same value is assigned to addressee contextual features that were marked with the Unclassifiable addressee tag.

Gaze features The experiments were conducted using two groups of gaze features. The first group consists of the features defined in the M4 feature set: **SP-looks-P$_x$** and **SP-looks-NT**, where $x \in \{0, 1, 2, 3\}$; **SP-looks-NT** represents that the speaker does not look at any of the participants. The second group of features includes all categories that are labelled as gazed targets in the AMI schema: participants (**SP-looks-P$_x$**), whiteboard (**SP-looks-WB**), presentation slides (**SP-looks-PS**) and table (**SP-looks-T**). As in the first feature group, **SP-looks-NT** is used to denote that the speaker does not look at any of the labelled gazed targets.

We also experimented with two different value sets for both groups of features. First, we defined gaze features as binary features that mark whether or not the speaker looks at the particular gazed target or whether or not he looks away during the time span of the current dialogue act. Then, we experimented with the value set that represents the number of times the speaker looks at a gazed target or looks away in the course of the current dialogue act. The extension of the target set to include other objects in the meeting room had an effect on the distribution of the speaker gaze over the targets. An analysis of the AMI data has shown that instances where the speaker looks three or more times at a particular gazed target occur less frequently in the data. Therefore, we defined the following value set: `zero` for 0, `one` for 1, `more` for 2 or more.

We have found that the addressee classifiers perform slightly better using the limited feature set. Furthermore, when gaze features are used alone in combination with speaker information higher accuracies have been achieved using the value set that denotes qualitative account of the number of times a feature occurs during the time span of the current dialogue act. However, when gaze features are combined with other types of features, the classifiers perform better using the binary gaze features. For that reason, binary features **SP-looks-P$_x$** and **SP-looks-NT** have been employed for the experiments presented in this section.

Utterance features Using the available annotations of dialogue acts and named entities, we experimented with a variety of utterance features that are considered with the

content, duration and the conversational function of the current dialogue act.

- **PP$ feature set** encompasses subjective and objective personal pronouns, possessive pronouns and possessive adjectives. It consists of the following binary features: **1.sing**, **1.pl**, **2.sing/pl** and **3.pl/sing**. For example, **1.pl** denotes whether or not the utterance contains "we", "us", "our" or "ours". In the M4 feature set, **PP$** is partially defined with four-values **PP** and **PPA** features that contain information about `we` and `you` person categories.
- **IP**- whether or not the utterance contains indefinite pronouns such as "somebody", "someone", "anybody", "anyone", "everybody" or "everyone"?
- **ParticipantRef feature set** includes the features that mark reference to meeting participants:
  - **Name-P_x**- whether or not the utterance contains the name of participant P_x where $x \in \{0, 1, 2, 3\}$. In order to distinguish the usage of the name as an addressed term from other usages, we also included the **BeginOrEnd-P_x** feature in the set. It denotes whether or not the name of participant P_x occurs at the beginning or at the end of the utterance.
  - **Role-P_x** - whether or not the utterance contains the role of the participant P_x.
  - **NameOrRole-P_x** - whether or not the utterance contains the name or the role of participant P_x. **NameOrRole-P_x** is mutually exclusive with **Name-P_x** as well as with **Role-P_x**.
- **Short**- whether or not the utterance duration is less than or equal to 1 sec.
- **NumWords**- qualitative description of the number of words in the utterance: `one` for 1, `few` for 2, 3, 4 words, `many` for 5 or more words. As these **NumWords** and **Short** features provide almost redundant information, we decided to select one of those features in the final model.
- **Reflexivity** - whether or not the utterance is reflexive.
- **DA-Type** - the conversational function of the current dialogue act. In defining a value set for the **DA-Type** feature, we experimented with different groupings of the dialogue act categories. The results presented in this chapter were obtained using the following value set: `inform`, `assess`, `social`, `elicit`, `offer`, `suggest`, `comment-about-understanding`

Out of all listed utterance features, **PP$**, **DA-Type** and **NumWords** were shown to be the most informative when combined with selected contextual and utterance features.

Meeting context Meeting context is modelled in terms of the **Topic** feature. Although the AMI topic segmentation schema allows topics to be nested up to several levels, we experimented only with top-level topics, which reflect largely the meeting structures based on the meeting scenario. Functional topics, as defined in the AMI schema, can also be

labeled as top-level topics. As they reflect the actual process and flow of meetings (e.g opening, closing), they were also taken into account in modeling meeting context. Although the schema provides a pre-defined set of topic descriptions for top-level topics, annotators were allowed to introduce their own descriptions when necessary. However, we considered only pre-defined topic descriptions; all other descriptions were grouped into the `other` category.

The value set for the Topic feature contains the following descriptions: `agenda/equipment`, `opening`, `closing`, `project specification`, `new requirements`, P_0`-present`, P_1`-present`, P_2`-present`, P_3`-present`, `discussion`, `prototype presentation`, `prototype evaluation`, `project evaluation`, `costing`, `drawing` and `other`. Regarding the topics that refer to presentations, the AMI annotation schema contains the descriptions that refer to participant roles such as marketing expert presentation or industrial designer presentation. However, in the data processing step we mapped these values into corresponding values P_0-present, P_1-present, P_2-present, P_3-present incorporating in that way the background knowledge of the participant roles into the classification models.

Participant roles In addition to incorporating the background knowledge about the roles participants play in the AMI meetings in an implicit way by mapping some of the features or feature values defined in terms of participant roles into corresponding features or feature values specified in terms of the participants that play these roles (e.g. **Name-P_x** or **Topic**), we also modeled this knowledge in an explicit way by defining new features that bear information about participant roles.

The experiments were conducted using solely information about the speaker role modeled in two different ways. First, we introduced the **Dominant** feature which denotes whether or not the speaker is the participant with the dominant role in the meeting, that is, project manager. Second, we experimented with the **SP-Role** feature which marks one of four AMI scenario roles the speaker fulfils in a meeting: `PM`, `ID`, `UI` or `ME`. The motivation for using the Dominant feature is that the participant with the dominant role in a meeting is expected to address the whole audience on average more than is case with the other meeting participants. However, the leading role in the meeting can also be determined by the current meeting activity. For example, a presenter during the presentation can take over the leading role for that part of the meeting or a participant with a particular role may become the dominant speaker when a topic related to his work and knowledge is being discussed. For some types of activities defined in the AMI meetings, such as presentations, this type of information has already been encoded in the data processing step. Introducing the SP-Role feature, we aimed to investigate whether the information about the dominant role for other types of activities and topics can be extracted from the SP-Role feature. However, we have found that the knowledge about the particular role that the speaker performs in the meeting does not provide any additional information in comparison to the information provided by the Dominant feature. This can also be caused by the fact that the meeting context is modeled in terms of the Topic feature which bears information about meeting structure specified more in terms of meeting activities (e.g. discussion or opening) than in terms of the particular topic being discussed (e.g. look and usability, components and materials, trend watching).

Data	Description	Total	P_0	P_1	P_2	P_3	Group
A set	- IS1006d	5380	13.61%	11.08%	9.28%	9.80%	56.25%
B set	+ IS1006d	6077	14.30%	10.70%	9.13%	11.14%	54.73%

Table 71: AMI data sets

8.2 Data sets, evaluation metrics and methods

Data sets Only a small part of the AMI scenario-driven collection has been annotated with addressee information. For the experiments presented in this chapter, we selected 14 meetings that were annotated with addressees and focus of attention[49]: ES2008a, TS3005a, IS1000a, IS1001a, IS1001b, IS1001c, IS1003b, IS1003d, IS1006b, IS1006d, IS1008a, IS1008b, IS1008c, IS1008d. Most of the selected meetings were recorded in the IDIAP meeting room. As the IS1006d meeting was not annotated with all types of information that we used in some of our experiments, we created two data sets to experiment with: the **A set** that excludes the IS1006d meeting and the **B set** that contains all 14 meetings. For each data set, the distribution of class values is given in Table 71. The total numbers of instances presented in the column Total denote the total numbers of relevant dialogue act segments that are marked with a class label.

Evaluation methods For evaluating performances of the static BN classifiers, we performed stratified 10-fold cross validation on the A set. For the experiments with DBNs, we made use of the larger B data set because the features selected for those experiments were annotated for all meetings included in the B set. Moreover, we divided the data set into 5 folds, 4 of which contained 3 meetings and one contained 2 meetings, ensuring that folds contain approximately similar number of instances. Due to uneven distribution of addressee values across the meetings, it was not feasible to specify the data set partition into n folds containing the meetings that completely satisfy the stratification criterion. In our partition, the average difference between the distribution of the addressee values in corresponding training and test folds is 2.5% with a maximal difference of about 10% for the group addressee value in one of the folds. To compare performances of the DBN and static BN classifiers, the static classifiers were also evaluated using 5-fold cross validation on the defined folds.

Structures of the static BN classifiers were learned from the data whereas structures of the DBN classifier were designed based on the learned static structures. For the comparison of the DBN and static BN classifiers, we experimented not only with the learned but also with the specified structure of the static BN classifiers.

For learning parameters of the DBN classifier, we applied the EM algorithm with uniform Dirichlet priors on network parameters. The MAP algorithm with uniform Dirichlet priors was employed for learning parameters of the static BN classifiers with fixed structures.

[49]There are several meetings in the AMI corpus annotated with addressee information for which focus of attention annotations are not available

Metrics In addition to the overall accuracy, the detailed accuracies per class value have been estimated in terms of precision, recall and F-measure. Relevant dialogue acts marked with the Unclassifiable addressee tag were employed for deriving contextual information used for predicting the addressee of the dialogue act at hand. In the static case, those instances were removed from the data set after the contextual information obtained from them had been encoded in the feature set for the dialogue act that follows. In the dynamic case, however, the instances of relevant dialogue acts marked with Unclassifiable addressee labels were not removed from the data set.

In the DBN model, addressee values for the Unclassifiable dialogue acts were treated as missing. However, instances of dialogue acts in the test set with missing addressee values were not considered in the estimation of the classifier's performance. In other words, the accuracies of the DBN classifier were calculated as the ratio between correctly classified test instances and the total number of the instances in the test set that were annotated with a class value.

8.3 Addressee classification using DBN classifiers

In the experiments with the static BN classifiers, we investigated how well the addressee can be predicted using, among other features, information regarding the addressees of preceding dialogue acts. The following notions of a 'preceding dialogue act' were taken into account:

- the immediately preceding dialogue acts from the same or a different channel
- the last dialogue act of a previous turn that precede the current dialogue act
- the preceding dialogue act of the same turn
- the related dialogue act

In DBNs, the contextual information about preceding DAs is propagated through history: the information provided by the dialogue act segment at time slice T-2 influences the addressee of the dialogue act at slice T implicitly through the information provided by the dialogue act at slice T-1. Explicitly modeling relations between dialogue act segments at T-2 and T may increase the complexity of the network and thereby require more data for learning the parameters. Experiments with static BNs show that there is no significant improvement between 1-gram and 2-gram models. The question is thus if use of the history in DBNs will improve the performance of the static BNs.

Besides excluding information about the addressee of the related dialogue act (**ADD-R**) from the contextual feature set as it represents the class variable, we also excluded the information about the type of the related dialogue act (**DA-R**), experimenting in this way only with the information about the speaker of the related dialogue act (**SP-R**). Additionally, we conducted experiments without the **SP-R** feature aiming to develop a model for addressee prediction of which the outcome can be used for the detection of the related utterance. Utterance and gaze features have also been employed for addressee classification using DBN.

The structures of the DBN classifier were designed based on the learned static structures, which provide visual insight into relationships between features and the class variable as well as between features themselves. We experimented with a variety of static structures, and their modifications, that were learned using the feature set selected for the experiments presented in this section by performing 10-fold cross validation on the B-set.

The highest accuracies are obtained using the designed structure presented in Figure 38.

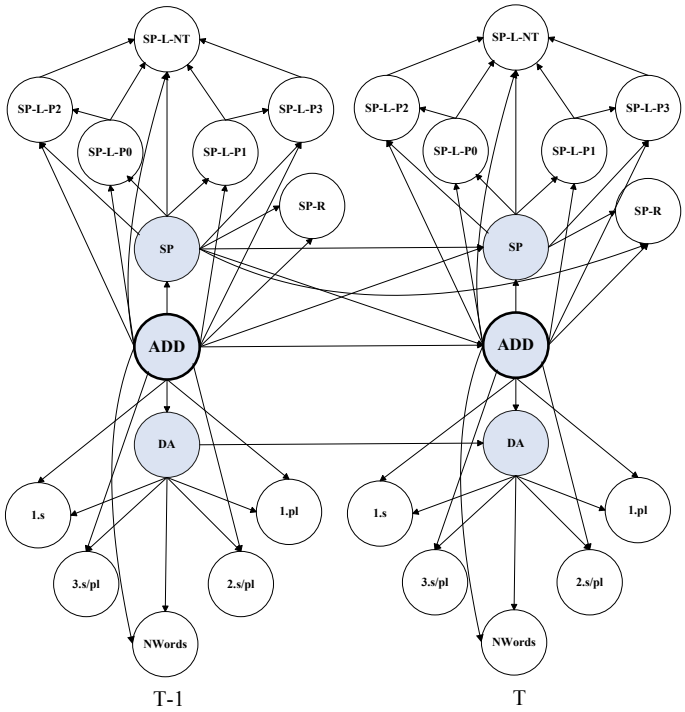


Figure 38: The structure of the DBN classifier

As shown in Figure 38, the static classifiers have learned dependencies between the speaker looking at participants seated at the same side of the table (e.g. **SP-looks-P$_0$** and **SP-looks-P$_2$**). In this way, the background knowledge regarding the seating arrangement is incorporated in the network structure. The structure also captures dependencies among contextual features as well as among utterance features, in particular **DA-1** and **DA** features, between two adjacent slices. We did not model dependencies between corresponding gaze features of two adjacent dialogue act segments for two reasons, mainly.

First, as the gaze information is modeled in terms of the speaker gaze features, a speaker change has the effect that the dependencies between corresponding gaze features (**SP-looks-P$_x$(t-1)** and **S-looks-P$_x$(t)**) does not model the probability that the same target is gazed in two adjacent slices as is the case if the speaker change did not occur. Second, as the sequence of the *relevant* dialogue acts is modeled in DBN, there can be a time gap in between two adjacent relevant dialogue acts which may have an influence on the speaker gaze behavior. It is to be noted, that **SP-R** is modeled as a static variable whereas the **SP-1** feature is modeled as a dynamic variable in the network (**SP(t-1)**).

We considered contextual information provided by relevant dialogue acts that are marked with the Unclassifiable addressee label for the prediction of the addressee of the dialogue

act at hand. In the dynamic case, as opposed to the static case, we did not exclude dialogue acts marked with the Unclassifiable addressee label from the data set as it would affect some other dialogue act to precede the current one and thus contextual information to be derived from that dialogue act. Therefore, a larger number of instances is employed for training the DBN classifier with some of them having missing addressee values. However, for evaluating the performances of the DBN classifier, we considered only those instances in a test set that were annotated with a class value. About 4% of all instances in the B set were labelled as Unclassifiable. Due to incompleteness of the data set, the EM algorithm with uniform Dirichlet prior has been employed for computing the MAP estimates of the network parameters. For more details we refer to [Jovanovic, 2007] and [Heckerman, 1995].

The performances of the DBN classifier were evaluated by performing 5-fold cross validation on the B set). To gain insight in how well the DBN classifier performs in comparison to the static BN classifiers, we evaluated the performances of the static BN classifiers using the same evaluation method. The static BN classifiers were developed in two different ways:

1. **static condition** - using the K2 algorithm for structure learning and the MAP estimator with $\alpha_{ijk} = 0.5$ for parameters learning (see [Heckerman, 1995] for structure learning).

2. **dynamic condition** - using the MAP algorithm with uniform Dirichlet priors for learning the parameters of the network with fixed structure. We used the structure presented in Figure 38 transformed into a static network.

The static classifier with a fixed network structure, similar to the one presented in Figure 38, was designed in a way that the addressee node is treated as a root node in the network. Furthermore, the addressee node was defined as a parent of all feature nodes. Since feature nodes form an arbitrary graph, the addressee classifier with such a structure is defined as BAN classifier. Therefore, we only report the results for the BAN classifier for the static condition. Table 72 summarizes classification accuracies of the DBN classifier as well as of the static BN classifiers for both static and dynamic conditions.

Feature set	**DBN**	**BAN(1)**	**BAN(2)**
All Features	71.83	75.63	75.58
All Features\{SP-R}	68.25	73.21	73.45

Table 72: Accuracies of the DBN classifier and the static BN classifiers under (1) static and (2) dynamic conditions

The results indicate that for both feature sets, the static BN classifiers significantly outperform the DBN classifier. Furthermore, both static and dynamic addressee classifiers significantly outperform the baseline classifier which always predicts the majority class (54.73%). Although the classification results for the dynamic and static BN classifiers are not quite comparable due to different treatment of the Unclassifiable addressee value, from the presented results we can conclude that the usage of the classified instead of the

hand annotated value has a *negative* impact on the classifier performances. Furthermore, both static and dynamic BN classifiers show a decrease in performance when information about the speaker of the related dialogue act is excluded from the contextual feature set (DBN: about 3.5%, BAN: less than 2.5%). These results are comparable to the results obtained using the static BN classifiers on the A set when information about related dialogue act was excluded from the contextual feature set. Note that the static BN classifier with the designed structure shows similar performance as the static BN classifier which structure was learned on each training fold.

Further analysis of the misclassified data instances has shown that the DBN classifier failed in a considerable number of cases to detect a change in addressing within a turn, especially when the speaker changes from talking to an individual to talking to a group and vice versa. This can be due to the fact that this type of change in many cases is not marked by the change in gaze behavior but by specific features of the utterance content that are not captured with the selected feature set.

8.4 Summary of findings

From the experiments with various graphical models and static as well as dynamic BNs and various types of features for classifying the addressee of a dialogue act in face-to-face meetings, we can conclude, the following in relation to the types of features that are used for the task.

- Extending the conversational local context from the 1-gram model to the 2-gram model slightly improves the performance of the static BN classifiers on the AMI data. However, further extension of the conversational context decreases the performances of all static BN network classifiers except of the NB classifier which show a little gain from the extension of the conversational context.
- Modeling conversational context in the global way does not significantly change classifier performances in comparison to performances obtained using the local context. However, the augmented NB classifiers show the largest gain from modeling context in the global way.
- Addressee classifiers, both static and dynamic, show a small decrease in the performances when contextual information obtained from the related dialogue act is excluded from the contextual feature set. This indicates that remaining contextual, utterance and gaze features cover the most useful information for addressee prediction provided by the related dialogue act.
- Addressee classifiers show a little gain from information about meeting context modeled in terms of the Topic feature.
- Information about the speaker role has no significant impact on addressee prediction when combined with utterance, gaze and contextual features as well as with the Topic feature.

If we compare different graphical models we can conclude as follows.

- For all classifiers, the accuracies are significantly higher compared to the baseline.
- Augmented NB classifiers show the best performances over all feature sets.
- The NB classifier performs significantly worse than the augmented NB and GBN classifiers for the 1-gram model and significantly worse than the augmented NB classifiers for the 2-gram model. In all other cases, there is no significant difference in the accuracies among addressee classifiers when local context features are employed.
- The augmented NB classifiers significantly outperform the NB and GBN classifiers for both the C11 and C22 models. For the C11 model, the GBN classifier significantly outperforms the NB classifier. For the C21 model, the NB and GBN classifiers do not show significant difference in the performances.
- Static BN classifiers which use as a feature the hand annotated value for the addressee of the preceding dialogue act significantly outperform the DBN classifier which employs the classified value for the addressee of the preceding dialogue act.
- Addressee classifiers on the AMI data show the same type of misclassifications as human annotators: individual and group are most confused.

8.5 Towards further automation of addressee detection

The experiments reported here were conducted using a set of manually annotated features: for utterance, gaze, conversational context and meeting context. Some of the features can be easily extracted (e.g **NumWords**), others require complex computational modelling for their automatic detection (e.g. **DA-Type**, **SP-R**, **SP-Looks-P$_x$**, etc.). Moreover, we assumed that the dialogue act segment boundaries are given. Fully automatic addressee identification requires that dialogue act segment boundaries as well as utterance features and some of the contextual features are based on the output of an automatic speech recognizer (ASR) while gaze features are estimated from visual information. Most of these features can be better detected by combining multimodal - audio and video - cues.

Several issues that arise at this point are: (1) whether there are technologies available for automatic recognition of the features for addressee identification in the context of meetings (2) if so, what is the quality of the extracted features and (3) to what extent the quality of automatic feature extraction decreases the performances of the addressee classifiers achieved using the hand-annotated features.

We shortly review the automatic classification of some phenomena in conversations that are related to addressing.

Adjacency Pairs and Addressing Regarding contextual features, the main issue is how to automatically identify contextual information regarding the related dialogue act. To our knowledge, there is not much work done on automatic adjacency pairs identification in multi-party dialogues. Recently, [Galley et al., 2004] reported the results on AP detection in multi-party meeting dialogues using the maximum entropy ranking model. The results indicate that given the b-part of an adjacency pair, the speaker of the a-part

can be detected with accuracy of 90.12% using a set structural, durational and lexical features. This accuracy is achieved using "backward-looking" and "forward-looking" features. However, when excluding forward looking features, which concern the closest utterance of the potential speaker of the a-part that follows the b-part, the model performs worse (86.99%). It is to be noted, that the classification task for the experiments presented in [Galley et al., 2004] concerns the detection of the speaker of the a-part without identifying to which dialogue act of that speaker the b-part is related: the basic unit of analysis is a spurt which represents a period of speech that has no pauses greater than 0.5 sec. Therefore, for the experiments with the DBN presented in Section 8.3 we excluded the DA-R feature from the contextual feature set in addition to the ADD-R feature that was primarily excluded for the purpose of sequential modelling. In the hand annotated AMI corpus we see that about 60% of the speaker addressee pairs of two related dialogue acts have the pattern *ABBA*, the speaker of the a-part (the *target* of the relation) is the addressee of the b-part (the *source* of the relation).

Automatic DA segmentation and Addressing Addressing is an aspect of a dialog act and it seems reasonable to say that dialogue acts that are directed towards a group or that are just broadcasted, or that are more self talk and unaddressed, are of a different type than dialogue acts that are addressed to a single addressee. A clarification question (Comment About Understanding) is typical directed towards the speaker, as is a feedback signal that sounds "I can't hear you". Where others are typical addressed to the group: "Who was the last before me?" This suggest to see the identification of the addressing mode as a part of the DA recognition task, that does give feedback on the proces of DA classification. Note that addressing change during a "speaker turn" is an indication of a DA segment boundary. Thus, also segmentation can favor from addressing features.

Gaze, focus of attention and addressing From our analysis of the hand annotations of Focus of Attention and addressing in the AMI corpus it came out that when a speaker S directs his speech at a single addressee A then S her focus of attention is 3 times more at participant A than overall when S speaks. It also shows that when a speaker looks during his talking more at one single other participant than at others in 60% of the cases she addresses that person, and not the group. Gaze of speakers is an important indication for coming to know who she is addressing. But addressing should not be confused with focus of attention. Moreover, the situation is different in small face-to-face conversations where people are sitting at fixed positions at a table.

Since it is very difficult to record eye gazing of meeting participants, information about visual focus of attention can be automatically induced from head orientation. Experimental results presented in [Stiefelhagen and Zhu, 2002] indicate that estimation of focus of attention based solely on head orientation achieve the accuracy of 88.7% in four-participants meetings. In their previous work, [Stiefelhagen et al., 2002] presented a system for estimating focus of attention based on multimodal cues: gaze directions and sound resources. First, participants' gaze direction was estimated from their head orientation. The gaze detected estimations are then used to predict focus of attention given a head pose. The scored accuracy using this approach is 74%. Adding audio information to video information increased the accuracy to 76%.

Preliminary results on recognition of focus of attention based on the head pose orientation on the AMI data are reported in [Al-Hames et al., 2006a]. In contrast to the work reported in [Stiefelhagen and Zhu, 2002] that was restricted to recognition of meeting participants as focus of attention targets, the recognition task presented in [Al-Hames et al., 2006a] was considered with the recognition of the extended focus of attention label set that includes also table, slide screen and unfocused label. The obtained classification rate was 68% and 47%. As the authors claim, the lower recognition results are mainly due to the usage of more complex setting and the extended label set. The current research on the focus of attention recognition on the AMI meetings is concerned with adaptation of the approach to include information from other modalities in order to improve the classification results.

Floor and Addressing Part of addressing is to check the communication line. At least in face to face meetings addressing assumes shared knowledge about contact between two parties. The speaker want to see whether his message is received by the addressed party. Thus the speaker will call for attention if she is not convinced that a communication line is open, and wait for feedback that assures the open line. Explicit addressing behavior is required more if a topic and or floor change occurs than in situations where an ongoing interaction is being pursued. Thus topic and floor changes are related to those aspects of behavior that we use to identify addressees. Certainly in groups where information is distributed and where the task is to share this information with others, required to perform the group task, we see a direct relation between the issue being addressed and the participants that feel being addressed by the speaker when she addresses the issue. A question about the market is more likely asked at the marketing expert than at the interface designer.

An integrated approach towards identification of various aspects of a conversational situation may be advantageous compared to a sequential approach. Dynamic Bayesian Networks seem like a general enough model for this task, since they allow modularity and the classification of sequential data.

8.6 Conclusions

We presented results on addressee classification in four-participants face-to-face meetings using several types of static BN classifiers as well as using the DBN classifier. The classifiers were evaluated on the AMI meeting corpora using features obtained from multiple resources: speech, gaze, conversational context, meeting context and background knowledge about participant roles. As the features used in the classification models are based on hand annotated information, the experiments presented in this chapter concern establishing the upper bounds for the task of addressee prediction in face-to-face meetings.

We have found that contextual information aids classifiers' performances over utterance and gaze information. Furthermore, utterance features were shown to be the most unreliable cues for addressee prediction. The exploration of the impact of listeners' gaze information on the performances of the addressee classifiers', leads us to the conclusion that listeners' gaze direction provides useful information only in the situation where gaze features are used alone. The addressee classifiers reach the highest accuracies when combin-

ing utterance and contextual features with the speaker's gaze directional cues. Combining information about meeting context modeled in terms of the current meeting activity with the utterance, contextual and speaker gaze features improves slightly but not significantly the classifiers' performances.

In contrast to [Vertegaal, 1998] and [Otsuka et al., 2005] findings, where it is shown that gaze can be a good predictor for addressee in four-participants face-to-face *conversations* our results indicate that in four-participants face-to-face *meetings*, gaze is less effective as an addressee indicator. This can be due to several reasons. First they used different seating arrangements which is implicated in the organization of gaze. Second, our meeting environments contain attention distracters. Finally, during a meeting, in contrast to an ordinary conversation, participants perform various meeting activities which may have an effect on gaze as an aspect of addressing behavior.

Since conversational context provides the most useful information for addressee prediction, we explored whether the performances of addressee classifiers on the AMI data can be improved by better exploitation of the contextual information. The conversational context has been modeled in two ways: local and global. The local context concerns n-grams of the preceding dialogue acts from the same or different channel. The global context, on the other hand, distinguishes contextual information obtained from the preceding turns from the contextual information obtained from the turn in progress. From the results concerning the local context, an important conclusion to be drawn was that the extension of the local context to include not only the immediately preceding dialogue act but also the dialogue act that precede that one, slightly improves performances of all static BN classifiers although the NB classifier gains the most. However, further extension of the conversational context decreases the performances of all addressee classifiers with the exception of the NB classifier which shows a small increase in the accuracies. From the experiments concerning the global context, we found that modeling context in the global way does not significantly change the performances of the addressee classifiers in comparison to the performances obtained using the local context although the augmented NB classifiers gain the most from the global representation of the context.

As addressee information can be used as a useful cue for the detection of the related utterance, we estimated the performances of addressee classifiers when information about related dialogue act was excluded from the contextual features set. It was found that the static BN classifiers evaluated on both meeting corpora and the DBN classifier evaluated on the AMI data show a similar decrease in the accuracies (about 3%) when information about related dialogue act is not taken into account. Since this information is a strong indicator for addressee prediction, this decrease in the performances indicates that remaining contextual, utterance and gaze features cover the most useful information provided by the related dialogue act.

Further research on addressing Addressee classifiers had problems in distinguishing between individual and group addressing. (Note that human annotators disagree most between these two types of addressing.) We still have to see in what particular types of situations the classifiers are reliable and in what situation the predictions are less reliable. In those situations in which someone clearly addresses some individual and uses "you" and or "your" as a deictic reference to that individual, does the classifier come up with the

right outcome? Does the one who is predicted as individual addressee take the turn?

Several people pointed at the relation between addressing and leadership or dominance. See eg. Gibsons interesting analyses of how individuals differentiate in terms of their involvement in "participation shifts", the changing in roles between speaker, addressee and unaddressed participant [Gibson, 2003]. But see also Bales work [Bales, 1950]. Dominant people, those who speak the most overall, address an unusual large proportion of their remarks to the group, and they are more addressed individually than less dominant participants in meetings. In AMI in most meetings (of those that are hand annotated with addressees) the project manager is the most dominant and she is also the one who is mostly addressed and the one who is mostly talking to the group.

It is to be expected that addressing in remote settings is often more explicit than in face-to-face settings. We expect more use of vocatives and more explicit call for attention when a speaker addresses someone at an other site.

In [Traum, 2004]) a rule-based method is presented for real-time addressee classification. It uses the following rules:

1 If utterance specifies addressee (e.g., a vocative or utterance of just a name when not expecting a short answer or clarification of type person) then Addressee equals the specified addressee

2 else if speaker of current utterance is the same as the speaker of the immediately previous utterance then Addressee = previous addressee

3 else if previous speaker is different from current speaker then Addressee equals the previous speaker

4 else if unique other conversational participant then Addressee equals that participant

5 else Addressee unknown.

The algorithm is used in an interactive virtual environment for real-time processing. We will investigate the performance of variants of this method on the AMI corpus.

9 Argumentation

We have tested the strength of the relationship between the way that people behave in a discussion and their level of influence using the data source that were collected from the AMI corpus for the research on argumentation [Verbree et al., 2006a], dialogue-act [Verbree et al., 2006b] and influence [Rienks et al., 2006]. Statistical dependencies and (cor)relations between the tags were mined for possible relationships.

9.1 The Data

The Argumentation Annotations The Twente Argument Schema (TAS) is an annotation schema designed to create argument diagrams from meeting transcripts. It identifies the argumentative functions of the different contributions made by debating participants and labels the relations that exist between these contributions. Only those parts in the meeting where participants involved in a discussion are put into an annotation tree. An overview of the complete annotated set is shown in Table 73. Units in the schema are a list of utterances by a speaker that together express an idea which can take the form of statement (expressing a belief, idea) or that poses an issue (something that needs to be resolved, like a question, or a suggestion). These units can be related in various semantic ways: one utterance can be a positive answer to an issue. One utterance may be a more specific description of the previous ones, or in the case of an argument one utterance may state exceptions or conditions that need to be in place for an argument to hold (Subject to). For more information on the specific meaning of the labels consider [Verbree et al., 2006a].

The Dialogue-act Annotations The AMI dialogue act scheme consists of 15 dialogue acts. For an overview of this data consider Table 74.

The Influence Annotations In 40 meetings, the participants were asked to rank all participants of their meeting, including themselves, from most to least influential by assigning them unique nominal values ranging from one (most influential) to four (least influential). These were rankings that held for the complete meeting. Participants were not allowed to rank people equivalently. The collected permutations of the numbers one, two, three and four, were quantized into three classes as described in [Rienks et al., 2006]. The resulting data set had a total of 160 labels (40 meetings times four participants) resulting in 34 observations for 'Low', 91 for 'Normal', and 35 for 'High'.

The Dataset Used The combined influence - dialogue act annotations were available for 30 AMI meetings and the combined influence - argumentation information was available for 29 discussions distributed over 18 meetings. All in all 865 of the total of 6920 (12.5%) TAS unit labels were covered with influence information. All in all it resulted in the data set that is shown in Table 75.

A first exploration reveals that the distribution of argument labels as a function of the influence values does not turn out to be significant ($\chi^2(4, N = 864) = 4.73, P < 0.31$), nor

Node labels	Amount
Statement (STA)	4077
Weak statement (WST)	194
Open issue (OIS)	232
A/B issue (AIS)	69
Yes/No issue (YIS)	443
Other (OTH)	1905
Total	6920
Relation labels	Amount
Positive	2319
Negative	471
Uncertain	259
Request	223
Specialization	131
Elaboration	689
Option	601
Option exclusion	14
Subject-to	190
Total	4897

Table 73: Distribution of TAS labels.

do ANOVA tests on the individual labels show any significant results. As a consequence, one might conclude that both phenomena seem to be independent.

We used several techniques to explore further potential correlations.

9.2 Rule Induction

We used an unsupervised mining method known as association rule mining to explore the data for patterns. The 'Tertius' algorithm [Flach and Lachiche, 2002] we used for rule mining provides two measures for the strength of the rule: the confirmation value[50] and the frequency of counter-instances (the number of counter-instances divided by the total number of data items). A rule is said to be better than another if it has a higher confirmation value.

For this experiment the influence class labels and the fractions of the various argumentation labels per meeting were used. To allow the data to be used for rule induction, the label fractions were quantized in three nominal categories 'High', 'Normal' and 'Low' using WEKA's simple binning algorithm [Witten and Frank, 2000a]. This was done to get hold of the argumentation label distributions per influence category. Results seem to suggest that a high 'Issue' frequency in combination with a low 'Other' frequency seems to be more representative for highly influential people. People of low influence, on the

[50]The confirmation value trades off the decrease in counter-instances from expected to observed and the ratio of expected but non observed counter-instances (see [Flach and Lachiche, 2002] for more detail).

Label	Amount	**Label**	Amount
Fragment	14348	Assessment	19020
Backchannel	11251	Comment about understanding	1931
Stall	6933	Elicit assesment	1942
Inform	28891	Elicit comment about understanding	169
Elicit Inform	3703	be positive	1936
Suggest	8114	be negative	77
Offer	1288	Other	1993
Elicit Offer or Suggestion	602		
Total	102198		

Table 74: Distribution of Dialogue acts in the AMI corpus.

	low	Normal	High	Total
Issues	12	40	27	79
Statements	78	254	152	484
Other	65	153	84	302
Total	155	447	263	865

Table 75: Distribution of label combinations for combined argumentation (merged) and influence data.

other hand, score high on the 'Other' units and low on the 'Issues'. As could be expected from the confirmation values, post-hoc statistical analysis revealed that these hypotheses do not prove to be statistically significant.

A second experiment was performed with a data set containing the influence values added to all TAS unit labels and its associated features (including the relation that attaches the node to the tree). All of the features were again binned into the three (high, normal and low) bins. From the outcome one can tentatively conclude that relatively high influential people respond to people who provide responses to Yes/No issues. People with a relatively low influence level seem to use fewer question marks, use the word 'or' less frequently and provide relatively short responses. This seems to align with the finding reported above that influential participants seem to raise more 'issues' and generally provide less units that can be labelled as 'other'.

Besides rule induction, one can use other methods to look for correlations.

9.3 A Closer Look

This section reports on experiments that were conducted to find out other dependencies between the TAS scheme and the participants influence rankings than those that could emerge via rule induction.

9.3.1 TAS units and influence

We started by conducting three different kinds of experiments to see whether, and if so which, aspects in relation to the TAS unit labels could be (cor)related to the various influence levels. Examined for possible relationship with the influence rankings were: the total number of units, the average unit duration, and the unit type distributions.

Examining the number of TAS units When considering the number of TAS units uttered per person per meeting, an average of 7.27 was found with a standard deviation of 3.56. We also counted the number of turns of each participant. No significant differences were found with respect to the number of turns for each type of influence level. When zooming in on the contribution of turns along the discussion (split up in five bins of equal time intervals) we obtain Figure 39.

Apart from the fact that no difference exists in the total number of TAS units uttered per influence level, no significant difference for the various influence levels when considering the number of TAS units uttered per bin were found. A significant positive correlation, however, was found between the fraction of turns and the progress of the discussion for all influence levels combined (Pearson's correlation coefficient r=0.22, with a significant regression model $F(1) = 30.34, P < 0.001$) as well as for the separate influence levels (r between 0.24 and 0.19, $P < 0.01$ for 'Medium' and $P < 0.03$ for 'Low' and 'High').

This finding shows that towards the end of the discussion people tend to talk in shorter turns. A logical explanation for this might be that people reach agreement towards the end, and that contributions in terms of 'yeah' and 'sure' occur more frequently. Another, but perhaps less likely explanation could be that people start to run out of time and therefore try to limit the length of their contributions.

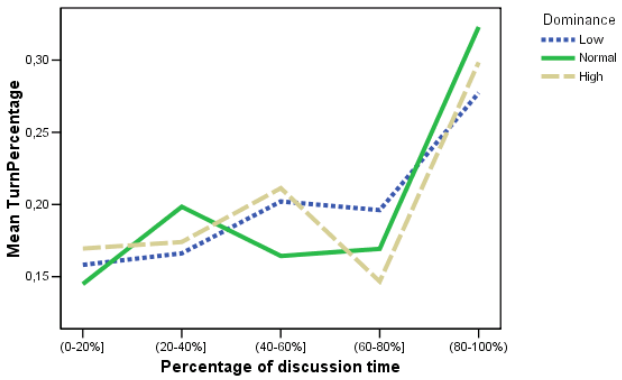


Figure 39: The fraction contributions divided over five time intervals per influence type.

9.3.2 Dialogue acts and influence

We examined more closely whether and how, certain categories of dialogue-acts can be related to the various influence rankings over the course of a meeting. Dialogue act annotations were available for 30 of the 40 meetings with influence rankings.

As a first attempt the fractions for the occurrence of all dialogue-acts was computed for all participants. These results were subsequently merged for each of the influence levels. The resulting average fractions are shown in Figure 40.

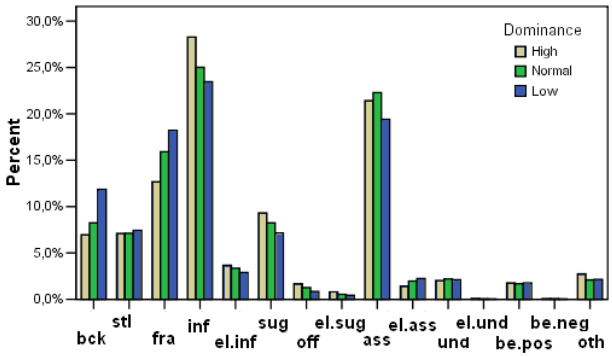


Figure 40: The fraction of dialogue acts per influence level.

Figure 40 seems to suggest some interesting differences between the various dialogue act distributions. Statistical analysis by means of ANOVA showed that on the P<0.05 level significant differences exist for the labels 'fragment' ($F(2)= 7.87$, $P < 0.001$), 'back-channel' ($F(2)= 6.01$, $P < 0.003$), 'elicit-suggestion' ($F(2)= 3.94$, $P < 0.022$) and 'suggestion' ($F(2)= 3.19$, $P < 0.045$).

Starting with the 'fragment' label, it appears that people who are highly influential utter less fragments than people who have low influence. This finding is in line with the finding from [Bales et al., 1951] who stated that people who are interrupted more than others are likely to be of a lower social status, and hence likely to be less influential. For the 'Back-channel' label it appeared that people who are 'Low' on dominance back-channel more than people who are 'high' on dominance. Both of these dialogue-act labels are related to the meeting process. The remaining two labels 'suggest' and the 'elicit-suggest' show that both types of utterances are uttered relatively more by people who are 'High' on dominance than by people who are 'Low' on dominance. Both the elicitation of suggestions, as well as making suggestions during a meeting, or a discussion, relate to the fact that people provide options, or ideas, that could be solutions to the problems, or issues at hand. This finding, hence seems to provide evidence for the hypothesis that dominance and argumentation are related.

The data was again transformed into a feature set for training some classifiers. For this experiment a data set was used containing 120 samples, out of which 25 were labelled 'High', 69 were labelled 'Normal' and 26 were labelled 'Low'. The results are shown in Table 76.

Given the majority class baseline of 57.5% it appears that, although some of the feature values differ significantly, the features themselves are unable to outperform the baseline. Also after applying a post-hoc feature analysis this turned out to be impossible[51].

[51]Note that the optimal feature set contains the 'fragment' and 'suggest' labels which, given the significance levels and their complementarity in distinctiveness (see Figure 40), is a logical choice.

FeatureSet	J48	SVM	NB
All Dialogue-acts	56.66	58.33	45
Fragment and Suggest*	55.83	57.5	53.3

Table 76: Results on automatic influence level classification using the fraction of dialogue act labels as features. $*$ = best subset.

9.3.3 TAS Relations and Influence

This section reports on attempts to relate the various relations that exist between nodes in the argument diagrams to the levels of influence. Similar to the previous sections, for each participant, for each meeting, the percentage of relation labels was sampled. The combined data resulted in a data-set of 59 participants, participating in 15 meetings (not in all meetings were discussions, nor did all participants participate in all discussions). 13 of the participants were labelled as 'High', 33 of them were labelled as 'Normal' and 13 of them were labelled as 'Low'.

An overview of the 95% confidence interval of the mean percentage of the six most frequently occurring relation labels is shown in Figure 41.

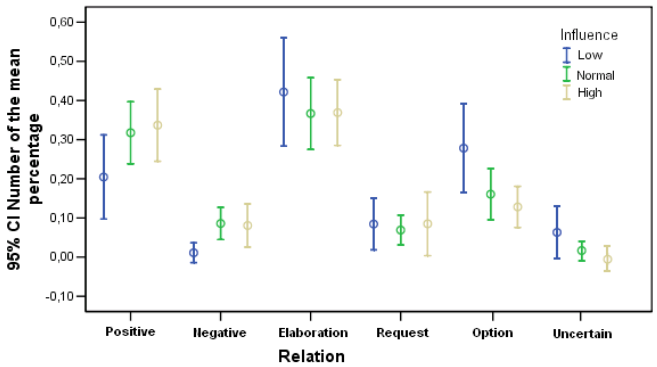


Figure 41: The mean number of relation occurrences per influence level.

ANOVA testing showed a significant dependency between the 'uncertain' relation category and the influence levels ($F(2)=3.52$, $p<0.037$). It appears that the lower the participant's influence, the more uncertain, or unclear, his or her contributions to the discussion are. Spearman's correlation coefficient ρ however did not prove significant. Here the relatively low number of samples is bothering us once more. For all the other relations we therefore cannot draw any hard conclusions.

When considering Figure 41 one could, however, construct the hypothesis that evaluative contributions, in terms of 'positive' and 'negative', seem to occur more frequently for higher influential participants. So if you give your opinion on things you might become more influential. But again, this is just a tendency that can be observed from the figure and this is not based on significant evidence.

Another interesting observation that can be made is that it seems that people of low influence seem to provide more 'options' to the discussions.

9.4 Cross-fertilizing features

A typical question we want to answer from a machine learning point of view when considering Figure 41, deals with the extent to which the different distributions of certain (class) labels are useful for the classification process. Even more, since the previous section also showed that indeed some regularities seem to exist between the level of influence of a participant and the way that argumentation enfolds in a discussion. This section therefore aims to investigate the usefulness of (the features of) one of the phenomena of influence and argumentation as predictor, or feature, for the other phenomenon in a machine learning context.

9.4.1 Predicting influence with argumentation

The first experiment tries to predict the influence level (dependent variable) making use of just the argumentation label distributions (independent variables).

As influence was measured on a meeting level, the feature vectors that contained the argumentation labels were also created on a meeting level by taking the label fraction distributions for the individual participants as feature values to predict the influence label of the associated participant. This resulted in 59 samples[52] with a baseline of 55.93%. Machine learning algorithms were trained and evaluated using 10-fold cross validation. The results are shown in Table 77.

FeatureSet	J48	SVM	NB
STA-WST-OTH-OIS-AIS-YIS (unbal)	55.93	55.93	54.24
STA-WST-OTH-OIS-AIS-YIS (bal)	25.64	25.64	25.64
STA-OTH-ISS (unbalanced)	54.23	55.93	52.54

Table 77: Results on automatic influence level classification using the fraction of argument labels as features.

From Table 77 it appears that on the balanced corpus none of the tested classifiers outperforms the baseline. Not with the class labels added as feature, nor with the features that predict the class label, nor after merging the different issues and the different statements.

To explore this finding, a multiple linear regression model was instantiated from the data. Not surprisingly it appeared that none of the coefficients proved significant, nor for the individual labels, nor after merging the statements and the issues (the stronger the correlation coefficients, the more discriminating the feature).

[52] 13 were labelled as 'High', 33 as 'Normal', and 13 as 'Low'.

9.4.2 Predicting argumentation with influence

For the second experiment the influence labels were used to see whether they could aid the prediction of TAS labels (both units and relations). So in this case the class labels were the TAS labels and the influence value of the speaker was added as a feature. The results are shown in Table 78.

Class	Feature set	J48	SVM	NB
Nodes	DOM	53.64	53.64	53.64
	QMT-ORT-L-LL-NS	73.53	68.21	64.05
	QMT-ORT-L-LL-NS-DOM	71.91	68.32	64.05
Relations	DOM	34.48	34.48	34.48
	TT	39.24	39.24	39.62
	TT-DOM	38.86	39.43	38.29
	TT-WT	44.95	39.80	44.00
	TT-WT-DOM	43.62	42.67	44.57

Table 78: Results on automatic TAS unit labelling with and without the dominance (DOM) feature. Features: TT = Target Type, WT = # Words in Target

The results indicate that the dominance feature does not seem to be of any use to the classifier. For the nodes of the TAS schema, the dominance feature itself does not score above the baseline of 55.95% (most frequent class is statement (484) amongst a total of 865 labels.). When adding the dominance feature to a set of more useful features, the performance does not increase either. For the relations of the TAS schema the baseline is set by the elaboration relation (181 occurrences amongst a total of 525 relations) to 34.4%. Again here the dominance feature does not prove useful, neither in combination with a set of other features that have proved useful in [Verbree et al., 2006a].

9.5 Conclusions

Given the results from the statistical investigations, the results on the classification performance and the rules that were induced, one could try to construct a tentative profile of how influential participants, as experienced by actual meeting participants, distinguish themselves from less influential participants. When considering the previous sections, one could say that:

- Influential participants seem to raise more issues.
- Influential participants leave the provision of options, or possible solutions, to others.
- Influential participants seem to provide more evaluative information with respect to the contributions of others.
- Influential participants seem to respond to statements from others that follow after Yes/No Issues.

- Influential participants significantly elicit and provide more suggestions for action over the course of a meeting.
- Influential participants significantly provide less back-channels over the course of a meeting.
- Influential participants seem to provide less 'other' TAS units.
- Influential participants provide fewer unfinished utterances, or speech fragments over the course of a meeting.
- Influential participants seem to resort later in a discussion to shorter turns.

So it seems that if a participant raises issues, elicits solutions, evaluates these solutions and then steers towards a choice amongst the possible solutions, one indeed gets an intuitive sense of a person who is highly influential, and who controls the course of discussion. On the other hand, if someone provides options, back-channels a lot to others, resorts to shorter contributions in the decision phase of the discussion indeed, then an intuitive profile of a less influential participant appears.

Exploitation of these profiles and the interrelation between both phenomena, however, do not prove to be sufficiently distinctive, in such a way that cross-fertilization of (features of) phenomena can yield machine learning algorithms to significantly improve their recognition performance. This result underlines that features have to correlate more than slightly with the phenomena of interest and also that 'just adding' features to the data set does not automatically improve the performance, in a sense that complementarity also plays a part.

10 Dominance Modeling

10.1 Targeted Objectives and Summary of Achievements

The overall goal of our work in dominance modeling is the design and implementation of methods to estimate participant dominance in small group meetings from single and multiple perceptual data (audio and video). In summary, the work in this research track produced the following achievements:

- Annotation of 11 sessions of the AMI meeting data set in terms of perceived levels of dominance.
- Annotation analysis for establishing research tasks and data sets with ground truth.
- Definition of several dominance tasks based on the analysis of the annotation. The tasks include the estimation of the most dominant person and the least dominant person. Additional tasks investigated the variability of the annotation. Analysis of the results where there was more intrinsic variability in the data set led to a systematic decrease in performance.
- Extraction of a large set of both low-level and mid-level audio-visual features. This set spanned video features (including compressed-domain features and visual focus of attention) and audio features (derived from head-set microphones).
- Investigation of unsupervised methods for estimating the most and the least dominant person in a meeting. The best performing single feature with the investigated unsupervised models was the total speaking length, which produced a classification accuracy of 85% and 86% for the most-dominant and the least-dominant tasks, respectively. Extensive studies were also conducted into how performance in dominance estimation is affected by a larger variation in perceived dominance by annotators.
- Investigation of supervised models and feature fusion (audio-only and audio-visual) for estimating the most and the least dominant person in a meeting . With a Support Vector Machine-based approach and combined features, the best achieved classification accuracy was 91% and 89% for the most-dominant and the least-dominant tasks, respectively.

10.2 Data Annotation and Task Definition

The AMI meeting data used for our experiments consisted of 11 meetings, divided into five-minute segments, which were provided for annotation, so that a total of 59 meeting segments were used. Five-minute segments were chosen since this would provide more data points for training and testing.

For each meeting, annotators were asked to rank the participants, from 1 (highest) to 4 (lowest), according to their level of perceived dominance. As well as an absolute ranking, annotators were also asked to rank proportionately with a total of 10 units, where more

units signified higher dominance. To account for cases where the rankings were difficult to allocate, a few questions about the confidence that the annotator had about their rankings of the most and least dominant person, and also about the proportionate ranking, were used. Following this, a set of detailed questions about each participant was requested from the annotators such as their degree of activity, timidness, dominance, and talkativeness. Finally, annotators were asked to explain, in free form, the personal criteria they used to decode dominance.

A total number of 21 annotators were used and where possible, were split into groups of 3 so that each group always annotated the same segments. For a given meeting, each annotator viewed only one five-minute segment (in other words, an annotator never judged more than one segment of the same meeting). Importantly, annotators were not given a prior definition of dominance, neither were they told what specific verbal or non-verbal cues to look for in order to make their judgments.

10.2.1 Analysis of the Annotations

The analysis of the annotation showed that a significant number of the meeting segments showed full agreement of the most dominant person. We found that in 34 of the meeting segments, all the annotators agreed exactly on the most dominant person. Furthermore, when these meetings were compared against the corresponding self-reported average confidence for the annotation, it was found that it was on average 1.7 (where 1 represents the highest confidence and 7 represents the lowest). This data subset represents almost 3 hours of meeting data with a reliable ground truth, where the agreement and confidence of the annotators was robust enough. An additional observation of interest is that in 24 out of the 34 cases, the most dominant person that was chosen by the corresponding annotators was assigned the role of project manager in the meeting activities.

We conducted similar analysis and found out that there were 23 additional meetings where 2 out of 3 annotators agreed on the most dominant person. This subset contains a larger intrinsic variation in the perceived dominance by human judges. Finally, a similar analysis showed that there 29 meetings with full agreement of the least dominant person. These data sets were used to define a number of classification tasks defined in the next subsection.

10.2.2 Dominance tasks and data subsets

Following our studies of the annotations, we decided to define a number of dominance classification tasks on different subsets of the AMI data, which incorporate more variability on the ground truth data for better understanding of how the performance might degrade. The tasks are the following:

- **Most dominant person task, full-agreement data**: The data set consists of 34 meetings where 3 annotators agree.
- **Most dominant person task, two-thirds-agreement data**: The data set consists of 23 meetings where only 2 annotators agree.

- **Most dominant person task, majority-agreement data**: The data set consists of 57 meetings where at least 2 annotators agree. This corresponds to the union of the full-agreement and the two-thirds-agreement data sets.
- **Least dominant person task, full-agreement data**: The data set consists of 29 meetings where 3 annotators agree.

10.3 Audio-visual Feature Extraction

In this year, we investigated features derived both from audio and from video. From audio, we adapted existing analysis techniques to extract a number of features to characterize the speaking activity of the meeting participants. From video, compressed-domain features were extracted from multiple cameras, and visual focus of attention features were derived from manually labeled data. The features are described more detail in the following.

10.3.1 Audio features

Audio features were extracted from two different sources: (1) four close-talk microphones attached to each of the participants (one per person), and (2) a distant microphone array placed at the center of the meeting table. In this year, most of the work on feature extraction focused on the first data source, i.e., the close-talk microphones. The starting point for audio feature extraction is an automatic, energy-based method for speaker turn segmentation, which uses the signal from each close-talk microphone to produce speaker turns for each participant. The following features were used in this year:

- **Speaking Activity.** A binary variable computed from audio that indicates the speaking / non-speaking status of each participant at each time step. This feature was extracted at 5 frames per second. The accumulated speaking activity for a person is called **Total Speaking Length (TSL)**.
- **Speaking Energy.** A real-valued variable also computed for each participant at each time step, also extracted at 5 frames per second. The accumulated speaking energy for a person is called **Total Speaking Energy (TSE)**.
- **Total Speaker Turns (TST).** The cumulative number of uninterrupted speaker turns for each person.
- **Total Successful Interruptions (TSI)**. This feature encodes the hypothesis that dominant people interrrupt others more often. The feature is defined by the cumulative number of frames that speaker $A = \{1, 2, 3, 4\}$ starts talking while another speaker $\{B : B \neq A\}$ is talking, and speaker B finishes his turn before A does, i.e. only interruptions that are successful are counted.
- **Total Speaker Turns without Backchannels (TSTwBC)**. This is a variation of the TST feature, computed as the cumulative number of turns that a speaker $A = \{1, 2, 3, 4\}$ takes such that the turn duration is longer that one second. The goal is to retain only those turns that are most likely to correspond to 'real' turns, eliminating

all short utterances which might be more likely to correspond to backchannels. Obviously, this is a simplifying assumption as we do not use transcribed speech.

- **Histogram of Turn Duration (TDHist)**. This feature, that models the distribution of turn duration, is first normalized by the total number of turns in a meeting. We used 11 bins, such that 10 bins were equally spaced at 1-second intervals, and the last bin included all turns of size greater than 10 seconds. Three different cases were tested: when all 11 bins were used (denoted 1:11), or only when the last 9 or 8 bins (denoted 2:11 and 3:11, respectively) were retained. The last two cases thus remove short turns.

10.3.2 Video features

Video features were of two types. In the first case, compressed-domain motion activity features were extracted from the four close-view cameras in the meeting room, one looking at one participant. In the second case, visual focus of attention features were derived from manually annotated data, as the work on improving automatic recognition of VFOA was also in progress. More specifically, the extracted visual features were the following:

- **Motion Activity.** A binary variable computed from compressed-domain video that indicates whether a participant is visually active (i.e., moving) or inactive at each time step (extracted at 25 frames per second). Three variations were tested, based on Motion Vectors, Coding Bitrate, and the combination Motion Vectors + Coding Bitrate. The accumulated motion activity for a person is called **Total Motion Activity Length** and can be of three types, depending on whether it is estimated from motion-only (**TMALM**), bitrate-only (**TMALB**), or their combination (**TMALC**).

- **Total Received Visual Focus of Attention (TVFOA)**. This feature corresponds to the cumulative number of events (rather than frames) that a participant $A = \{1, 2, 3, 4\}$ is looked at by another participant $\{B : B \neq A\}$.

10.4 Unsupervised most-dominant person classification

10.4.1 Most-dominant task with full-agreement data

Initially, we targeted the task of automatic classification of the most dominant person of a meeting. The data set, as already explained, consists of 34 five-minute meeting segments. We began by studying the performance of simple unsupervised dominance models, in which the accumulated features described in the previous section - i.e., TSL, TSE, TSI etc. - were computed for each participant, and the participant with the maximum value of a given feature was considered to be the most dominant person. Table 5 shows the classification accuracy of the various cases.

Regarding audio features, as seen in Table 79, total speaking length and total speaking energy perform well on classifying the most dominant person. An observation is that for the meetings where kappa agreement is not perfect (i.e., kappa less than 1 which indicates that not all the annotators agreed on the four-person rankings), speaking energy

Method	Classification Accuracy(%)
TSL	85
TSE	82
TST	62
TSTwBC	85
TSI	62
TMALM	59
TMALB	56
TMALC	62
TVFOA	71
Random Guess	25

Table 79: Performance of various unsupervised dominance models for the most dominant person classification task with full-agreement data.

sum seems to be a robust estimate (compared to speaking length) for dominance estimation. While the total number of turns did not perform as well, removing the short turns, likely related to backchannels (TSTwBC), performs as well as TSL. Finally, the number of interrruptions, used in isolation, did not perform as well as other audio features. All investigated audio features performed significantly better than chance (which would result in 25% classification accuracy).

Regarding video features, the motion activity features perform less well compared to the best audio features, but the results are considerably better than a random guess. This indicates that these features also have discriminative power. It was interesting to observe that bitrate performed better than motion, and that the combination worked the best. As a possible explanation, the residual bitrate is normally related to the amount of non-rigid motion, such as lip motion in the close-view camera case. This type of activity is usually not captured by motion vectors, since they are better at tracking translational motion, being derived from block motion compensation in video compression. Regarding the total received VFOA, this feature performed considerably better than the compressed-domain video features, but not as well as the best audio features. A closer look at the meetings, where speaking length and speaking energy sum fail, shows that the video features based on VFOA could be promising indicators of dominance.

In summary, these experiments indicate that, for the 34 meeting segments in which the most dominant person was reliably decoded by all annotators, some of our investigated audio cues were able to classify the most dominant person with good accuracy. In addition, our compressed-domain video features performed less well but still provided some discrimination, especially when used in combination, and our VFOA feature was also promising.

10.4.2 Team-Player Influence Model for most-dominant person classification

We also studied the team-player influence model (TPIM), which had showed promise from our previous work (see AMI deliverable 5.2). The team-player influence model

(TPIM) is a layered Dynamic Bayesian Network (DBN) in which each person (player) represents a hidden state variable that is linked to a hidden group variable (the team), which is affected by and affects the states of each player. In this model, a distribution over a parameter that represents the influence of each player on the group variable is automatically learned from data. A learned model will provide a continuous influence value for each participant, which could be used to rank participants in a meeting.

We implemented the TPIM using the Graphical Models Toolkit (GMTK), a DBN system for speech, language, and time series data. Specifically, we used the switching parents feature of GMTK, which facilitates the implementation of the two-level model to learn the influence values using the Expectation Maximization (EM) algorithm in an unsupervised manner. For the experiments reported here, the players' states were coded as observed and binary, corresponding to the speaking activity feature described in Section 10.3, i.e., the speaking/non-speaking status of each participant at each time step.

Table 80 gives the performance of the influence model on the most dominant person classification task with full-agreement data, for various choices of the number of group states (NG), a free parameter. As mentioned earlier, the number of states for the players was fixed at 2. The influence values were initialized equally for all players. From the results, we observe that we get the best classification accuracy when NG was 5, one more that the number of players. Although experiments were carried out with NG set to less than 4 and more than 10, the performance did not improved and hence has not been reported in the Table. It is clear that this parameter plays an important role in overall performance.

Number of Group States	Classification Accuracy (%)
4	32
5	50
6	38
7	38
9	41
10	29

Table 80: Classification Accuracy for TPIM for the most dominant person classification task with full-agreement data.

The TPIM using binary speaking activity as input did not perform as well as the unsupervised models presented in Table 79 that use audio features, more specifically w.r.t. the total speaking length (TSL). Analyzing the results in more detail, we observe the following:

- In 18 out of 34 meetings where there is a person who speaks a lot (50% or more of the total speaking time), the TPIM predicted the ground truth correctly in 10 cases, but the TSL did so in 15 cases.
- In the 17 cases where the TPIM failed, there were 12 cases where the person with the second highest speaking length has a similar influence value to the person with the highest speaking length.

- For the five cases where TSL failed, the TPIM correctly classified three of them. However, for the 17 cases where the TPIM failed, TSL performed correctly in 15 of them.

- The influence values predicted by the TPIM span a smaller range compared to those of TSL; this seems to have a negative effect on the TPIM discriminative power.

In summary, as the simpler model outperformed the team-player influence model, we continue our investigation using the simple unsupervised approaches.

10.4.3 Other Most Dominant Person Classification Tasks

We also addressed the other most-dominant-person classification tasks described in Section 10.2. The first one uses the two-third-agreement data set, i.e, the 23 meetings in which, out of the three annotators, two agree on one person $A \in \{1, 2, 3, 4\}$ and the other annotator judges $B, B \neq A$ as the most dominant. The second one uses the majority-agreement data set, i.e., the set of meetings where at least two of the three annotators agree, consisting of 34 + 23 = 57 meetings.

We decided to use three different ways of computing classification accuracy for these meetings. Let N denote the total number of meetings, and let n be the number of times the predicted most dominant be A and m be the number of times predicted most dominant is B. A first evaluation strategy, called Evaluation 1 (or Ev1 for short) computes the classification accuracy as n/N. An alternative strategy is Evaluation 2 (Ev2), which computes classification accuracy as $(2/3(n) + 1/3(m))/N$. Finally, a third strategy, Evaluation 3 (Ev3), computes classification accuracy as $(n + m)/N$. Ev1 assumes that there is only one correctly labeled most-dominant-person for each meeting (the one corresponding to the majority vote by the annotators) and is obviously the appropriate way to evaluate performance on the full-agreement data set. Ev2 assigns fractions of classification accuracy depending on whether a predicted person is either the 'most-voted' or ' least-voted' person by the annotators for a given meeting. It should be noted that with Ev2, the maximum achievable performance is always less than 100%. In our particular case, the maximum performance is 67% using the two-third-agreement data (23 meetings), and 86.5% for the majority-agreement data (57 meetings). Finally, Ev3 assumes both the 'most-voted' and the 'least-voted' most-dominant-person labeled by the annotators for a given meeting are correct, and thus the prediction of either of them is considered as correct.

In Table 81 we compare the performance of the various single features with the unsupervised model (which uses the maximum value of the single features) for the various meeting sets.

In Table 81, we reproduce the results of Table 79, which correspond to the full-agreement data set, to ease comparisons between the various data sets. For the two-third-agreement data set (23 meetings) TSL and TSTwBC have the best classification accuracy with Ev1 and Ev2 but with Ev3 TVFOA performs the best, which suggests that for this data set, the visual focus predicted slightly better either of the two most dominant people. For the majority-agreement data set (57 meetings), TSL and TSTwBC are the best performing features for all three evaluations.

Feat	*Ev*1 (34)	*Ev*1 (23)	*Ev*2 (23)	*Ev*3 (23)	*Ev*1 (57)	*Ev*2 (57)	*Ev*3 (57)
TSL	85	65	49	83	77	71	84
TSE	82	61	45	74	74	67	79
TSTwBC	85	56	47	84	75	71	86
TST	62	43	38	70	54	52	65
TSI	62	44	38	70	54	52	65
TVFOA	71	48	45	87	61	60	77
TMALC	62	35	38	78	51	52	68

Table 81: Classification accuracy for the most dominant person classification task with various data sets and unsupervised models.

Overall, the inclusion of the data that is intrinsically more ambiguous with respect to the highest perceived dominance results in a more challenging task (compare the results for the evaluation strategy Ev1 for all features and the 34-, 23-, and 57-meeting data sets). On the other hand, the evaluation strategy Ev3, that assumes that more than person can be most-dominant, brings the performance of most features for the 57-meeting set back to the same level they had for the 34-meeting set and Ev1.

10.5 Supervised most-dominant person classification

The previous section indicated the feature fusion might produce better results for all dominance estimation tasks. We investigated a discriminative method in this section.

10.5.1 Most-dominant task with full-agreement data

A closer look at the meetings where total speaking length or total speaking energy failed indicates that in some cases speaking turns or motion activity predicted the most dominant person correctly. This motivated us to look at feature fusion to improve the prediction accuracy. We use a Support Vector Machines (SVM) approach defined on single and multiple audio and video cues. In all cases, the dataset consisted of $34 \times 4 = 136$ data points, corresponding to 4 persons from each of the 34 meetings. Since the data available for learning is small, a leave-one-out strategy was employed to train and test the SVM model. For classification of the most dominant person, the SVM score for each of the 4 participants was computed, and the person with the extreme score was classified as the most dominant one.

In Table 82, the SVM performance on single features is compared with the results obtained with the unsupervised models of the section 10.4.1. Using single features, the performance is essentially the same using both methods, which is in accordance with our intuition. In the case of the SVM training with TSL, there was a 3% drop in classification accuracy, likely due to the small number of training examples. Regarding TDHist, the 2:11 feature, which has the backchannels (turns less than 1 sec) removed, performed the best.

Feature	Classification accuracy Unsupervised Model(%)	Classification accuracy SVM(%)
TSL	85	82
TSE	82	82
TST	62	62
TSTwBC	85	NA
TMALC	62	62
TSI	62	62
TVFOA	71	71
TDHist(1:11)	NA	77
TDHist(2:11)	NA	79
TDHist(3:11)	NA	77
Turns < 1s	NA	26
Turns > 5s	NA	79
Turns > 10s	NA	77
ENTHistTDHist	NA	74

Table 82: Most-dominant classification accuracy for SVM-based method on single features and the full-agreement data set.

Previously, we looked at the Turn Duration Histogram (TDHist) normalized by the total turns in a meeting. We used 11 bins, such that 10 bins were equally spaced at 1-second intervals, and the last bin included all turns of size greater than 10 seconds. Three different cases were tested: when all 11 bins were used (denoted 1:11), or only when the last 9 or 8 bins (denoted 2:11 and 3:11, respectively) were retained. We also looked at the performance of shorter and longer turns as single features. It was interesting to observe that turns less than 1 second which can be thought of as "backchannels" did not have any dicriminative power to decide on the most dominant person, while the longer turns had good discriminative power. Some of the results are also listed in Table 82.

Another interesting feature we looked at was the entropy of the histogram of TDHist feature (ENTHistTDHist). The bin size of the histogram was chosen to be 256. This feature performed well (classification accuracy of 74%) as seen in Table 82. It is to be noted that TDHist is 11-dimensional whereas ENTHistTDHist is one-dimensional, hence we get a computational advantage for the same performance. One reason why this feature works could be that for the most dominant person, the TDHist feature is like a ramp and hence the histogram of a ramp translates into a uniform distribution whose entropy is large. For the non-most-dominant class, the TDHist feature tends to be more uniform and hence its histogram translates into a sharper distribution, whose entropy is close to zero.

The single features were later combined into multi-dimensional representations. The combinations that yielded better performance appear in Table 83. We observe that feature fusion proves beneficial. Though the TST is not very discriminative as a single feature, it becomes more so when combined with the TSE alone or with the TSE and TSL. We observed a marginal improvement of 3% in classification accuracy with these fused features. Unfortunately, we also observed that the compressed-domain video features, when combined with other audio features, did not yield any further improvement in classifica-

tion performance. combinations which yield an absolute performance improvement of 6% with respect to the performance obtained with Total Speaking Length (85%). Two of these feature combinations are multimodal (combining VFOA and various audio features), and two more are currently extracted in a fully automatic manner using audio-only features.

Feature	Classification accuracy SVM(%)
TSE, TST	88
TSL, TSE, TST	88
TDHist,TSE,TST	85
TSE,TST,TSI	88
TSE,TST,TVFOA	88
TSL,TSE,TST,TSI	88
TDHist,TSE,TST,TVFOA	88
TDHist,TSE,TST,TSI	91
TSE,TST,TSI,TVFOA	91
TDHist,TSE,TST,TSI,TVFOA	91
ENTHistTDHist,TSE,TST,TSI	91

Table 83: Most-dominant classification accuracy for SVM-based method on fused features and the full-agreement data set.

10.5.2 Other Most Dominant Person Classification Tasks

When we studied the performance of SVM learned on single features, we tried two options of training the SVM. One is to use the 'clean' data (i.e., data from the 34 meetings where there is full agreement on who is the most dominant person is) for training, and the other option is to use the 'noisy' data (i.e., data from the 57 meetings, where a minimum of 2 annotators agree). We observe that using the clean data for training improves the performance for almost every feature.

We then combined the single features into multi-dimensional features and trained the SVM approach. Some of the best performing combinations are listed in Table 84. We observe that, for the two-third agreement data set (23 meetings), the (TDHist,TSE,TST,TVFOA) combination improves the classification accuracy by 4% with Ev3. Furthermore, for the majority-agreement data set (57 meetings), the combinations (TSE, TST, TVFOA) and (TDHist, TSE, TST, TVFOA) improve the classification accuracy by up to 2% in Ev2 and by up to $4-5\%$ with Ev3. Feature fusion, therefore, has also proved to be advantageous for these more challenging data sets.

10.6 Least-dominant person classification

We conducted experiments on the least dominant person classification task with full-agreement data (29 meetings). The unsupervised model is modified so that that now the person that corresponds to the lowest proportion of the feature among all participants

Feat	*Ev*1 (34)	*Ev*1 (23)	*Ev*2 (23)	*Ev*3 (23)	*Ev*1 (57)	*Ev*2 (57)	*Ev*3 (57)
TSL(noisy)	82	65	49	83	75	69	82
TSL(clean)	85	65	49	83	77	71	84
TSE,TST,TVFOA	91	57	46	83	77	73	88
TSE,TST,TSI,TVFOA	88	57	45	78	75	71	84
TDHist,TSE,TST,TVFOA	91	52	46	87	75	73	89

Table 84: Classification accuracy for the most dominant person classification task with various data sets and supervised models.

is classified as least dominant. The supervised model is trained on the new two classes (least- vs. non-least dominant). The classification accuracy of the features is shown in Table 85. We observe that feature fusion proves beneficial for one case which yields an absolute performance improvement of 3% with respect to the performance obtained with Total Speaking Length (86%). It is interesting to notice that the feature combinations that performed very well for the most-dominant person classification task did not necessarily perform well for the least-dominant case.

Feature	Classification accuracy Unsupervised(%)	Classification accuracy SVM (%)
TSL	86	86
TSE	72	62
TST	72	72
TSI	52	52
TMALC	45	45
TSTwBC	86	NA
TDHist, TSE, TST, TVFOA	NA	89

Table 85: Least-dominant-person classification accuracy with full-agreement data set. Single and multiple features results.

10.7 Conclusions

In this period, we have shown that simple audio-visual features can be used for dominance estimation. In addition, our studies into feature fusion have shown that some are indeed complementary. In particular, we were able to find both audio-visual and fully audio combinations of features that gave a high performance for estimating the most dominant person with a classification accuracy of 91% using a fully audio multi-dimensional feature. We also obtained a performance of 89% for the least dominant person task using audio-visual feature fusion.

Regarding the annotation of the AMI data set, we have found that while there is variablity in the annotations, a significant proportion of the meetings showed a reliable general

consensus for who the most dominant and the least dominant people were. In addition, including variabity in the dominance task led to systematic decreases in performance.

11 Speech Indexing and Retrieval

This sections describes two important activities in speech indexing and retrieval: subsection 11.1 deals with the spoken term detection system submitted to NIST STD 2006 evaluations and subsection 11.2 reports from the SIGIR Workshop on Searching Spontaneous Conversational Speech held as part of the 2007 ACM SIGIR Conference in Amsterdam.

11.1 Spoken term detection system based on combination of LVCSR and phonetic search

This text presents the Brno University of Technology (BUT) system for indexing and search of speech, combining LVCSR and phonetic approach. It brings a complete description of individual building blocks of the system from signal processing, through the recognizers, indexing and search until the normalization of detection scores. It also describes the data used in the first edition of NIST Spoken term detection (STD) evaluation. The results are presented on three US-English conditions - meetings, broadcast news and conversational telephone speech, in terms of detection error trade-off (DET) curves and term-weighted values (TWV) metrics defined by NIST.

The system combines two techniques:

- Large vocabulary continuous speech recognition, where the recognition and indexing unit is a word.
- Phoneme recognition, where phonemes are recognized, and, for faster access, triphoneme sequences are indexed.

The theoretical basis of the search were described in [Burget et al., 2006] and we do not deal with them in detail in this paper. Here, we concentrate on our submission for the NIST Spoken Term Detection (STD) Evaluations organized for the first time in 2006.

11.1.1 NIST STD evaluations 2006

The first edition of Spoken term detection evaluation was organized to facilitate research and development of technology for finding short word sequences rapidly and accurately in large heterogeneous audio archives[53]. In this paper, we will deal with STD for US English[54].

Data There were three kinds of data with the following amounts available for both the development and evaluation:

- broadcast news (BCN) – 2.2 hours,
- conversational telephone speech (CTS) – 3 hours

[53] `http://www.nist.gov/speech/tests/std/`

[54] Arabic and Mandarin were the two other languages analyzed in this evaluation.

- meeting speech (MTG) recorded over multiple distant microphones (MDM) – 2 hours.

For all sets, NIST has defined 1100 search-terms[55] having 1, 2, 3 and 4 words:

- 42 of them do not appear in any of BCN, CTS and MTG data
- 898 of 1100 appear in BCN with ≈4900 occurrences
- 411 of 1100 appear in CTS with ≈5900 occurrences
- 241 of 1100 appear in MTG with ≈3700 occurrences
- 160 of 1100 appear in all three BCN, CTS and MTG.

Examples of terms are:
"dr. carol lippa", "bush's father george bush", "thousand kurdish", "senator charles", "nato chief", "every evening", "kostunica", "audio", "okay".

Evaluation metrics The main mean for comparison of different systems were detection error trade-off (DET) curves, displaying, for various detection thresholds θ, the false alarm probability $P_{FA}(\theta)$ on x-axis and miss probability $P_{MISS}(\theta)$ on the y-axis:

$$P_{MISS}(\theta) = \underset{term}{avg}\{1 - N_{correct}(term, \theta)/N_{true}(term)\} \quad (31)$$

$$P_{FA}(\theta) = \underset{term}{avg}\{N_{spurious}(term, \theta)/N_{NT}(term)\} \quad (32)$$

where $N_{correct}(term, \theta)$ is the number of correct detections of *term* with a score greater or equal to θ, $N_{spurious}(term, \theta)$ is the number of spurious (incorrect) detections of *term* with a score greater or equal to θ, $N_{true}(term)$ is the number of occurrences of *term* in corpus and $N_{NT}(term)$ is the number of opportunities for incorrect detection of *term* which is equal to length of the corpus in seconds minus $N_{true}(term)$.

NIST defined so called Term-Weighted Value $TWV(\theta)$ metric to "score" a system by one number. Term weighted value is evaluated by first computing the miss and false alarm probabilities for each term separately, then using these and a pre-determined prior probability to compute term-specific values, and finally averaging these term-specific values over all terms to produce an overall system value:

$$TWV(\theta) = 1 - \underset{term}{avg}\{P_{MISS}(term, \theta) + 999.9\, P_{FA}(term, \theta)\}$$

The threshold θ_M is found on development data by maximization of $TWV(\theta)$. $TWV(\theta_M)$ is then computed on evaluation data with θ_M threshold and denoted as *ATWV* (actual TWV).

11.1.2 The system

The overall scheme of the system is in Figure 42.

[55] "quoted" queries where "quoted" refers to Google and similar search engines and means that no other word(s) can appear inside the query.

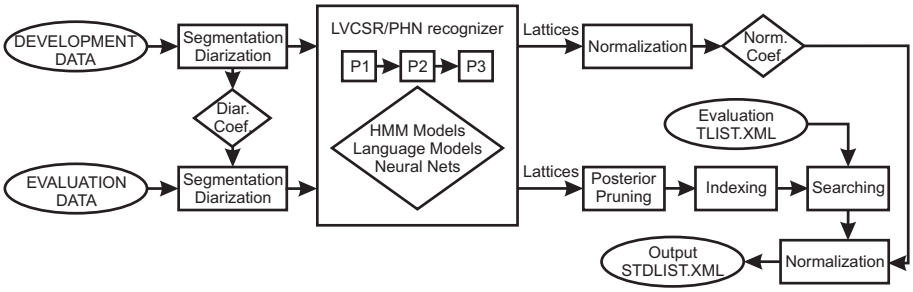


Figure 42: Scheme of spoken term detection system for NIST STD 2006 evaluations

.

Signal processing First, all NIST speech files were converted to raw format using `sox`. Segmenting speech into speech and silence was done by our neural net based phoneme recognizer [Schwarz et al., 2007]. All phoneme classes were linked to 'speech' class. CTS data were segmented according to energy in channels and speech/non-speech segmentation. The diarization for BCN and MTG data was done by David van Leeuwen. He used a Bayesian Information Criterion (BIC) based speaker segmentation and clustering system developed for the AMI RT06s speaker diarization evaluation [van Leeuwen and Huijbregts, 2006]. 12 Perceptual Liner Prediction (PLP) features plus log energy were used as features, and he modeled clusters using a single Gaussian with full covariance matrix.

The data was split into shorter segments using the following heuristics: (1) in silences longer than 0.5s (output of speech/non-speech detector), (2) when speaker changed (in BCN and MTG), (3) if a segment was longer than 1 minute, it was split into 2 parts in silence closest to the center of segment.

Recognition Segmented data was than processed by word (LVCSR) and phoneme (PHN) recognizers.

11.1.3 LVCSR – the general scheme

The STD 2006 LVCSR system is a simplified version of AMI LVCSR system used for NIST RT 2006 evaluations [Hain et al., 2006]. It has has same structure for all tasks: CTS, BCN and MTG; the differences lie in acoustic and language models only. The scheme of LVCSR is on Fig. 43. The system operates in 3 passes of feature extraction and recognition:

In the first pass (P1), the front-end converts the segmented recordings into feature streams, with vectors comprised of 12 Mel-frequency Perceptual Liner Prediction (MF-PLP) features and raw log energy, first and second order derivatives are added. After, a cepstral mean and variance normalization (CMN/CVN) is performed on a per-channel basis with given segmentation. The first decoding pass yields initial transcripts that are subsequently used for estimation of vocal tract length normalization (VTLN) warping factors. The feature vectors and CMN and CVN are re-computed. The second pass (P2) processes the

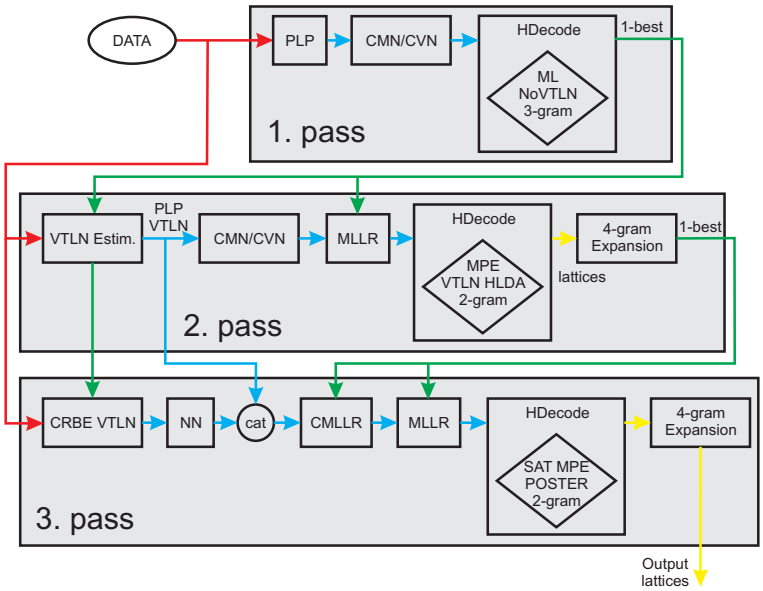


Figure 43: Three passes of the recognition.

new features and its output is used to adapt models with maximum likelihood linear regression (MLLR). Bigram lattices are produced and re-scored by trigram and four-gram language model. In the third pass (P3), posterior features [Schwarz et al., 2007, Grezl et al., 2007] are generated. The output from the second pass is used to adapt models with Constrained MLLR (CMLLR) and MLLR. The bigram lattices with posterior features are produced and finally re-scored with trigram and four-gram language model.

11.1.4 Feature extraction and acoustic modeling

All systems use standard cross-word tied states HMM using Mel-PLP's generated in classical way with: 23 filter-bank channels for BCN and MTG systemor 15 filter-bank channels for CTS. The resulting number of cepstral coefficients is always 13. The following techniques are used in HMM training: (1) CMN/CVN is applied per speaker, (2) VTLN warping factors are computed using Brent search method and features are recomputed, (3) deltas, double-and triple-deltas are added into the basic PLP feature stream, so that the feature vector has 52 dimensions. Heteroscedastic linear discriminant analysis (HLDA) is estimated with Gaussian components as classes. HLDA is estimated to reduce the dimensionality to 39. (4) Posterior features - two kinds of posterior features are used:

LC-RC Posterior features The LC-RC system [Schwarz et al., 2007] splits 310 ms temporal context in each filter-bank output into two halves and each half is processed by one neural net (NN) producing phoneme-state posteriors. These are merged by the third neural net. The resulting vector size is 135 (45 phonemes each with 3 states). After log and dimensionality-reduction by Karhunen-Loeve transform (KLT) to 70 dimensions (this step was necessary to fit the following HLDA statistics into memory), HLDA is estimated with Gaussian components as classes. HLDA was estimated to reduce the dimensionality

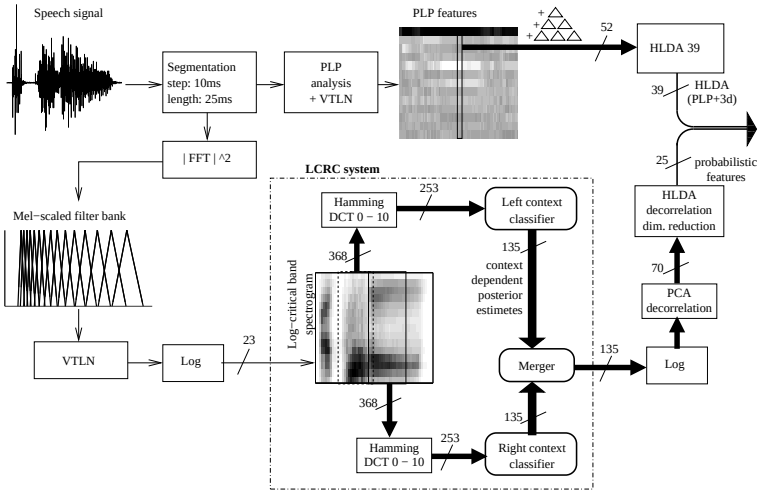


Figure 44: Features used in the recognition system.

to 25. The resulting features are concatenated with PLP feature stream (25+39=64) and mean and variance normalized. The procedure is outlined in Fig. 44.

Bottle-neck LC-RC features Bottleneck LC-RC differ from basic LC-RC in the last NN: the merger. It is a 5-Layer NN with middle layer containing 35 neurons only [Grezl et al., 2007]. Non-linearly compressed information here is used as output. The HLDA is estimated to de-correlate and to reduce the dimensionality from 35 to 25.

Again, the resulting features are concatenated with PLP features (25+39=64) and mean and variance normalized.

Training of posterior features At first, the neural network training with CMN/CVN at the input was done on 30h of VTLN normalized data used for training of LVCSR acoustic models. Using these nets, full features were generated for all the data. The output was concatenated with PLP VTLN HLDA feature stream. The CMN/CVN were recomputed again and the models were trained by single-pass re-training. Further, the models were re-clustered and trained by the mixing-up procedure from 1 to N Gaussians. The optimal numbers of Gaussians were tuned for each task independently, the resulting numbers of Gaussians are 18 for MTG and BCN, 26 for CTS.

Speaker-adaptive training (SAT) One single CMLLR transform was trained per each meeting channel. Features were mapped to unique SAT space by CMLLR and 8 iterations of ML-training (standard Baum-Welch) were run. After, new CMLLR transforms were trained, features transformed and 8 ML-iterations followed. And once more, so that the number of CMLLR+re-training macro-iterations was 3.

Discriminative training The models were re-trained in 15 iterations of Minimum Phone-Error (MPE) training [Povey, 2004]. The alternative hypotheses for MPE were generated

task	P1	P2	P3
BCN	Basic PLP HMM	VTLN HLDA MPE	VTLN LC-RC SAT MPE
MTG	Basic PLP HMM	VTLN HLDA MPE	VTLN Bottleneck-LC-RC SAT MPE
CTS	HLDA	VTLN HLDA MPE	VTLN Bottleneck-LC-RC SAT MPE

Table 86: The acoustic models used in different steps for each task.

by much simpler system including just ML-trained models on PLP+HLDA without any adaptation. In case of SAT-MPE-training, we did not re-train the CMLLR transforms.

Table 86 outlines the acoustic models used in P1–P3 for different tasks.

11.1.5 Language models

The training of 4-gram language models was done at University of Sheffield by Vincent Wan. See below for the training data used. All language models were trained using the same data. The the perplexity was maximized for each task independently.

11.1.6 Phoneme models

Phoneme recognition was based on the same features and models as LVCSR. Only the recognition network was changed to context dependent phoneme (triphone) loop (with context independent output ie. the output is phonemes) with phoneme bigram language model.

11.1.7 Decoding and posterior pruning

The decoding was performed using the standard LVCSR decoder `HDecode` from University of Cambridge. Generated lattices took significant space, so the posterior pruning was used for lattice size reduction. LVCSR and PHN lattices were pruned using different pruning factors.

11.1.8 Indexing and Search

During indexing, word lattices are converted to forward index: each word-hypothesis (the word, its confidence, time and nodeID in the lattice file) is stored in a hit list. Forward index is then converted to inverted index which is sorted by words and by confidences of hypothesis. To save space and gain in speed of access, lattices are converted to binary format [Burget et al., 2006]. Phoneme lattices are also converted to forward index, the indexing units are phoneme trigrams (tri-phonemes). Forward index is also sorted to inverted index and lattice are converted to binary format.

In search, the term is first split to words (tokens). These are checked against the LVCSR dictionary and divided into in-vocabulary (IV) and out-of-vocabulary (OOV). IV tokens

are searched in inverted index to estimate their position in latices and then they are verified in the lattice (using token passing). OOV tokens are converted to phonemes. Automatic grapheme-to-phoneme (G2P) tool based on rules is used for the conversion. Then the phoneme string is split to a train of overlapped tri-phonemes. Then they are also searched in inverted index (phoneme) and verified in lattice (phoneme). OOVs shorter than 3 phonemes (in total) are not searched and are dropped. If all tokens are successfully verified, the time and score is produced. Score is computed as the sum of IV (LVCSR) part and OOV (PHN) part. IV scores are computed (by Viterbi approximation) using likelihood ratio in word lattice and then normalized. OOV scores are computed (by Viterbi approximation) using likelihood ratio in phoneme lattice and then normalized.

11.1.9 Training data

The training data for acoustic models was the following:

- for BCN, the `ihmtrain05` training set from NIST RT'06 evaluations [Hain et al., 2006] was used - it is a mixture of four meeting corpora, the NIST, ISL, ICSI and a preliminary release of the AMI corpus. In total, there are 112h of data. No BCN data were used.
- for MTG, the `mdmtrain05` training set from NIST RT'06 evaluations [Hain et al., 2006] was used. The crosstalk parts were removed and beam-forming to one superchannel was done. In total, there are 63h of speech.
- for CTS, `ctstrain04` - a subset of h5train03 set defined at Cambridge was used, in total 277h.

For language model training, done by Vincent Wan at the University of Sheffield, several resources were used (the numbers give the size of the corpus in megawords): Swbd/CHE 3.5, Fisher 10.5, Web (Swbd) 163, Web (Fisher) 484, Web (Fisher topics) 156, BBC - THISL 33, HUB4-LM96 152, SDR99-Newswire 39, Enron email 152, ICSI/ISL/NIST/AMI 1.5, Web (ICSI) 128, Web (AMI) 100, Web (CHIL) 70.

Grapheme to phoneme transcription rules were trained on AMI and BEEP pronunciation dictionaries.

The phoneme recognizer for segmentation was trained on Hungarian SpeechDat-E [56] for BCN, `ihmtrain05` for BCN and `mdmtrain05` for MTG. LC-RC and Bottle-neck nets for generation of posterior features used the same training data as acoustic models.

11.1.10 Normalization

The normalization serves to make scores of different queries comparable (note that NIST scores STD systems with *one single threshold*). Our normalization is based on contributions of phonemes to normalization factors:

$$s_N(KW) = s(KW) - G - F\, len(KW) - P_1|p_1| + ... + P_K|p_K|,$$

[56] Eastern European Speech Databases for Creation of Voice Driven Teleservices: `http://www.fee.vutbr.cz/SPEECHDAT-E/`

task	EVAL ATWV Merged	EVAL MTWV Merged	EVAL MTWV LVCSR	DEVEL MTWV Merged
BCN	0.654	0.655	0.630	0.702
CTS	0.523	0.534	0.530	0.558
MTG	0.054	0.073	0.069	0.295

Table 87: Minimum (M) TWV and actual (A) TWV values for individual and merged systems.

where $s(KW)$ is raw score of the keyword, $s_N(KW)$ is the normalized score, $len(KW)$ is length of the keyword and $|p_1| \ldots |p_K|$ are counts of individual phonemes in the keyword. G (a constant), F (length-dependent factor) and $P_1 \ldots P_N$ (phoneme-dependent factors) need to be trained: First, for large set of keywords, we derive scores for hits and false alarms (FA) on the development set. The scores corresponding to each keyword are used to construct pairs of (HIT, FA). For each pair, an equation is generated:

$$\frac{s(HIT) + s(FA)}{2} = G + F\, len(HIT) + P_1|p_1| + ... + P_K|p_K|,$$

where the left side represents an optimal threshold for given (HIT, FA) pair. We solve the over-defined set of equations in minimum square error sense and use the resulting factors to normalize scores. The normalization coefficients were trained on the respective (BCN, CTS, MTG) part of NIST STD 2006 development data.

11.1.11 Results

The results of LVCSR systems for different tasks in terms of word error rate (WER) evaluated on the development sets, are the following: BCN 21.03%, CTS 22.83% and MTG 46.65%. The oracle results obtained by scoring the path in lattice that matches the best the reference, are respectively: BCN 9.06%, CTS 8.32% and MTG 21.79%. It is obvious that while BCN and CTS results are good and comparable to the state-of-the-art, the recognition on meetings is worse. This is due to the MDM condition, for which all the systems in NIST RT'06 evaluation performed quite poorly.

The STD results on all three conditions in terms of DET curves on development data can be seen in Fig. 45 and the results in terms of TWV are summarized in Table 87. First, we can see that the results on meetings are even worse than for the development data suggesting a problem with the data. Unfortunately, we are not able to analyze this in detail, as NIST does not intend to provide word transcriptions for the evaluation data. In the other tasks, the results were satisfactory and we have seen the actual TWV not differing substantially from minimum TWV – a sign of good estimation of the optimal threshold. Except for BCN, we see minimum effect of merging phonetic search with LVCSR, this is however caused by the term-lists provided – in CTS data, we have counted only 6 OOVs out of all 1100 requested terms.

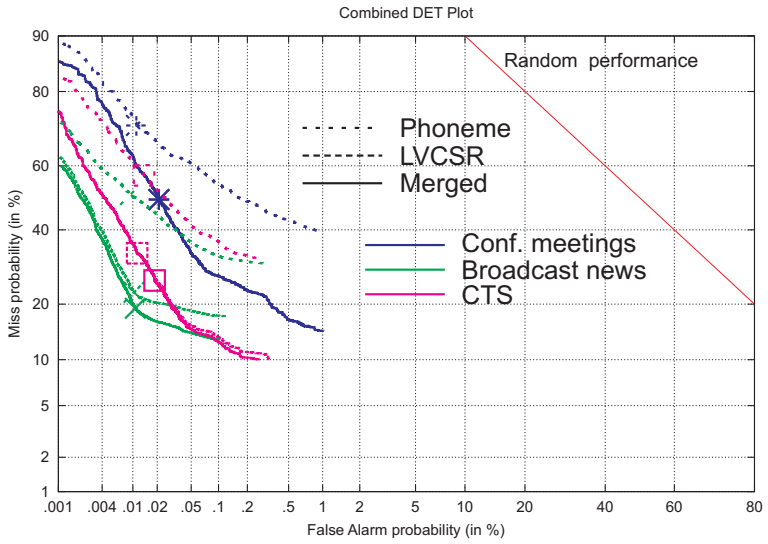


Figure 45: DET curves for development US English data: LVCSR, phoneme-based and merged systems

11.1.12 Conclusions

The STD evaluation confirmed the usability of our STD system and provided us with the opportunity to compare it to other labs working in the field. The evaluation provided us also with several technical lessons, such as that using 4-gram expansion is only slightly better than 3-gram expansion, posterior pruning of LVCSR lattices shortens DET but does not decreases TWV significantly, etc.

In future, we need to work on the normalization - the scheme we implemented is a basic one, we can experiment with NN, calibration methods, etc.

CPU time and memory footprint needed are also the primary issue – despite its good accuracy, our system was far too slow compared to the other in the evaluation.

When designing the system for a real oriented user, we also need to take into account other user requirements, such as signal pre-processing, entering queries and combination with other speech search modalities.

11.2 SIGIR Workshop on Searching Spontaneous Conversational Speech

The SIGIR Workshop on Searching Spontaneous Conversational Speech was held as part of the 2007 ACM SIGIR Conference in Amsterdam. The workshop program was a mix of elements, including a keynote speech, paper presentations and panel discussions. This brief report describes the organization of this workshop and summarizes the discussions.

11.2.1 Background

Nearly a decade ago, we learned from the Text Retrieval Conference's Spoken Document Retrieval track that searching speech was a "solved problem." Three factors were key to

this success:

- Broadcast news has a ”story” structure that resembles written documents.
- The redundancy present in human language means that search effectiveness held up well over a reasonable range of transcription accuracy.
- Sufficiently accurate Large-Vocabulary Continuous Speech Recognition (LVCSR) systems had been built for the planned speech of news announcers.

The long-term trend in speech recognition research has been toward transcription of progressively more challenging sources. Over the last few years, LVCSR for spontaneous conversational speech has improved to a degree where transcription accuracy comparable to what was previously found to be effective for broadcast news can now be achieved for a diverse range of sources. This has inspired a renaissance in research on search and browsing technology for spoken word collections in communities focused on:

- Archived cultural heritage materials (e.g., interviews and parliamentary debates).
- Discussion venues (e.g., business meetings and classroom instruction).
- Broadcast conversations (e.g., in-studio talk shows and call-in programs).

Test collections are being developed in individual projects around the world, including AMI/AMIDA and CHIL (recorded meeting projects funded by the EU under the 6th Framework Program), IM2 (a Swiss recorded meeting project), MALACH (a NSF-funded project in the USA working with oral history), CHoral (a cultural heritage project in the Dutch NWO-funded programme CATCH), and GALE (a DARPA-funded project in the USA working with broadcast conversations). Some comparative evaluation activities for speech search technology are ongoing, including the Spoken Term Detection (STD) evaluation run by the National Institute for Standards and Technology (NIST) in the USA and the Cross-Language Evaluation Forum’s Cross-Language Speech Retrieval track in Europe.

Each of the research communities involved in the initiatives mentioned above has established venues for agenda setting and for comparison of research results. For recorded meetings, this has included the MLMI workshops, and the NIST Rich Transcription evaluation, and the CLEAR evaluation sponsored by NIST and CHIL. Research on cultural heritage materials has recently been reported at workshops at the 2007 conference of the Association for Computational Linguistics in Prague and at the 2007 User Modeling conference in Corfu, Greece. For broadcast conversations, the DARPA GALE program (which includes research teams in North America, Europe and Asia) has to date been a principal research venue. Cross-cutting workshops have been held before at SIGIR (in 2001) and at the Human Language Technologies conference (in 2004), and a EU/NSF working group on spoken word archives recently identified several research issues related to the accessibility of recorded speech [Goldmann and Renals et al., 2005]. The time therefore seemed right to look more broadly across these research communities for potential synergies that can help to shape the information retrieval research agenda.

11.2.2 Before the Workshop

In the call for participation, contributions on a range of cross-cutting issues were solicited, including segmentation, content characterization, classification, exploiting multimodality, search effectiveness, interaction design, evaluation, and broader issues (e.g., applications, intellectual property, privacy). We invited fifteen experts from industry and academia to serve on the workshop's program committee. On the basis of their recommendations, seven papers that together spanned the identified topics were accepted.

On July 16, Technology Review published an interview with Peter Norvig (head of Google Research) in which he remarked on the key role of speech retrieval technology for providing access to large collections of multimedia materials [Greene, 2007]. Eleven days later, we met in Amsterdam to take up that challenge.

11.2.3 During the Workshop

Thirty researchers with a broad range of experience and expertise participated in the workshop. The program included a mix of elements designed to maximize interaction among participants from diverse backgrounds.

Keynote Mark Maybury, Executive Director of MITRE's Information Technology Division (USA), led off the workshop with a keynote address. He began by summarizing the challenges posed by searching spontaneous conversational speech. Two MITRE efforts were then presented to illustrate some of those challenges: Audio Hot-Spotting and Cross-Language Automatic Speech Recognition. Some promising opportunities for future research were outlined as well. The keynote session was followed by a discussant, Gareth Jones (Dublin City University, Ireland).

Presentations and Panels Table 88 briefly summarizes the seven research papers that were presented; full titles, author lists and abstracts are available on the workshop's Web page[57], and the full text of each paper is available in the workshop proceedings [de Jong et al., 2007]. In addition to the paper presentations, one invited presentation (by Doug Oard, entitled *Who needs this?*) was included to stimulate discussion of interactions between user needs and technical capabilities. Two panels discussion we interleaved with the more formal presentations. The first, on "What new technologies do we need?" included Pavel Ircing, Marijn Huijbregts, Martha Larson, and Jonathan Mamou as panelists, with Stephan Raaijmakers as moderator. The second, on "Research directions" included Ken Church, Jon Fiscus, Franciska de Jong and Mark Maybury as panelists, with Doug Oard as moderator.

Discussion Themes Sessions were structured to maximize opportunities for discussion, and a wide range of both high-level and detailed issues were addressed. The summary below is an effort to draw together some of the broader themes that emerged.

[57] http://hmi.ewi.utwente.nl/sscs/

Authors	Title
Cuendet et al.	*An Analysis of Sentence Segmentation Features for Broadcast News, Broadcast Conversations, and Meetings*
Fiscus et al.	*Results of the 2006 Spoken Term Detection Evaluation*
Jones et al.	*Examining the Contributions of Automatic Speech Transcriptions and Metadata Sources for Searching Spontaneous Conversational Speech*
Kim et al.	*Advances in SpeechFind: CRSS-UTD Spoken Document Retrieval System*
Larson et al.	*Supporting Radio Archive Workflows with Vocabulary Independent Spoken Keyword Search*
Olsson	*Improved Measures for Predicting the Usefulness of Recognition Lattices in Ranked Utterance Retrieval*
van der Werff et al.	*Evaluating ASR Output for Information Retrieval*

Table 88: Papers presented at the workshop.

- **Leveraging Existing Capabilities.** Word error rates (WER) for planned speech (e.g., by news announcers) in studio conditions are nowadays around 10%, whereas for conversational speech, error rates are still often as high as 30 or 40%. Variations across recordings are, however, often far greater than variations across words: it is therefore often more reasonable from an IR perspective to ask what fraction of the content can be processed well enough to support specific tasks. Supervised machine learning techniques for topic segmentation, for example, place a greater premium on consistency than on raw accuracy, and "bag of words" retrieval techniques are robust in the presence of occasional errors. Extractive summarization, by contrast, requires that consecutive words be correctly recognized (so higher error rates may yield shorter and less informative snippets), and more sophisticated analysis (e.g., the entity tagging used in question answering systems) may be even more sensitive to recognition errors. As one of our panelists observed many years ago (in a machine translation context [Church and Hovy, 1993]), we already have some "good applications for crummy speech recognition." Those opportunities deserve our attention, even as speech researchers work to further improve their techniques.

- **Getting Beyond the Laboratory.** As is often the case early in the technology life cycle, leading-edge speech technology has relied on carefully controlled benchmark evaluations to stimulate and evaluate progress. One consequence of this is that robustness to training-test mismatch is well understood as an important issue, but it remains an under-researched problem. Scalability is recognized as another important challenge, but present speech processing techniques are in general quite resource-intensive. Information retrieval research, by contrast, often emphasizes both robustness and scalability. There is therefore significant potential for synergy,

with speech research bringing us new capabilities that we can productively use, and our experience bringing new application contexts that can help to drive speech research in important directions.

- **Operational Employment.** Questions about what technologies we can build are an important first step, but our long experience with users of our technology allow us to bring another important set of questions to the table. Indexing workflows often contain specialized resources (e.g., topic inventories for use with text classification systems), and the "digital library" researchers with whom we work often pay particular attention to how those resources will be created. Selecting and preparing domain-specific training data for speech recognition would be one example of a similar task in the context of speech processing. Can we foster the development of a new generation of tools that leverage the participation of domain experts in such tasks? The collections people work with in the real world are often quite diverse; can we provide ways for managers of such collections to use some of their materials (e.g., e-text) to improve access to others (e.g., by allowing large scale adaptation of language models in the field rather than in the laboratory)? And do we have anything to say to the people who are initially creating spoken word materials; for example, are there simple techniques (e.g., speaker enrollment for talk show hosts) that might dramatically improve access in some applications if the search technology could be designed to optimally leverage the resulting improvements?

Ultimately, information retrieval research brings two things to the table: real collections, and real users. The recent progress on processing spontaneous conversational speech serves a complementary role, bringing us new types of collections, and hence new types of research questions. Together, it seems that we're a good match!

12 Automatic Video Editing

12.1 Introduction

In this section we address a problem that occurs in two different scenarios: Video-conferences [Sabri and Prasada, 1985] and meetings in a smart room [Moore, 2002].

In a video-conference the participants are in different locations. Each participant is recorded with a camera and a microphone. This audio-visual data is then transmitted to all other participants. Usually the audio stream is preprocessed such that only the active speaker is indeed played. This process is similar to phone conferences (see e. g. *Skype* as a non- and *Spiderphone* as a commercial version). The video channel is different: Current versions either show the active speaker and therefore simply reuse the audio information; or they show a selection or all participants of the meeting at the same time by scaling down the individual video streams until all persons fit on the display (see e. g. *InterCall*'s *InView* solution, or *Visual Nexus*). Neither approach is a good solution: Showing all participants is limited to a few participants. With an increasing number the individual videos get to small. The second approach of simply showing the video of the active speaker is straight-forward and reduces the video size problem. But by doing that the video has only limited extra information: Imagine someone gives a presentation. As he is the only person speaking, he will always be shown. This way you loose the very important information, that the project manager is shaking his head constantly, indicating he is not satisfied with the idea.

Meetings are truly multi-modal in nature [Al-Hames et al., 2006c], thus it can be very important to show persons who currently do not speak. Professional directors of talk-shows follow this rule and from time to time show facial reactions or gestures of the participants. Thus a good video-conference system should neither show all participants at the same time, nor simply show the speaker, but choose one of the participants based on both the audio information, as well as visual information.

In the second scenario all participants are located in the same room and the meeting is recorded with multiple cameras and microphones. Such smart meeting rooms become increasingly important, as the recordings allow to analyse the meeting content, as well as a later comprehension of the decisions [Waibel et al., 2004, Al-Hames et al., 2006b]. Then the recordings together with some high level information can be watched in a meeting browser [Wellner et al., 2004]. However it is usually not possible to simply view all recorded video streams at the same time; thus it is necessary to select one camera and show this stream to the user. Of course this view will in general change within the course of the meeting.

Thus, while video-conferences and local meetings are sociologically quite different; the problem of selection a camera is the same for both scenarios: for each time instance (generally frames) of the meeting we need to select one camera or – as we refer them to – video mode that shows best what happens in the meeting. Generally a mode is a camera view, but could also be a slide or two merged videos (see Sec. 12.3). This mode is then transmitted to the other participants or stored for browsing. The problem can therefore be described as an automatic, virtual meeting director. While the task is commercially very interesting, it has not yet been deeply researched. Previous works suggest video

editing rules for the camera decision [Sumec, 2004, Al-Hames et al., 2006d]. In [Liu and Kimber, 2003] a controllable camera is used and the view is automatically learned. [Liu et al., 2005a] proposes a system to extract relevant meeting regions from wide screen cameras. A user study with expert camera operators [Uchihashi, 2001] offers suggestions how to design an interface. For video surveillance, [Snidaro et al., 2003] suggests how to select cameras, but the decision concentrates only on video quality. Thus, the results from these works can not be directly applied to conference scenarios.

AMIDA developed an advanced rule-based system for automatic video editing. This system builds on a video editing algorithm developed within AMI. It uses a set of rules to define shot compositions: Shots are selected according to their importance and aesthetical aspects are taken into account. The system also allows to include virtual cameras or to create video summaries of the meeting. The developed system works in real-time and has been deeply investigated with available meting data from the M4 and the AMI project (Sec. 12.7).

Building on the experience with rule-based systems and especially the problem of increasing complexity with to many cameras and therefore a large set of rules, AMIDA furthermore suggests to formulate the camera selection as a pattern recognition problem [Al-Hames et al., 2007], where each possible video mode is modelled as a pattern class. The problem can then be reduced to classify each frame of the meeting to one of the classes (i.e. video modes). This way we can train machine learning algorithms and use them for the camera selection. We propose a system based on different Hidden Markov Model (HMM) techniques. We extract audio-visual features (Sec. 12.4) from a data set (Sec. 12.2) and use them in an early fusion HMM (Sec. 12.5), as well as in a problem adapted two layer HMM (Sec. 12.5). Finally the proposed methods are evaluated (Sec. 12.6) and compared to the state-of-the-art rule-based approach (Sec. 12.5).

12.2 Meeting Room and Data Set

The data for this work has been collected in the AMI project and is publicly available [Carletta et al., 2005a]. Each meeting has four participants. We use a subset of 24 five minute videos, each with different participants.

All meetings have been recorded in the IDIAP smart meeting room [Moore, 2002]. This room is equipped with a table, a whiteboard, and a projector with a screen. Close-talking audio is recorded with an omni-directional lapel and a headset with condenser microphone for each participant. Far-field recordings are performed with two microphone arrays. Video is recorded with seven static cameras: four cameras record participants closeup views ($C_1 - C_4$). Two cameras record a left (L), resp. right (R) view of the room; each showing two participants and the table in front of them. The last camera (C) captures a total of the room with all four participants, the table, as well as the whiteboard, and the projector screen. A schematic of the meeting room with the camera positions and three sample shots from these cameras are shown in Fig. 46. The closeup recording corresponds to the camera recordings in a video-conference scenario.

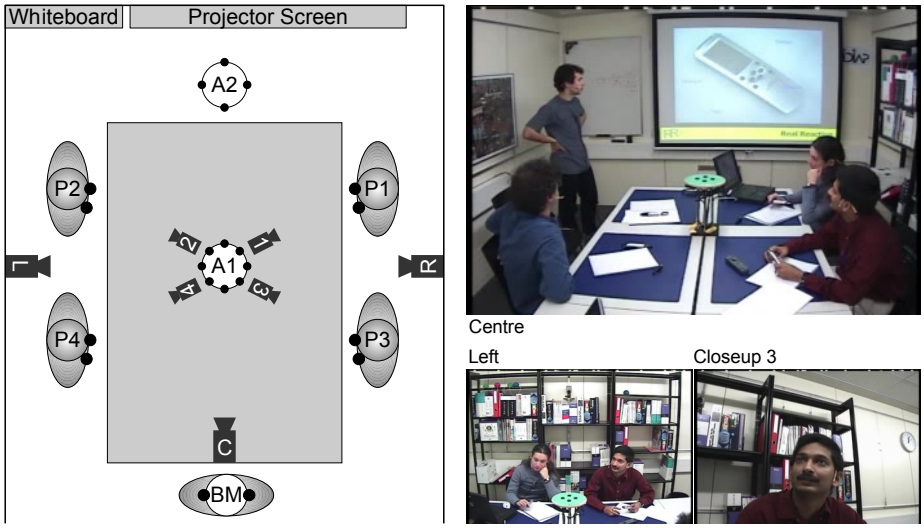


Figure 46: Left: Schematic of the meeting room (not drawn to scale) with the position of the seven cameras. Right: Sample shots from the data set: centre (C) view of the room, shot from the left (L), and a closeup view of a participant (C_3).

12.3 Video Modes and Annotation

For each frame of the meeting we have to select one camera or one view. We will refer to these possible views as video modes V_k. In the case of a video-conference, each participants camera represents one mode, furthermore slides could be another mode. Thus in a video-conference with four persons we would have five modes.

For browsing a recorded meeting, we use each camera in the meeting room as one possible video mode. Based on the available seven cameras in the meeting room and the possible user requirement we defined seven different video modes. They are shown in Fig. 47 and shall be described shortly:

Mode 1 (P1-P4): Shows the closeup camera of one of the persons P1 - P4. This is the main mode when a person is talking or shows facial expressions.

Mode 2: Shows the left-camera view and thus the persons P1 and P3. This mode is ideal for a discussion between the two, or as a diversification if P1 or P3 talks (a stylistic device that human directors often use in talk shows). It can also be used if P1 or P3 talks, and the other one reacts in some way – e.g. a shaking of the head.

Mode 3: Shows the right-camera and thus the persons P2 and P4, it corresponds to mode 2 and has the same properties.

Mode 4: Shows a total of the room from the central camera. This total involves the whiteboard, the projection board, and all four participants. It is ideal if somebody gives a presentation, or to show group interactions. However the individual persons are rather small in this mode. Furthermore the persons are shown from the side, thus details get lost.

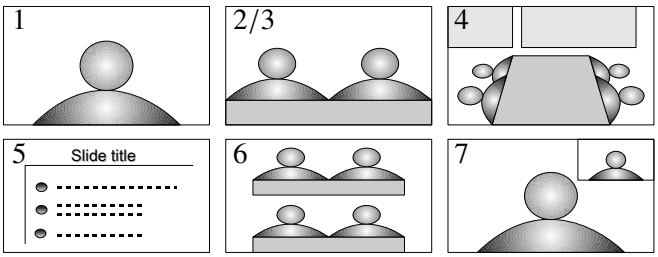


Figure 47: Possible video modes for the smart meeting room scenario. Mode 1 shows a closeup of one of the persons. Mode 2 and 3 show the left, resp. right view of the room. Mode 4 shows a total of the room. Mode 5 shows the current slide from the projector screen. Mode 6 combines both the left and the right camera view. Mode 7 shows one closeup and inserts another closeup into the corner (correspondent view).

Mode 5: This mode inserts a still image (slides, pictures, etc.) into the video. It is ideal to show up the presentation slides when they are changed.

Mode 6: Shows both the output of the left and the right camera. They are slightly cut on top and the bottom, scaled down, and then merged on top of each other. This mode shows all participants in a frontal view and is therefore good for group discussions, note-taking, or group interactions. The individual persons are larger and better shown as in mode 4, but due to the adding up of two views smaller than in mode 1, 2, and 3. Thus individual reactions are less impressive. Furthermore the cutting contains the risk of cutting out heads or hands.

Mode 7 (P1-P4, P1-P4): Shows the closeup camera of one of the persons P1 - P4. A further closeup of another person is merged into the corner. This view can be used to show reactions of one participant, while another person is talking. However if the persons are sitting next to each other, mode 2 or 3 are preferred, as mode 7 is rather unnatural.

The proposed method is not limited to these modes. New ones can easily be added by defining the new mode and simply train a new class without influencing the existing modes. This way the system can easily be adapted to various needs and applications without changing the underlying system. For a further extensive discussion on possible video modes in meetings see [Al-Hames et al., 2006d].

To apply our pattern recognition approach we needed training data for the video mode classes. We therefore set up a limited set of annotation rules, ensuring some basic guidelines: Mainly preventing annotators from very fast switches between the cameras (we encouraged them to stay for at least 10 seconds on one view). However we gave the annotators the freedom to select cameras they thought would best represent the meeting at a given time. Thus the degree of freedom was rather high. Consequently, first studies showed that inter-annotator agreement on the data set was rather low ($\kappa < 0.5$) and therefore not consistent enough. Further studies showed, that persons where very consistent if they annotated the same meeting more than once. This shows that the annotation and the desired camera view indeed depends on the taste of the annotator, but then represents a consistent selection. We therefore decided to use only two annotators, to ensure a consistent training data set.

12.4 Features

Global Motions: As first feature we use global motions (GM). They are simple, but have been successfully applied to various meeting tasks [Wallhoff et al., 2004] and can be calculated in real-time with a latency of only one frame. We split the room into six locations L. Each of the four closeup cameras represents one location. From the centre view camera we extract the projection board and the whiteboard location. Then a difference image sequence $I_d^L(x, y, t)$ is calculated for each of these six locations and each frame t by subtracting the pixel values of two subsequent frames from the video stream. Then the centre of motion is calculated for the x- and y-direction:

$$m_x^L(t) = \frac{\sum_{(x,y)} x \cdot |I_d^L(x, y, t)|}{\sum_{(x,y)} |I_d^L(x, y, t)|}, \quad m_y^L(t) = \frac{\sum_{(x,y)} y \cdot |I_d^L(x, y, t)|}{\sum_{(x,y)} |I_d^L(x, y, t)|} \tag{33}$$

The changes in motion are used to express the dynamics of the movements:

$$\Delta m_x^L(t) = m_x^L(t) - m_x^L(t-1), \quad \Delta m_y^L(t) = m_y^L(t) - m_y^L(t-1) \tag{34}$$

Furthermore the mean absolute deviation of the pixels relative to the centre of motion is computed:

$$\sigma_x^L(t) = \frac{\sum_{(x,y)} |I_d^L(x, y, t)| \cdot (x - m_x^L(t))}{\sum_{(x,y)} |I_d^L(x, y, t)|}$$

and

$$\sigma_y^L(t) = \frac{\sum_{(x,y)} |I_d^L(x, y, t)| \cdot (y - m_y^L(t))}{\sum_{(x,y)} |I_d^L(x, y, t)|} \tag{35}$$

Finally the intensity of motion is calculated from the average absolute value of the motion distribution:

$$i^L(t) = \frac{\sum_{(x,y)} |I_d^L(x, y, t)|}{\sum_{x,y} 1} \tag{36}$$

These seven features are concatenated for each time step in the location dependent motion vector

$$\vec{x}^L(t) = [m_x^L, m_y^L, \Delta m_x^L, \Delta m_y^L, \sigma_x^L, \sigma_y^L, i^L]^T. \tag{37}$$

With this motion vector the high dimensional video stream is reduced to a seven dimensional vector, but it preserves the major characteristics of the currently observed motion. Graphically the motion can be interpreted as an ellipse with a the centre of motion, the mean absolute deviation as the axes and the intensity as the size of the ellipse. The GMs for head and hand movements in the left camera view are shown in Fig. 48 (left).

Concatenating the motion vectors from each of the six positions $\vec{x}^L(t)$ leads to the final motion vector

$$\vec{x}_V(t) = [\vec{x}^{C_1}, \vec{x}^{C_2}, \vec{x}^{C_3}, \vec{x}^{C_4}, \vec{x}^W, \vec{x}^P]^T, \tag{38}$$

that describes the overall motion in the meeting room with 42 features.

Figure 48: Left: Graphical interpretation of the global motion features. The motion in a given region of the image can be interpreted as an ellipse. This is shown for hand and head movements. Right: Detected skin blobs.

Hand and Head Movements: A further way to access the participants activities are hand and head movements. In [Potucek et al., 2004] it was shown how skin blobs can be used to detect the activity of individual meeting participants. We therefore add skin blobs (SB) as a visual feature.

We extract the head and hand SBs with a skin colour look up table. The RGB-images are transformed into the rg-space. Each pixel is then compared to a 16 bit rg-look up table, which results in a binary image, where each possible skin pixel is marked. To fill gaps in skin areas, a 5x5 dilation filter is applied. The found skin areas are then analysed for their shape, the relation of their eigenvalues, and context knowledge about possible positions. Finally subsequent images are averaged with a recursive approach, that is applied individually to blobs in the meeting videos

$$\vec{m}(t) = 1 - \frac{1}{T}\vec{m}(t-1) + \frac{1}{T}\vec{x}(t), \tag{39}$$

where $\vec{x}(t)$ is the current measured value, $\vec{m}(t)$ is the resulting averages vector for the blob position, $\vec{m}(t-1)$ the position in the last image, and T a constant that determines the relation between previous frames and the current measurement. The position and movement of each participant's blobs are concatenated in the final SB motion vector $\vec{x}_{\mathrm{SB}}(t)$. Examples of detected hand and head-blobs in the left camera view are shown in Fig. 48 (right). This approach is simple but can be performed in real-time; more details can be found in [Al-Hames et al., 2006d].

Acoustic Features: From each participant's lapel microphone we extract 12 Mel frequency cepstral coefficients (MFCC) and the energy, as well as the first and second derivations. This results in a 39 dimensional acoustic feature vector $\vec{x}_{\mathrm{MFCC}}(t)$ for each participant.

12.5 Video Mode Selection Models

State-of-the-Art Rule-Based Model For comparison we summarise the state-of-the-art rule-based approach (for details see e.g. [Al-Hames et al., 2006d]). In the following let

t denote the current time step, W the window size, $P \in \{P_1, P_2, P_3, P_4\}$ one of the meeting participants, and $E^P(t)$ the audio energy for person P at time t. The windowed output of the feature is denoted as $D^P(t)$ and derived by summing up the energy in the window:

$$D^P(t) = \sum_{\tau=t-W}^{t} E^P(\tau) \tag{40}$$

The output $D^P(t)$ therefore represents what has recently happened in the audio channel of person P. For each time step t, the rule-based systems then chooses the "most active" person with

$$k(t) = \underset{P}{\operatorname{argmax}}\, D^P(t) \tag{41}$$

Depending on the desired output, this decision $k(t)$ is now directly mapped to one of the video modes $V_k(t)$ (e.g. an activity of person two will of course show the mode corresponding to camera two). This process does not optimise the features, nor does it model interactions between the features, it simply uses the energy. Yet, it is reliable and the behaviour well controlled, thus it is widely applied.

Hidden Markov Model We search for a sequence of camera views from the meeting. As we formulated this video selection as a pattern recognition problem and provided data with annotated video modes, we can apply the Hidden Markov Model (HMM) [Rabiner, 1989]. It can be used for classification of feature streams. In combination with the Viterbi algorithm [Viterbi, 1977] it also segments the stream into a sequence of video modes.

For the recognition with HMMs, each video mode is modelled by one HMM. Each HMM k (and thus each video mode) is represented by a set of parameters $\lambda_k = (\mathbf{A}, \mathbf{B}, \vec{\pi})$, where $\mathbf{A}$ denotes the transition matrix, $\vec{\pi}$ the initial state distribution, and $\mathbf{B}$ is the output distribution, here modelled with mixtures of Gaussians.

For the HMMS, we can use only audio ($\vec{x}_{\text{MFCC}}$), visual ($\vec{x}_{\text{GM}}$ and/or $\vec{x}_{\text{SB}}$), or all features. The selection of the video-mode should be based on both the acoustic and the visual information. Thus we use an early fusion HMM: The frame rates of the streams are adjusted and then concatenated into one multi-modal feature stream $\vec{x}$.

Given this multi-modal training data $\mathbf{X}_k$ from our data set for mode k, the parameters λ_k of the HMM k can be trained with the well known EM-algorithm [Dempster et al., 1977]. The aim of this training is to maximise $p(\mathbf{X}_k|\lambda_k)$. For the training of this HMM k only representatives of the video mode k are used. The resulting models are therefore independent from each other. The HMM corresponding to the centre view is only trained with representatives of this mode. This HMM neither takes the number of classes into account, nor does it know other modes. Thus the system can easily be expanded with new modes: The other – already trained – HMMs are not influenced. One simply needs to train a new HMM for each new video mode, this makes the approach very flexible and easily adaptable.

Once an HMM for each class (i.e. video mode) is trained, the unknown video feature stream $\vec{x}$ is presented to all HMMs λ_k and we select the model k with

$$k = \underset{i}{\operatorname{argmax}}\, p(\vec{x}|\lambda_i) \tag{42}$$

the highest likelihood. This is done with an online version of the Viterbi algorithm [Viterbi, 1977], which can also perform a segmentation of the streamed input vector $\vec{x}$. This way, the feature stream of the meeting is automatically segmented into a sequence of video modes: the desired sequence of camera views from the meeting.

Two-layer Hidden Markov Model Compared to the rule-based approach, the early fusion HMM reacts on both visual and acoustic information and implicitly models the relation between the streams. However, the virtual director should react on the individual actions. Mainly it should stay on the speaker, but if somebody reacts, the system should switch to this person. If the training data represents this behaviour, we can assume that the early fusion HMM learns and therefore models this behaviour.

On the other hand we can explicitely model this with a two-layer HMM: the first layer recognises the individual actions of each participant. These recognised actions together with group related features (e.g. the motion in front of the whiteboard) are then used as input for the second layer that decodes the actual video mode.

For the person HMM layer we defined 14 important individual actions: e.g. standing up or sitting down, but also more subtle actions like nodding or shaking of the head. We use the actions of all four participants in the meeting to train the models, i.e we have a person independent training. Thus we effectively have four times the training data available. The second layer is then trained analogous to the early fusion HMM. However we extend the early fusion feature vector $\vec{x}$ with the person actions: we add the action of each participant in a coded way for each frame of the meeting resulting in the extended feature vector $\vec{x}^e$. This way the video mode HMM explicitely learns the relation between person actions and desired video mode output, but preserves the implicit learning of feature relations. The complete training procedure can then be summarised in algorithm 1.

Algorithm 1 Two-Layer HMM Training

Require: Training feature vectors $\mathbf{X}$
  for all person actions A_j **do**
    $\lambda_{A_j} \leftarrow$ train person action HMM, s.t. max $P(\mathbf{X}_{A_j}|\lambda_{A_j})$
  end for
  $\mathbf{X}^e \leftarrow$ extend the features $\mathbf{X}$ with the true person actions a_i
  for all videomodes V_k **do**
    $\lambda_{V_k} \leftarrow$ train video mode HMM, s.t. max $P(\mathbf{X}^e_{V_k}|\lambda_{V_k})$
  end for

In the recognition phase we apply a two-fold decoding: First the unknown feature stream $\vec{x}$ is used to classify the actions of each person in the meeting. Then the feature vector $\vec{x}$ is extended with the found person actions, resulting in the extended stream $\vec{x}^e$. This feature stream now explicitely comprehends the found individual actions. Finally $\vec{x}^e$ is used to segment and classify the video mode in the second layer. This way the video mode HMM has explicit information about the person actions, however they are of course afflicted with some uncertainty (note the difference to the training, where the true actions are available). While the process separates the individual actions from the video mode, it introduces some latency: The first layer first has to decode the feature streams, and then this output

is fed into the second layer, thus the second layer is always a couple of frames behind. The overall decoding can be summarised in algorithm 2

Algorithm 2 Two-Layer HMM Decoding

Require: Unknown feature vector stream $\vec{x}$
  for all persons P_i in the meeting **do**
    $a_i \leftarrow$ classify individual person action $\text{argmax}_j P(\vec{x}|\lambda_{A_j})$
  end for
  $\vec{x}^* \leftarrow$ extend the stream $\vec{x}$ with the found person actions a_i
  $V \leftarrow$ classify the video mode $\text{argmax}_k P(\vec{x}^*|\lambda_{V_k})$

12.6 Experiments

To evaluate our proposed system we performed two experiments: For the first experiment we assumed the true shot boundaries were known, and the only task was to assign a video mode to each segment. In the second experiment the shot boundaries were unknown and the system had to segment and classify the videos, i.e. the second scenario is the true application. We used the 24 five minute videos from our data set and performed a six-fold cross-validation. We further split the experiments into two scenarios: The first contained seven video modes (all four persons, left, right, and centre camera); the second experiment corresponds to a video-conference with five modes (four persons and the centre camera for presentations).

For the classification we measured the recognition results (RR, i.e. correct found modes; high numbers are better). For the joint segmentation and classification, we measured the frame error rate (FER, i. e. proportion of frames, where a wrong video mode was selected; low numbers are better). All results are shown in Tab. 89.

In the classification task, the rule based system achieves a RR of 45% for seven, resp. 57% for five modes. The proposed multi-modal systems are significantly better: the early fusion HMM achieves 51%, resp. 72% RR. The layered HMM does not outperform the early fusion HMM. A further analysis showed that this is mainly caused by the first action layer (RR only 43%). Thus we also analysed the maximum possible performance of the two-layer HMM by providing the ground truth (GT) individual actions to the second layer.

Modell	**Classification**		**Segmentation**	
	RR-7	*RR-5*	*FER-7*	*FER-5*
Rule Based	45.4%	56.6%	61.4%	53.3%
Early Fusion HMM	51.4%	71.6%	47.9%	27.0%
Two-layer HMM	51.0%	69.6%	45.9%	27.1%
Two-layer HMM (GT)	51.5%	74.2%	42.5%	22.8%

Table 89: Recognition rates (RR, high better) for classification; frame error rates (FER, low better) for segmentation and classification.

Then the two-layer HMM is slightly better than the early fusion HMM. Of course, this GT is not available in a real system.

The tendency of the classification task is even increased in the real application of joint segmentation and classification: Here the rule based approach is highly outperformed by the proposed systems. For the video-conference scenario (five modes), the rule based system selects the wrong video mode for over half the frames (53% FER). Here the early fusion HMM selects the wrong frame in only a quarter of the meeting (27% FER). Thus by applying standard machine learning techniques, we get a much better video.

Interestingly, while the absolute FERs seem quite high, the video output of the system represents a very good view upon the meeting, and only some actions of the participants are missing.

12.7 Real-time rule based system

An alternative system based on rules is developed too. This system uses a video editing algorithm [Stanislav and Igor, 2007] introduced in AMI project. The algorithm uses a set of rules, which define a shot composition. Important events are preferred in the output videos. Shots are selected according to their measure of importance. Aesthetical aspects are also taken into account during shot composition, because some elementary rules from movie makers are included. Cameras with different type of view can be utilized (distant view, close view). Type of the view is considered during the shot composition. Virtual camera tool is available so e.g. persons can be tracked on camera with distant view and satisfactory resolution. The editing can be adjustable to desired information. Viewer can affect editing process; some information can be preferred or suppressed. Various effects can be included in the resulting video e.g. zooming cameras, picture in picture, fade in-/out etc. The developed system can be adapted to various conditions like configuration of cameras, available features, type of processed event like meeting, lecture. Various applications of proposed system are possible. The system can be used for creating of summary video from meetings, which are recorded with several cameras. The most important events e.g. speakers can be shown. The main benefit is an adjustability of results; viewer can specify information that should be preferably presented during video editing process. Shortened version of the meeting video can be produced too. Video conference systems can use proposed technology for saving of network bandwidth, if more than one camera is used at remote participants. Video stream from one camera can be selected at the each time point, and only this stream can be transmitted. Further, video editing can select which one of the remote participants will be displayed on user's screen. Other events like lectures can be processed too, because system is configurable. Lectures are often recorded for e-learning or other purposes. Proposed solution allows acquiring lecture recording using simple setup with one static high-resolution camera. Virtual cameras can be used for tracking of lecturer or capturing projection screen. Output video will be composed from such shots according to lecturer activity, projected slides etc. Demonstration version of the proposed system has been presented at the AMI Community of Interest Workshop[AMI]. This application allows real-time video editing. Shot composition can be affected by the viewer e.g. preference of selected meeting participant can be increased during video editing process. Configurations of system for processing of data

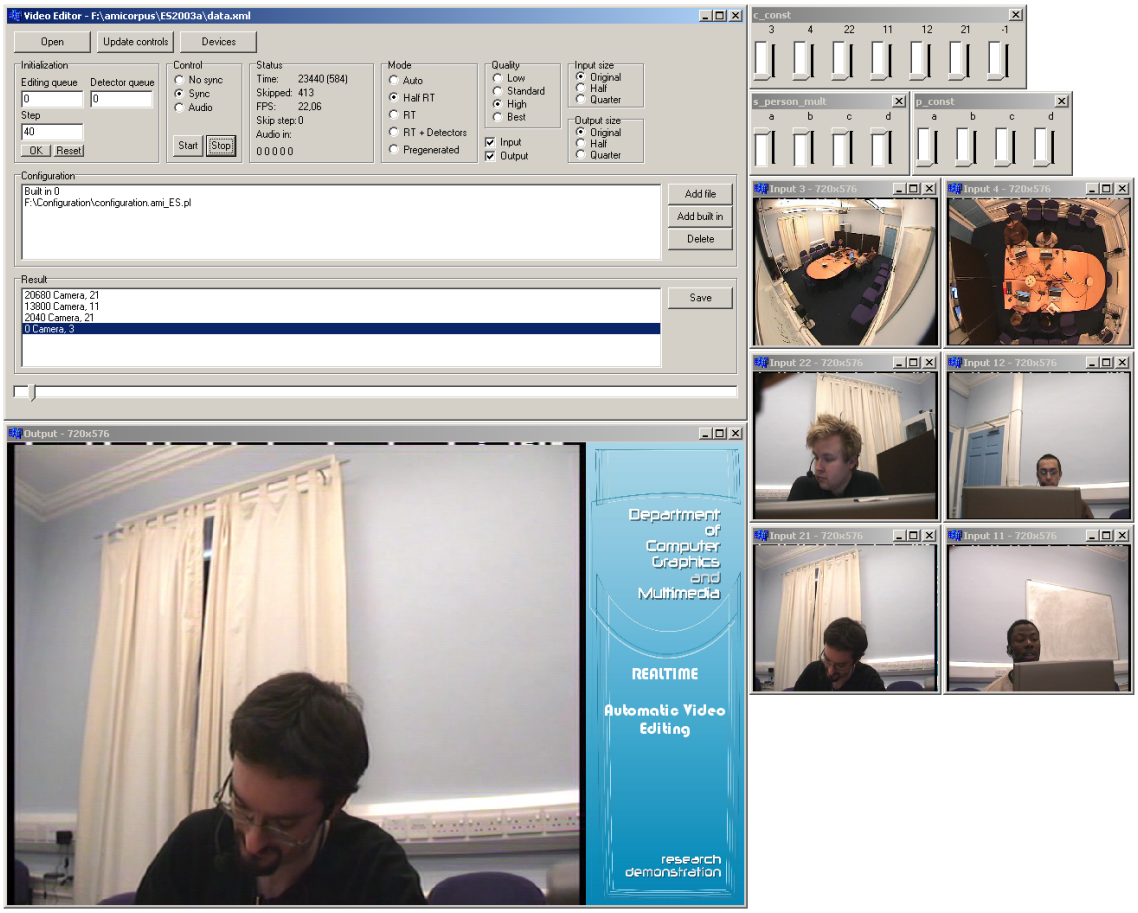


Figure 49: Real-time automatic video editing system

from M4 [M4] and AMI data corpus [ami] have been defined. Video files, audio files, and XML files with selected features (e.g. speaker activity, localization) represent source of the system. In addition, recordings of lectures can be processed too. Some features detectors, which allow tracking of lecturer and recognition of slide changes, have been already integrated. System does not need any additional data in this configuration. Video editing as well as evaluation of features is working real-time.

12.8 Conclusions and Future Work

In this section we proposed a system for selecting a camera view in video-conferences and for browsing recorded meetings. We formulated the task as a pattern recognition problem and could therefore apply Hidden Markov Models for the segmentation of a meeting into a series of camera views. The proposed system is very flexible and can easily be adapted to different applications. Whenever a new view or camera is desired, only one new model has to be trained, without influencing the existing models, or the underlying system.

In an experimental section we showed that the proposed HMM highly outperforms the state-of-the-art rule-based method. While this system always stays on the active speaker, the proposed system changes to other channels, if somebody reacts. This leads to a video that represents the meeting much better. Currently most commercial video-conferencing

systems use DSPs, thus the computational time required for the HMM decoding could easily be performed. Given the good performance of the system, this seems worth the effort.

In the future of AMIDA we will integrate a higher "grammar-level", to prevent fast switches between video modes and retrain the models based on used studies. Furthermore we will evaluate different machine learning techniques to further improve the system performance and use more features derived in WP4.

13 Closure and Future Work

This deliverables reports on the progress during the last 12 months in the ten main areas of work in WP5. These areas address our core ojectives: multimodal structure and content analysis, indexing and retrieval. In all areas, we have applied the components evaluation schemes developed in AMI to ensure measureable quality. In addition, we have designed and implemented an extrinsic evaluation scheme including our components in a meeting browser used for a specific task (sec. 6).

The new results from the last 12 months come on various levels: Some areas are new work in AMIDA (e.g., the work on subjectivity in sec. 3) while others extend work from AMI with new or refined approaches, extend the models to the remote meeting scenario and move towards real-time and on-line algorithms (e.g., sec. 4).

13.1 Future Work

In most areas work will concentrate on refining the algorithms and feature sets, improving robustness to work on automatically generated features (in particular ASR output instead of manual transcript) and improving real-time and on-line capabilities. Also, we will package the results of WP5 so that they can be integrated into meeting assistants and browsers as components. In return, this will allow for further extrinsic evaluations of the work in WP5.

Also, in the next 18 months we will begin work on a few new areas that add to the objectives of WP5.

13.1.1 Disfluencies

We will develop machine learning-based algorithms that also include rule-based approaches for the automatic detection and removal of disfluencies. This will support dialog act classification and segmentation as well as the application of semantic parsing for propositional content analysis.

13.1.2 Paraphrases

We have already developed a preliminary version of Mutaphrase, a system that generates paraphrases of input sentences. The algorithm generates a large number of paraphrases with a wide range of syntactic and semantic distances from the input. For example, given the input ”I like eating cheese”, the system outputs the syntactically distant ”Eating cheese is liked by me”, the semantically distant ”I fear sipping juice”, and thousands of other sentences. The wide range of generated paraphrases makes the algorithm ideal for a range of statistical machine learning problems (such as language modeling), as well as other semantics-dependent tasks such as query, language generation, summarization, etc.

Currently, the Mutaphraser requires that the input sentences be annotated with frame semantic markup. We plan to integrate automatic frame parsers (developed elsewhere) with the Mutaphraser, thereby enabling fully automatic generation of mutaphrases. Also, we

plan to dramatically reduce the memory and processing requirements of the current implementation of the Mutaphraser. Finally, we plan to use the combination of automatic frame parsing and mutaphrasing to enhance the language models for meetings.

13.1.3 Medical Domain

CSIRO is focussing on applications for meetings in the health domain. To progress towards this, work in the first year has implemented Natural Language Processing and Statistical Machine Learning tools for automatically extracting medical concepts from general free-text medical documents. Work to date has developed a UMLS (Unified Medical Language System) plug-in for the GATE text engineering toolkit to enable extraction of SNOMED CT concepts from free-text medical reports.

Ongoing work in the remainder of AMIDA will investigate methods for using medical terminologies and ontologies to better structure and summarise the content of medical text and transcripts of clinical team discussions (CSIRO plans to record some health meeting audio data later in the project for this purpose). Research will also study whether augmenting text feature representations with terminology concept ids improves performance for text classification tasks.

References

http://www.issco.unige.ch/staff/mariag/tools/evalMeasure.html.

http://www.ics.purdue.edu/~ymao/lowbow.htm.

M. Al-Hames, A. Dielmann, D. GaticaPerez, S. Reiter, S. Renals, and D. Zhang. Multimodal integration for meeting group action segmentation and recognition. In *Proc. of MLMI 2005*, 2005.

M. Al-Hames, T. Hain, J. Cernocky, S. Schreiber, M. Poel, R. Muller, S. Marcel, D. van Leeuwen, J. Odobez, S. Ba, H. Bourlard, F. Cardinaux, D. Gatica-Perez, A. Janin, P. Motlicek, S. Reiter, S. Renals, J. van Rest, R. Rienks, G. Rigoll, K. Smith, A. Thean, and P. Zemcik. Audio-visual processing in meetings: Seven questions and current AMI answers. In *Proceedings of Workshop on Machine Learning for Multimodal Interaction (MLMI)*, Washington, DC, USA, May 2006a.

M. Al-Hames, B. Hoernler, R. Mueller, J. Schenk, and G. Rigoll. Automatic multi-modal meeting camera selection for video-conferences and meeting browsing. In *Proceedings of the 8th International Conference on Multimedia and Expo (ICME)*, Beijing, China, July 2007.

M. Al-Hames et al. Audio-visual processing in meetings: Seven questions and current AMI answers. In *P. MLMI*, 2006b.

M. Al-Hames et al. Multimodal integration for meeting group action segmentation and recognition. In *P. MLMI*, 2006c.

M. Al-Hames et al. Using audio, visual, and lexical features in a multi-modal virtual meeting director. In *P. MLMI*, 2006d.

J. Alexandersson, R. op den Akker, J. Baan, T. Becker, J. Černocký, M. Fapšo, A. Dielmann, Y. Gotoh, D. Heylen, S. Hsueh, P. Huisman, N. Jovanovic, T. Kleinbauer, W. Kraaij, S. Lesch, I. McCowan, J. Moore, B. Peskin, S. Raaijmakers, D. Reidsma, S. Renals, R. Rienks, I. Szöke, A. Thean, J. van Rest, D. Verbree, A. Viciarelli, and W. Xu. Augmented Multiparty Interaction D5.2 Implementation and Evaluation Results. Technical report, Brno University of Technology, DFKI, ICSI, IDIAP, TNO, University of Edinburgh, University of Twente and University of Sheffield, 2006.

J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

AMI. Ami community of interest workshop, 2007. http://www.amiproject.org/business-portal/ami-community-of-interest-workshop.

T. Andernach. A machine learning approach to the classification of dialogue utterances. *Computing Research Repository*, july, 1996.

Elisabeth André. *Ein planbasierter Ansatz zur Generierung multimedialer Präsentationen*. Infix, St. Augustin, 1995.

Elisabeth André, Thomas Rist, and Jochen Müller. Wip/ppp: automatic generation of personalized multimedia presentations. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 407–408, Boston, MA, 1997.

Alina Andreevskaia, Sabine Bergler, and Monica Urseanu. All blogs are not made equal, exploring genre differences in sentiment tagging of blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM-2007) Boulder, CO, March 26-28*, pages 195–198, 2007.

J. Ang, Y. Liu, and E. Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. In *IEEE International Conference on Acoustics Speech and Signal Processing*, Philadelphia PA USA, 2005a.

Jeremy Ang, Yang Liu, and Elizabeth Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings of 30th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, PA, USA, 2005b.

B. Arons. Speechskimmer: a system for interactively skimming recorded speech. *ACM Trans. Comput.-Hum. Interact.*, 4(1):3–38, 1997. ISSN 1073-0516. doi: http://doi.acm.org/10.1145/244754.244758.

J. L. Austin. *How to do Things with Words*. Oxford: Clarendon Press, 1962.

G. M. Ayers. Discourse functions of pitch range in spontaneous and read speech. In Jennifer J. Venditti, editor, *OSU Working Papers in Linguistics*, volume 44, pages 1–49, 1994.

M. Balakrishna, D. Moldovan, and E.K. Cave. N-best list reranking using higher level phonetic, lexical, syntactic and semantic knowledge sources. In *Proc. IEEE ICASSP*, volume 1, pages 413–416, May 2006.

R. F. Bales. *Interaction Process Analysis: A method for the study of small groups*. Cambridge: Addison-Wesley, 1950.

R.F. Bales, F.L. Strodbeck, T.M. Mills, and M.E. Roseborough. Channels of communication in small groups. *American Sociological Review*, 16:461–468, 1951.

S. Banerjee and A. Rudnicky. Segmenting meetings into agenda items by extracting implicit supervision from human note-taking. In *Proc. of IUI 2006*, 2006.

S. Banerjee, C. Rose, and A. I Rudnicky. The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In *Proceedings of the 10th International Conference on Human Computer Interaction*, Rome, Italy, 2005.

D. Baron, E. Shriberg, and A. Stolcke. Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. In *ICSLP*, Denver, Colorado, USA, September 2002.

D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34:177–210, 1999.

Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Automatic extraction of opinion propositions and their holders. In *Working Notes — Exploring Attitude and Affect in Text: Theories and Applications (AAAI Spring Symposium Series)*, 2004.

J. Bilmes and K. Kirchhoff. Factored language models and generalized parallel backoff. *Proceedings of HLT/NAACL 2003*, May 2003.

J. Bilmes and G. Zweig. The Graphical Model ToolKit: an open source software system for speech and time-series processing. *Proc. IEEE ICASSP*, Jun. 2002.

J.A. Bilmes. Dynamic bayesian multinets. *Proc. Int. Conf. on Uncertainty in Artificial Intelligence*, 2000.

D. M. Blei and P. J. Moreno. Topic segmentation with an aspect hidden Markov model. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 2001.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

Constantinos Boulis and Mari Ostendorf. A quantitative analysis of lexical differences between genders in telephone conversation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. ACM Press, 2005.

Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, Hyderabad, India, 2007.

L. Breiman. Bias, variance, and arcing classifiers, 1996.

G. Brown, K. L. Currie, and J. Kenworthe. *Questions of Intonation*. University Park Press, 1980.

L. Burget, J. Cernocky, M. Fapso, M. Karafiat, P. Matejka, P. Schwarz, P. Smrz, and I. Szoke. Indexing and search methods for spoken documents. In *Proceedings of the Ninth International Conference on Text, Speech and Dialogue, TSD 2006*, pages 351–358, Brno, Czech Republic, September 2006.

CALO. Cognitive agent that learns and organizes. *http* : *//www.ai.sri.com/project/CALO*, 2006. URL \texttt{\$http://www.ai.sri.com/project/CALO\$}.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The ami meetings corpus. In *Proceedings of the Measuring Behavior 2005 symposium on "Annotating and measuring Meeting Behavior"*, 2005a. AMI-108.

J. Carletta, S. Evert, U. Heid, and J. Kilgour. The NITE XML Toolkit: Data Model and Query Language. *Language Resources and Evaluation Journal*, 39(4):313–334, 2005b.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus: A pre-announcement. *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–05)*, 2006.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`.

F.Y.Y. Choi, P. Wiemer-Hastings, and J. D. Moore. Latent semantic analysis for text segmentation. In Lillian Lee and Donna Harman, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 109–117, 2001.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 355–362, Vancouver, Canada, 2005.

Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, pages 431–439, Sydney, Australia, 2006.

H. Christensen, B. Kolluru, Y. Gotoh, and S. Renals. Maximum entropy segmentation of broadcast news. In *Proc. of ICASP*, Philadelphia USA, 2005.

K.W Church and E.H. Hovy. Good applications for crummy machine translation. *Machine Translation*, 8(4):239–258, 1993.

C. Cieri, D. Miller, and K. Walker. Research methodologies, observations and outcomes in conversational speech data collection. In *Proceedings of the Human Language Technologies Conference (HLT)*, 2002.

M. Collins and T. Koo. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70, 2005.

F.M.G. de Jong, D.W. Oard, R. Ordelman, and S. Raaijmakers (eds.), editors. *Proceedings of ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*. 2007. ISBN=978-90-365-2542-8.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1), 1977.

A. Dielmann and S. Renals. Multistream recognition of dialogue acts in meetings. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–06)*, pages 178–189. Springer, 2007a.

A. Dielmann and S. Renals. DBN based joint dialogue act recognition of multiparty meetings. In *Proc. IEEE ICASSP*, volume 4, pages 133–136, April 2007b.

A. Dielmann and S. Renals. Automatic dialogue act recognition using a dynamic Bayesian network. In S. Renals, S. Bengio, and J. Fiscus, editors, *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–06)*, pages 178–189. Springer, 2007c.

Alfred Dielmann and Steve Renals. Automatic meeting segmentation using dynamic bayesian networks. *IEEE Transactions on Multimedia*, 9(1):25–36, 2007d.

B. Dorr, C. Monz, S. President, R. Schwartz, and D. Zajic. A methodology for extrinsic evaluation of text summarization: Does rouge correlate? In *ACL 2005, MTSE Workshop, Ann Arbor, USA*, pages 1–8, 2005.

Chris Drummond and Robert C. Holte. What roc curves can't do (and cost curves can). In *ROCAI*, pages 19–26, 2004.

Chris Drummond and Robert C. Holte. Cost curves: An improved method for visualizing classifier performance. *Mach. Learn.*, 65(1):95–130, 2006. ISSN 0885-6125. doi: http://dx.doi.org/10.1007/s10994-006-8199-5.

K. Duh and K. Kirchhoff. Automatic learning of language model structure. In *Proc. International Conference on Computational Linguistics (COLING)*, November 2004.

C. Edelsky. Who's got the floor? *Language in Society*, 10:383–421, 1981.

Ralf Engel. SPIN: A semantic parser for spoken dialog systems. In *Proceedings of the Fifth Slovenian And First International Language Technology Conference (IS-LTC 2006)*, 2006.

M. Eskenazi, A. Rudnicky, K. Gregory, P. Constantinides, R. Brennan, C. Bennett, and J. Allen. Data collection and processing in the carnegie mellon communicator. In *Proceedings of Eurospeech*, 1999.

Andrea Esuli and Fabrizio Sebastiani. Determining term subjectivity and term orientation for opinion mining. In *Proceedings the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 193–200, Trento, IT, 2006. doi: http://acl.ldc.upenn.edu/E/E06/E06-1025.pdf.

Andrea Esuli and Fabrizio Sebastiani. Determining the semantic orientation of terms through gloss analysis. In *Proceedings of ACM SIGIR Conference on Information and Knowledge Management (CIKM-05)*, pages 617–624, Bremen, Germany, 2005.

Christiane Fellbaum, editor. *WordNet–An Electronic Lexical Database*. MIT Press, 1998.

R. Fernandez and R.W. Picard. Dialog act classification from prosodic features using support vector machines. In *Proceedings of speech prosody 2002*, April 2002.

P.A. Flach and N. Lachiche. Confirmation-guided discovery of first-order rules with tertius. *Machine Learning*, 1(42):61–95, 2002.

J. Fung, D. Hakkani-Tur, M. Doss, L. Shriberg, S. Cuendet, and N. Mirghafori. Cross-linguistic analysis of prosodic features for sentence segmentation. Technical report, University of California Berkeley, 2007.

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. Discourse segmentation of multiparty conversation. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 562–569, Sapporo, Japan, 2003.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. Identifying agreement and disagreement in conversational speech: Use of Bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, 2004.

J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees. The trec spoken document retrieval track: A success sotry. In *Proc. RIAO*, 2000.

J. S. Garofolo, C. D. Laprun, M. Michel, V.M. Stanford, and E. Tabassi. The NIST meeting room pilot corpus. In *Proceedings of LREC04*, 2004.

D. Gatica-Perez, I. McCowan, D. Zhang, and S. Bengio. Detecting group interest level in meetings. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

Maria Georgescul, Alexander Clark, and Susan Armstrong. An analysis of quantitative aspects in the evaluation of thematic segmentation algorithms. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 144–151, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W06/W06-1320`.

David R. Gibson. Participation shifts: Order and differentiation in group conversation. *Social forces*, 81(4):1335–1381, 2003.

A. Girgensohn, J. Boreczky, and L. Wilcox. Keyframe-based user interfaces for digital video. *IEEE Computer*, 34(9):61–67, 2001. ISSN 0018-9162. doi: http://dx.doi.org/10.1109/2.947093.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proc. EACL2006*, pages 401–408, 2006.

J. Godfrey, E. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of ICASSP*, pages 517–520, 1992.

J. Goldmann and S. Renals et al. Accessing the spoken word. *International Journal on Digital Libraries*, 5(4):287–298, 2005. ISSN=1432-5012.

Andrew S. Gordon and Kavita Ganesan. Automated story extraction from conversational speech. In *Proceedings of the Third International Conference on Knowledge Capture (K-CAP 05)*, 2005.

K. Greene. The future of search: The head of google research talks about his group's projects. *Technology Review*, 2007. http://www.technologyreview.com/Biztech/19050/.

T. Grenager, D. Klein, and C.g Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of ACL 2005*, 2005.

F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky. Probabilistic and bottle-neck features for lvcsr of meetings. In *Proc. ICASSP 2007*, Hawaii, 2007.

B. Grosz and J. Hirschberg. Some intonational characteristics of discourse structure. In *Proceedings of the International Conference on Spoken Language Processing*, 1992.

B. Grosz and C. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3), 1986.

Sangyun Hahn, Richard Ladner, and Mari Ostendorf. Agreement/disagreement classification: Exploiting unlabeled data using contrast classifiers. In *Proceedings of HLT/-NAACL*, 2006.

T. Hain, J. Dines, G. Garau, M. Karafiat, D. Moore, V. Wan, R. Ordelman, and S. Renals. Transcription of conference room meetings: An investigation. In *Proc. of Interspeech 2005*, 2005.

T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, and V. Wan. The ami meeting transcription system. In *Proc. NIST Rich Transcription 2006 Spring Meeting Recognition Evaluation Worskhop*, Washington D.C., 2006.

T. Hain, L. Burget, J. Dines, G. Garau, V. Wan, M. Karafiat, J. Vepa, and M. Lincoln. The AMI system for the transcription of speech in meetings. In *Proc. IEEE ICASSP*, volume 4, pages 357–360, April 2007.

M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

D. Harman and P. Over. In *Proc. of the DUC 2004, Boston, USA*, 2004.

H. Hastie, M. Poesio, and S. Isard. Automatically predicting dialogue structure using prosodic features. *Speech Communication*, (36):63–79, 2002.

Vasileios Hatzivassiloglou and Kathy McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 174–181, Madrid, Spain, 1997.

M. Hearst. TextTiling: Segmenting text into multiparagraph subtopic passages. *Computational Linguistics*, 25(3):527–571, 1997.

Marti Hearst. Multi-paragraph segmentation of expository text. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 9 – 16, New Mexico State University, Las Cruces, New Mexico, 1994. URL `citeseer.ist.psu.edu/article/hearst94multiparagraph.html`.

D. Heckerman. A tutorial on learning with Bayesian networks. Techical report MSR-TR-95-06, Microsoft Research, March 1995.

D. Hillard, M. Ostendorf, and E. Shriberg. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proc. HLT-NAACL 2003*, 2003a.

Dustin Hillard, Mari Ostendorf, and Elizabeth Shriberg. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proceedings of HLT/NAACL*, 2003b.

J. Hirschberg and D. Litman. Now let's talk about now: identifying cue phrases intonationally. In *Proc. of ACL 1987*, 1987.

J. Hirschberg and C. H. Nakatani. A prosodic analysis of discourse segments in direction-giving monologues. In *Proc. of ACL 1996*, 1996.

J. Hirschberg, M. Bacchiani, D. Hindle, P. Isenhour, A. Rosenberg, L. Stark, L. Stead, S. Whittaker, and G. Zamchick. Scanmail: Browsing and searching speech data by content. In *Proc. of Interspeech 2001, Aalborg, Denmark*, pages 1299–1302, 2001.

L. Hirschman, M. Light, and E. Breck. Deep read: A reading comprehension system. In *Proc. of ACL 1999, College Park, MD, USA*, pages 325–332, 1999. URL `citeseer.ist.psu.edu/article/hirschman99deep.html`.

P. Hsueh. Ami/amida state-of-the-art overview: Recognition of discourse segments in meetings. Technical report, AMI Consortium, 2007.

P. Hsueh and J. Moore. What decisions have you made: Automatic decision detection in conversational speech. In *Proceedings of NACCL/HLT 2007*, 2007a.

P. Hsueh and J. D. Moore. Combining multiple knowledge sources for dialogue segmentation in multimedia archives. In *Proceedings of the 45th Annual Meeting of the ACL*, 2007b.

P. Hsueh and J.D. Moore. Automatic topic segmentation and lablelling in multiparty dialogue. In *the first IEEE/ACM workshop on Spoken Language Technology (SLT) 2006*, 2006.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004 (KDD 2004)*, pages 168–177, Seattle, Washington, 2004.

A. Janin et al. The ICSI meeting corpus. In *Proc. of ICASSP 2003*, 2003.

G. Jefferson. Notes on a possible metric which provides for a "standard maximum" silence of approximately one second in conversation. In D. Roger and P. Bull, editors, *Conversation: an interdisciplinary perspective.*, pages 166–196. Clevedon: Multilingual Matters, 1989.

G. Ji and J. Bilmes. Dialog act tagging using graphical models. *Proc. of the IEEE ICASSP*, March 2005.

H. Jing, R. Barzilay, K. McKeown, and M. Elhadad. Summarization evaluation methods: Experiments and analysis. In *Proc. of the AAAI Symposium on Intelligent Summarization, Stanford, USA*, pages 60–68, 1998.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142, Berlin, 1998. Springer.

N. Jovanovic. *To whom it may concern. Addressee identification in face-to-face meetings*. PhD thesis, University of Twente, Enschede, The Netherlands, March 2007.

D. Jurafsky, E. Shriberg, B. Fox, and T. Curl. Lexical, prosodic, and syntactic cues for dialog acts. In M. Stede, L. Wanner, and E. Hovy, editors, *Discourse Relations and Discourse Markers: Proceedings of the Conference*, pages 114–120. Association for Computational Linguistics, Somerset, New Jersey, 1998.

Min-Yen Kan, Kathleen R. McKeown, and Judith L. Klavans. Applying natural language generation to indicative summarization. In *Proceedings of 8th European Workshop on Natural Language Generation*, pages 92–100, Toulouse, France, July 2001.

Ioannis Kanaris and Efstathios Stamatatos. Webpage genre identification using variable-length character n-grams. In *Proc. ICTAI 2007*, 2007.

Robert E. Kass. The geometry of asymptotic inference. *Statistical Science*, 4(3):188–234, 1989.

S. Katrenko. Textual data categorization: back to the phrase-based representation. In *Proceedings in 2nd International IEEE Conference "Intelligent systems", Vol. III*, pages 64–67, June 2004.

S. Keizer and R. op den Akker. Dialogue act recognition under uncertainty using bayesian networks. *Natural Language Engineering*, 1:1–30, 2005.

Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, May 2006.

Soo-Min Kim and Eduard Hovy. Automatic detection of opinion bearing words and sentences. In *Companion Volume to the Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-2005)*, pages 61–66, Jeju Island, KR, 2005.

Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the Twentieth International Conference on Computational Linguistics (COLING 2004)*, pages 1267–1373, Geneva, Switzerland, 2004.

D. Kimber, L. Wilcox, F. Chen, and T. Moran. Speaker segmentation for browsing recorded audio. In *Proc. of CHI 95, Denver, United States*, pages 212–213, 1995.

T. Kleinbauer, S. Becker, and T. Becker. Combining multiple information layers for the automatic generation of indicative meeting abstracts. In *Proc. of ENLG 2007, Dagstuhl, Germany*, 2007.

Ron Kohavi and David H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 275–283. Morgan Kaufmann, 1996.

J. Kolar, E. Shriberg, and Y. Liu. On speaker-specific prosodic models for automatic dialog act segmentation of multi-party meetings. In *Proc. INTERSPEECH ICSLP 2006*, 2006a.

J. Kolar, E. Shriberg, and Y. Liu. Using prosody for automatic sentence segmentation of multi-party meetings. In *Proc. TSD 2006*, volume 9, pages 629–636, 2006b.

H. Kozima. Text segmentation based on similarity between words. In *Proc. of ACL 1993*, 1993.

W. Kraaij and W. Post. Task based evaluation of exploratory search systems. In *Proc. of SIGIR 2006 Workshop, Evaluation Exploratory Search Systems, Seattle, USA*, pages 24–27, 2006.

Taku Kudo. $http://chasen.org/\ taku/software/CRF++/$, 2006.

W. Kunz and H. W. J. Ritte. Issue as elements of information system. Technical Report Working Paper 131, Institute of Urban and Regional Development Research, University of California, Berkeley, 1970.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. Int. Conference on Machine Learning (ICML)*, June 2001a.

J. Lafferty, F. Pereira, and A. McCallum. Conditional random ?elds: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001b.

John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning*, 6:129–163, 2005.

G. Lebanon. Learning riemannian metrics. In *Proc. of the 19th Conference on Uncertainty in Artificial Intelligence, AUAI Press*, 2003.

G. Lebanon, Y. Mao, and J. Dillon. The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research*, 8(10):2405–2441, 2007.

Guy Lebanon. Sequential document representations and simplicial curves. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 2006.

D. S. Lee, B. Erol, J. Graham, J. J. Hull, and N. Murata. Portable meeting recorder. In *Proceedings of ACM Multimedia 2002*, Juan Les Pins, France, 2002.

I. Lehiste. Phonetic characteristics of discourse. In *the Meeting of the Committee on Speech Research, Acoustical Society of Japan*, 1980.

P. Lendvai and J. Geertzen. Token-based chunking of turn-internal dialogue act sequences. In *Proceedings of the 8th SIGDIAL Workshop on Discourse and Dialogue*, pages 174–181, Antwerp, Belgium, 2007a.

P. Lendvai, A. Van den Bosch, and E. Krahmer. Machine learning for shallow interpretation of user utterances in spoken dialogue systems. In *Proceedings of EACL-03 Workshop on Dialogue Systems: interaction, adaptation and styles of management*, pages 69–78, 2003.

Piroska Lendvai and Jeroen Geertzen. Token-based chunking of turn-internal dialogue act sequences. In *Proceedings SigDial 2007*, 2007b.

G. Levow. Prosody-based topic segmentation for mandarin broadcast news. In *Proc. of HLT 2004*, 2004.

X. Li and D. Roth. Exploring evidence for shallow parsing, 2001.

C. Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out of ACL 2004*, 2004.

C.-Y. Lin and E. H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL 2003, Edmonton, Calgary, Canada*, May 2003.

C.-L. Liu, H. Sako, and H. Fujisawa. Integrated segmentation and recognition of handwritten numerals: Comparison of classification algorithms. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 369–374, Niagra-on-the-Lake, Canada, August 2002.

Q. Liu and D. Kimber. Learning automatic video capture from human's camera operations. In *Proc. ICIP*, 2003.

Q. Liu et al. An online video composition system. In *Proc. ICME*, 2005a.

Y. Liu. Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. In *Proc. Interspeech - ICSLP*, pages 1938–1941, September 2006.

Y. Liu, E. Shriberg, A. Stolcke, and M. Harper. Using machine learning to cope with imbalanced classes in natural sppech: Evidence from sentence boundary and disfluency detection. In *Proceedings of the Intl. Conf. Spoken Language Processing*, 2004.

Y. Liu, E. Shriberg, A. Stolcke, and M. Harper. Comparing hmm, maximum entropy, and conditional random fields for disfluency detection. In *Proceedings of Eurospeech 2005, Lisboa*, 2005b.

M4. M4 corpus, 2007. http://www.idiap.ch/mmm/corpora/m4-corpus.

Igor Malioutov, Alex Park, Regina Barzilay, and James Glass. Making sense of sound:unsupervised topic segmentation over acoustic input. In *Proceedings of ACL 2007*, 2007.

R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of Sixth Conf. on Natural Language Learning, 2002.*, pages 49–55, 2002. URL citeseer.ist.psu.edu/malouf02comparison.html.

I. Mani. Summarization evaluation: An overview. In *Proc. of the NTCIR Workshop 2 Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization, Tokyo, Japan*, pages 77–85, 2001.

I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim. The tipster summac text summarization evaluation. In *Proc. of EACL 1999, Bergen, Norway*, pages 77–85, 1999.

Inderjeet Mani and Mark T. Maybury, editors. *Advances in automatic text summarization*. MIT Press, 1999.

S. Marchand-Mailet. Meeting record modeling for enhanced browsing. Technical report, Computer Vision and Multimedia Lab, Computer Centre, University of Geneva, Switzerland, 2003.

C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Wonderweb deliverable D18 ontology library (final), December 2003.

P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil. Phonotactic language identification using high quality phoneme recognition. In *Proc. Eurospeech2005*, 2005.

I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus. In *Proceedings of the Measuring Behavior 2005 symposium on Annotating and Measuring Meeting Behavior*, Wageningen, NL, September 2005a.

I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(3):305–317, 2005b.

D. Moore. The IDIAP smart meeting room. Research Report 07, IDIAP, 2002.

Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, pages 341–349, Edmonton, Canada, 2002.

Kevin P. Murphy. The Bayes Net toolbox for Matlab. *Computing Science and Statistics*, 33:331–350, 2001.

G. Murray and S. Renals. Term-weighting for summarization of multi-party spoken dialogues. In *Proc. of MLMI 2007, Brno, Czech Republic*, 2007.

G. Murray, S. Renals, and J. Carletta. Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, September 2005.

G. Murray, S. Renals, and M. Taboada. Prosodic correlates of rhetorical relations. In *Proceedings of HLT/NAACL ACTS Workshop*, 2006.

M. Nagata and T. Morimoto. An experimental statistical dialogue model to predict the speech act type of the next utterance. *Proc. of the International Symposium on Spoken Dialogue*, pages 83–86, November 1993.

Daniel Neiberg, Kjell Elenius, and Kornel Laskowski. Emotion recognition in spontaneous speech using GMMs. In *Proceedings of INTERSPEECH*, 2006.

Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *ACL*, pages 611–618, 2006.

J. Niekrasz, M. Purver, J. Dowding, and S. Peters. Ontology-based discourse understanding for a persistent meeting assistant. In *Proc. of the AAAI Spring Symposium*, 2005.

NIST website. Rt-03 fall rich transcription. http://www.nist.gov/speech/tests/rt/rt2003/fall/, 2003.

K. Otsuka, Y. Takemae, J. Yamato, and H. Murase. A probabilistic inference of multiparty-conversation structure based on markov-switching models of gaze patterns, head directions, and utterances. In *Proceedings of International Conference on Multimodal Interfaces (ICMI)*, pages 191–198, Trento, Italy, 2005.

V. Pallotta, J. Niekrasz, and M. Purver. Collaborative and argumentative models of meeting discussions. In *Proceeding of CMNA-05 workshop on Computational Models of Natural Arguments in IJCAI 05*, 2005.

V. Pallotta, V. Seretan, and M. Ailomaa. User requirements analysis for meeting information retrieval based on query elicitation. In *Proceedings of ACL 2007*, 2007.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 79–86, Philadelphia, Pennsylvania, 2002.

A. Park and J. R. Glass. Unsupervised word acquisition from speech using pattern discovery. in . In *Proceedings of ICASSP*, 2006.

R. Passonneau and D. Litman. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proc. of ACL 1993*, 1993.

L. Pevzner and M. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002.

J. Ponte and W. Croft. Text segmentation by topic. In *Proc. of the Conference on Research and Advanced Technology for Digital Libraries 1997*, 1997.

Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 339–346, Vancouver, Canada, 2005.

W.M. Post, A.H. Cremers, and O.B. Henkemans. A research environment for meeting behavior. In A. Nijholt, T. Nishida, R. Fruchter, and D. Rosenberg, editors, *Social Intelligence Design*, Enschede, The Netherlands, 2004.

I. Potucek, S. Sumec, and M. Spanel. Participant activity detection by hands and face movement tracking in the meeting room. In *Proceedings CGI*, 2004.

D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, 2004.

Matthew Purver, Patrick Ehlen, and John Niekrasz. Detecting action items in multi-party meetings: Annotation and initial experiments. In *Proceedings of the 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Washington, DC, USA, 2006a.

Matthew Purver, Konrad Krding, Tom Griffiths, and Josh Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of COLING/ACL 2006*, 2006b.

Stephan Raaijmakers. Sentiment classificatin with interpolated information diffusion kernels. In *Proceedings of the First International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD'07)*, volume 8(10), pages 2405–2441, 2007a.

Stephan Raaijmakers. Language learning with information diffusion kernels. Technical Report TNO ICT, 2007, Available from `http://stephanraaijmakers.files.wordpress.com/2007/05/ce-hyperparam.pdf`, 2007b.

Stephan Raaijmakers. Hyperparameter estimation with an elitist cross-entropy method. Technical Report TNO ICT, 2007. Available from `http://stephanraaijmakers.files.wordpress.com/2007/05/multinomial-languagelearning-v2.pdf`, 2007c.

Stephan Raaijmakers and Wessel Kraaij. A shallow approach to subjectivity classification. In *Submission to ICWSM'08*, 2008.

L.R. Rabiner. A tutorial on HMMs and selected applications in speech recognition. *Proc. of the IEEE*, 77(2), 1989.

Dennis Reidsma and Jean Carletta. Reliability measurement: theres no safe limit. *Journal of Computational Linguistics*, pages 1–6, 2007. submitted.

N. Reithinger and M. Klesen. Dialogue act classification using language models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, 1997.

Norbert Reithinger, Michael Kipp, Ralf Engel, and Jan Alexandersson. Summarizing multilingual spoken negotiation dialogues. In *Proceedings of the 38th Conference of the Association for Computational Linguistics*, pages 310–317, Hong Kong, China, October 2000.

Remco Bouckaert Remco. Bayesian network classifiers in weka, 2007. URL `citeseer.ist.psu.edu/705669.html`.

S. Renals and D. Ellis. Audio information access from meeting rooms. In *Proc. IEEE ICASSP, volume 4*, pages 744–747, 2003.

J. Reynar. *Topic Segmentation: Algorithms and Applications*. PhD thesis, UPenn, PA USA, 1998.

R. Rienks, D. Heylen, and E. van der Weijden. Argument diagramming of meeting conversations. In *Multimodal Multiparty Meeting Processing Workshop at the ICMI*, 2005.

R.J. Rienks, D. Zhang, D. Gatica-Perez, and W. Post. Detection and application of influence rankings in small group meetings. In F. Kwek, editor, *Proceedings of the Eighth International Conference on Multimodal Interfaces (ICMI06)*, pages 257–264, New York, 2006. ACM Press.

Rutger J. Rienks. *Meetings in Smart Environments, Implications of Progressing Technology*. PhD thesis, University of Twente, 2007.

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

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL-2003)*, pages 25–32, Edmonton, Canada, 2003.

Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 440–448, Sydney, Australia, July 2006. Association for Computational Linguistics.

N. C. Romano and J. F Nunamaker. Meeting analysis: findings from research and practice. In *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34)*, 2001.

S. Rosset and L. Lamel. Automatic detection of dialog acts based on multi-level information. In *Proceedings of the International Conference of Speech and Language Processing (ICSLP)*, pages 540–543, Jeju Island, Hawaii, October 2004.

M. Rotaru. Dialog act tagging using memory-based learning. Technical report, University of Pittsburgh, spring 2002. Term project in Dialogue-Systems class.

S. Sabri and B. Prasada. Video conferencing systems. *Proceedings of the IEEE*, 73(4), 1985.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

S. Sarawagi and W.W. Cohen. Semi-markov conditional random ?elds for information extraction. In *NIPS*, 2004.

Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

P. Schwarz, P. Matejka, and J. Cernocky. Towards lower error rates in phoneme recognition. In *Proc. TSD 2004*, Brno, Czech Republic, 2007.

Petr Schwarz, Pavel Matjka, and Jan ernock. Towards lower error rates in phoneme recognition. *Lecture Notes in Computer Science*, (3206):465–472, 2004. ISSN 0302-9743. URL `http://www.fit.vutbr.cz/research/view_pub.php?id=7483`.

John Searle. *Speech Acts*. Cambridge University Press, 1969.

L. Shen, A. Sarkar, and F. Och. Discriminative reranking for machine translation. In *Proc. HLT–NAACL*, pages 177–184, May 2004.

E. Shriberg, R. Bates, P. Taylor, A. Stolcke, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. Van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, (41):439–487, 1998.

E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communications*, 31(1-2):127–254, 2000.

E. Shriberg, A. Stolcke, and D. Baron. Observations on overlap: findings and implications for automatic processing of multi-party conversation. In *Proceedings of7 th European Conference on Speech Communication and Technology (EUROSPEECH)*, Aalborg, Denmark, 2001.

E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. The ICSI meeting recorder dialog act (MRDA) corpus. *Proc. HLT-NAACL SIGDIAL Workshop*, April–May 2004.

E. E. Shriberg. To "errrr" is human: Ecology and acoustics of speech disfluencies. *Journal of the International Phonetic Association*, 31(1):153–169, 2001. Cambridge University Press.

L. Snidaro, R. Niu, P.K. Varshney, and G.L. Foresti. Automatic camera selection and fusion for outdoor surveillance under changing weather conditions. In *Proc. AVSS*, 2003.

Swapna Somasundaran, Josef Ruppenhofer, and Janyce Wiebe. Detecting arguing and sentiment in meetings. In *Proceedings of the 8th ACL SIGdial Workshop on Discourse and Dialogue (SIGdial 2007)*, 2007.

E. Stamatatos. Ensemble-based author identification using character n-grams. In *Proc. of the 3rd Int. Workshop on Text-based Information Retrieval (TIR'06)*, pages 41–46, 2006.

Sumec Stanislav and Potcek Igor. Evaluation of automatic video editing. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI), Brno, CZ*, 2007.

Luite Stegeman, Mannes Poel, and Rieks op den Akker. A support vector machine approach to dutch part-of-speech tagging. In Springer-Verlag, editor, *IDA'07*, 2007.

Stefan Steidl, Michael Levit, Anton Batliner, Elmar Nöth, and Heinrich Niemann. "of all things the measure is man." automatic classification of emotion and intra labeler consistency. In *ICASSP 2005, International Conference on Acoustics, Speech, and Signal Processing*, 2005.

R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. In *Conference on Human Factors in Computing Systems (CHI2002)*, Minneapolis, MI, USA, 2002.

Reiner Stiefelhagen, Jie Yang, and Alex Waibel. Modeling focus of attention for meeting indexing based on multiple cues. *IEEE Transactions on Neural Networks*, 13(4):928–938, 2002.

A. Stolcke and E. Shriberg. Automatic linguistic segmentation of conversational speech. In *Proc. ICSLP '96*, volume 2, pages 1005–1008, Philadelphia, PA, 1996. URL `citeseer.ist.psu.edu/stolcke96automatic.html`.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26:339–373, 2000. URL `citeseer.ist.psu.edu/stolcke00dialogue.html`.

S. Sumec. Multi camera automatic video editing. In *Proc. ICCVG*, 2004.

D. Surendran and G. A. Levow. Dialog act tagging with support vector machines and hidden Markov models. In *Proc. Interspeech - ICSLP*, September 2006.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting emotional polarity of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan, 2005.

D. Talkin. A robust algorithm for pitch tracking (RAPT). In W B Kleijn and K K Paliwal, editors, *Speech Coding and Synthesis*, pages 495–518. Elsevier, 1995.

D. Traum. Issues in multi-party dialogues. In F. Dignum, editor, *Advances in Agent Communication*, pages 201–211. Springer-Verlag, 2004.

Dolf Trieschnigg and Wessel Kraaij. Hierarchical topic detection in large digital news archives: Exploring a sample based approach. *Journal of Digital Information Management*, 3(1), 2005.

S. Tucker and S. Whittaker. Accessing multimodal meeting data: Systems, problems and possibilities. In S. Bengio and H. Bourlard, editors, *Machine Learning for Multimodal Interaction, First International Workshop, MLMI 2004, Martigny, Switzerland, June 21-23, 2004, Revised Selected Papers*, volume 3361 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2005.

S. Tucker and S. Whittaker. Time is of the essence: An evaluation of temporal compression algorithms. In *Conference on Human Factors in Computing Systems (CHI)*, 2006.

G. Tur, D. Hakkani-Tur, A. Stolcke, and E. Shriberg. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27(1):31–57, 2001.

Peter Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

Peter D. Turney. Thumbs up or thumbs down: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL 2002*, 2002.

S. Uchihashi. Direct camera control for capturing meetings into multimedia documents. In *Proc. ICME*, 2001.

U.M.Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on AI*, pages 194–202, 1995.

Masao Utiyama and Hitoshi Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of the 28th Annual Meeting of the ACL*, 2001.

A. van den Bosch. Wrapped progressive sampling search for optimizing learning algorithm parameters. In N. Taatgen R. Verbrugge and L. Schomaker, editors, *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*, 2004.

D. A. van Leeuwen and M. Huijbregts. Ami speaker diarization system for nist rt06s meeting data. In *Proc. MLMI 2006*, Washington D.C., 2006.

P. van Mulbregt, J. Carp, L. Gillick, S. Lowe, and J. Yamron. Segmentation of automatically transcribed broadcast news text. In *Proceedings of the DARPA Broadcast News Workshop*, pages 77–80. Morgan Kaufman Publishers, 1999.

A. Venkataraman, A. Stolcke, and E. Shirberg. Automatic dialog act labeling with minimal supervision. In *Proceedings of the 9th Australian International Conference on Speech Science & Technology*, December 2002.

A. Venkataraman, L. Ferrer, A. Stolcke, and E. Shriberg. Training a prosody-based dialog act tagger from unlabeled data. *Proc. of the IEEE ICASSP*, April 2003.

A. Venkataraman, Y. Liu, E. Shriberg, and Stolcke A. Does active learning help automatic dialog act taggin in meeting data. In *Proceedings Eurospeech*, 2005.

V. Venkataramani, S. Chakrabartty, and W. Byrne. *Gini*support vector machines for segmental minimum Bayes risk decoding of continuous speech. *Computer Speech and Language*, 21(3):423–442, July 2007.

A.T. Verbree, R.J. Rienks, and D.K.J. Heylen. First steps towards the automatic construction of argument-diagrams from real discussions. In P. Dunne and T.J.E. Bench-Capon, editors, *Proceedings of the in 1st International Conference on Computational Models of Argument*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 183–194, Liverpool, UK., 2006a.

A.T. Verbree, R.J. Rienks, and D.K.J. Heylen. Dialogue-act tagging using smart feature selection: results on multiple corpora. In *The first International IEEE Workshop on Spoken Language Technology (SLT)*, Palm Beach, Aruba, December 2006b.

D. Verbree, R. Rienks, and D. Heylen. Dialogue-act tagging using smart feature selection; results on multiple corpora. In *IEEE Spoken Language Technology Workshop*, pages 70–73, December 2006c.

R. Vertegaal. *Look who is talking to whom*. PhD thesis, University of Twente, Enschede, The Netherlands, September 1998.

Renata Vieira. How to evaluate systems against human judgment on the presense of disagreement? In *Proc. workshop on joint evaluation of computational processing of Portugese at PorTAL 2002*, June 2002.

A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In *IEEE Trans. on Information Theory*, 1977.

A. Waibel, M. Bett, F. Metze, K. Ries, T. Schaaf amd T. Schultz, H. Soltau, H. Yu, and K. Zechner. Advances in automatic meeting record creation and access. In *Proceedings of ICASSP*, 2001.

A. Waibel et al. CHIL: Computers in the human interaction loop. In *Proc. NIST ICASSP Meeting Recogn. Worksh.*, 2004.

F. Wallhoff, M. Zobl, and G. Rigoll. Action segmentation and recognition in meeting room scenarios. In *Proc. ICIP*, 2004.

V. Warnke, R. Kompe, H. Niemann, and E. Nöth. Integrated Dialog Act Segmentation and Classification using Prosodic Features and Language Models. In *Proc. European Conf. on Speech Communication and Technology*, volume 1, pages 207–210, 1997. URL `citeseer.ist.psu.edu/warnke97integrated.html`.

Geoffrey I. Webb. MultiBoosting: A technique for combining Boosting and Wagging. *Machine Learning*, 40(2):159–196, 2000.

N. Webb, M. Hepple, and Y. Wilks. Dialogue act classification based on intra-utterance features. In *Proceedings of the AAAI 05*, 2005.

P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with ferret. In *Proc. of MLMI 2004, Martigny, Switzerland*, pages 12–21, 2004.

P. Wellner, M. Flynn, S. Tucker, and S. Whittaker. A meeting browser evaluation test. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 2021–2024, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-002-7. doi: http://doi.acm.org/10.1145/1056808.1057082.

S. Whittaker. Theories and methods in mediated communication. In A. Graesser, M. Gernsbacher, and S. Goldman, editors, *The Handbook of Discourse Processes*, pages 243–286. Erlbaum, New Jersey, 2002.

S. Whittaker, R. Laban, and S. Tucker. Analysing meeting records: An ethnographic study and technological implications. In *Proceedings of MLMI 2005*, 2005.

S. Whittaker, S. Tucker, K. Swampillai, and R. Laban. Design and evaluation of systems to support interaction capture and retrieval. *Personal and Ubiquitous Computing*, to appear.

Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 735–740, Austin, Texas, 2000.

Janyce Wiebe and Ellen Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, pages 486–497, Mexico City, Mexico, 2005.

Janyce Wiebe, Rebecca Bruce, and Thomas O'Hara. Development and use of a gold standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 246–253, College Park, Maryland, 1999.

Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Computational Linguistics*, 30(3):277–308, 2004.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210, 2005.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: A system for subjectivity analysis. In *Proc. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005) Companion Volume (software demonstration)*, 2005a.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, 2005b.

I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000a.

Ian. H. Witten and Eibe Frank. *Data mining: Practical machine learning tools and techniques with Java implementations*. San Francisco: Morgan Kaufmann, 2000b.

Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and technique*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

B. Wrede and E. Shriberg. Spotting hot spots in meetings: Human judgements and prosodic cues. In *Proceedings of EUROSPEECH 2003*, 2003a.

Britta Wrede and Elizabeth Shriberg. Spotting "hot spots" in meetings: Human judgments and prosodic cues. In *Proceedings of EUROSPEECH*, 2003b.

XTAG Research Group. A lexicalized tree adjoining grammar for english. Technical Report IRCS-01-03, IRCS, University of Pennsylvania, 2001.

Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, pages 129–136, Sapporo, Japan, 2003.

K. Zechner and A. Waibel. DIASUMM: Flexible summarization of spontaneous dialogues in unrestricted domains. In *Proc. of COLING-2000*, 2000.

Klaus Zechner. *Automatic Summarization of Spoken Dialogues in Unrestricted Domains*. PhD thesis, Carnegie Mellon University, 2001.

Dell Zhang. Key substring group software; available from http://www.dcs.bbk.ac.uk/~dell/publications/ dellzhang_kdd2006_supplement.html, 2006a.

Dell Zhang and Wee Sun Lee. Extracting key-substring-group features for text classification. In *Proceedings of KDD*, 2006.

Dell Zhang, Xi Chen, and Wee Sun Lee. Text classification with kernels on the multinomial manifold. In *Proceedings of SIGIR*, 2005.

Le Zhang. *http* : *//homepages.inf.ed.ac.uk/s*0450736*/maxent_toolkit.html*, 2006b.

M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke. Toward joint segmentation and classification of dialog acts in multiparty meetings. In *Proceedings of 2nd conference on Machine Learning and Multi-Modal Interactions (MLMI'05)*, 2005a.

M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke. A* based joint segmentation and classification of dialog acts in multiparty meetings. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, november 2005b.

M. Zimmermann, A. Stolcke, and E. Shriberg. Joint segmentation and classification of dialog acts in multi-party meetings. In *Proc. 31st ICASSP*, volume 1, pages 581–584, Toulouse, France, 2006a.

Matthias Zimmermann, Yang Liu, Elizabeth Shriberg, and Andreas Stolcke. Toward joint segmentation and classification of dialog acts in multiparty meetings. In Steve Renals and Samy Bengio, editors, *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI05), Volume 3869 of Lecture Notes in Computer Science*, pages 187–193, 2006b.